

Министерство образования Республики Беларусь
Белорусский государственный университет
Механико-математический факультет
Кафедра веб-технологий и компьютерного моделирования

СОГЛАСОВАНО

Заведующий кафедрой

_____ М.В. Игнатенко

«15» марта 2024 г.

СОГЛАСОВАНО

Декан факультета

_____ С.М. Босяков

«26» апреля 2024 г.

Методика преподавания информатики

Электронный учебно-методический комплекс для специальности:
6-05-0533-06 «Математика»

Регистрационный № 2.4.2-24/477

Автор:

Аленский Н.А., старший преподаватель.

Рассмотрено и утверждено на заседании Научно-методического совета БГУ
31.05.2024 г., протокол № 8.

Минск 2024

УДК 004(072)(075.8)
А 484

Утверждено на заседании Научно-методического совета БГУ
Протокол № 8 от 31.05.2024 г.

Решение о депонировании вынес:
Совет Механико-математического факультета
Протокол № 9 от 26.04.2024 г.

А в т о р:

Аленский Николай Алексеевич, старший преподаватель кафедры веб-технологий и компьютерного моделирования механико-математического факультета Белорусского государственного университета.

Рецензенты:

кафедра информатики и методики преподавания информатики учреждения образования «Белорусский государственный педагогический университет им. М. Танка» (заведующий кафедрой Чубаров Сергей Ильич, кандидат физико-математических наук, доцент).

Котов Владимир Михайлович, заведующий кафедрой дискретной математики и алгоритмики факультета прикладной математики и информатики Белорусского государственного университета, доктор физико-математических наук, профессор.

Аленский Н. А. Методика преподавания информатики : электронный учебно-методический комплекс для специальности: 6-05-0533-06 «Математика» / Н. А. Аленский ; БГУ, Механико-математический фак., Каф. веб-технологий и компьютерного моделирования. – Минск: БГУ, 2024. – 99 с. : табл. – Библиогр.: с. 96–99.

Электронный учебно-методический комплекс (ЭУМК) по учебной дисциплине «Методика преподавания информатики» предназначен для студентов специальности 6-05-0533-06 «Математика». ЭУМК содержит тексты лекций, планы семинарских занятий и вопросы для обсуждения на них, содержание докладов и рефератов и рекомендуемые темы для них, типы и примеры тестов, экзаменационные вопросы, содержание дисциплины и её учебно-методическую карту, список рекомендуемой литературы.

ОГЛАВЛЕНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	6
1. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ.....	8
1.1. Понятие информатики	8
1.2. История преподавания информатики и её современное состояние	10
1.2.1. Основные этапы преподавания информатики	10
1.2.2. Формирование содержания образования	12
1.3. Дисциплина «Методика преподавания информатики» в системе педагогических знаний	15
1.3.1. Цели и задачи учебной дисциплины	15
1.3.2. Место дисциплины и её связи с другими ветвями наук .	17
1.3.3. Требования к усвоению дисциплины.....	18
1.3.4. Другие особенности дисциплины.....	20
1.4. Цели преподавания информатики	23
1.4.1. Значение предмета	23
1.4.2. Цели преподавания школьной информатики.....	26
1.4.3. Компьютерная грамотность и образованность.....	28
1.5. Принципы дидактики и преподавание информатики	33
1.5.1. Принцип научности	33
1.5.2. Сознательность усвоения и деятельности, доступность и наглядность, активность и самостоятельность.....	34
1.5.3. Другие принципы дидактики	35
1.6. Методы и организационные формы обучения информатике	37
1.6.1. Особенность методов и их реализация на практике.....	37
1.6.2. Мыслительные операции и работа с вычислительной техникой	39
1.6.3. Организационные формы обучения информатике.....	41
1.6.4. Вспомогательные организационные формы	46
1.7. Урок информатики	48
1.7.1. Урок и типы уроков.....	48

1.7.2.	Роль учителя и особенности урока по информатике	50
1.7.3.	Технические и программные средства обучения информатике.....	51
1.8.	Методика преподавания основ алгоритмизации и программирования	54
1.8.1.	Методические особенности раздела.....	54
1.8.2.	Общие методические принципы обучения	54
1.8.3.	Формы занятий.....	58
1.8.4.	Изучение основ алгоритмизации и программирования с использованием компьютеров	59
1.9.	Решение задач по основам алгоритмизации и программирования	60
1.9.1.	Выбор задач	60
1.9.2.	Упражнения на составление и анализ блок-схем задач ...	61
1.9.3.	Комплекс взаимосвязанных задач.....	63
1.9.4.	Методические приёмы, используемые при решении задач	65
1.10.	Методика преподавания некоторых основных тем информатики	67
1.10.1.	Преподавание текстовых и графических редакторов ...	68
1.10.2.	Преподавания вопросов, связанных с информацией ...	70
1.10.3.	Методика изучения систем счисления и представления информации	75
1.10.4.	Изучение графических возможностей языка «Pascal» ..	77
1.10.5.	Некоторые особенности изучения других разделов	79
2.	ПРАКТИЧЕСКИЙ РАЗДЕЛ.....	82
2.1.	Вопросы для управляемой самостоятельной работы	82
2.2.	Практикум по информатике	83
2.3.	Содержание реферата (доклада с презентацией) по частной методике	85
2.4.	Темы рефератов (докладов с презентацией) по частной методике	87
3.	РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ.....	89

3.1. Типы и примеры тестов	89
3.2. Экзаменационные вопросы	91
3.3. Перечень рекомендуемых средств диагностики	93
4. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ	94
4.1. Содержание учебного материала	94
4.2. Электронные ресурсы	96
4.3. Список рекомендуемой литературы	97

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Электронный учебно-методический комплекс (ЭУМК) по учебной дисциплине «Методика преподавания информатики» создан в соответствии с ОСВО 6-05-0533-06 от 4 августа 2023 г., учебных планов 6-05-0533-06 Математика №6-05-05-027/пр. от 30.01.2023 и №6-05-0533-06 Математика. № 6-5.4-54/01. от 15.05.2023.

Учебно-методический комплекс преследует цель по оказанию посильной помощи студентам в усвоении учебного и нормативного материала, сориентировать в подборе специальной литературы для подготовки к семинарским занятиям и экзамену по дисциплине «Методика преподавания информатики» в соответствии с учебной программой учреждения высшего образования по учебной дисциплине для специальности первой степени высшего образования 6-05 0533-06 Математика. № УД-12152/уч [Электронный ресурс]. – Режим доступа: <https://elib.bsu.by/handle/123456789/304460>

Основной целью дисциплины «Методика преподавания информатики» является подготовка студентов к предстоящему преподаванию информатики в учреждениях общего среднего образования. Содержание данной дисциплины отвечает на три основных вопроса методических дисциплин: **зачем** учить информатике, **что** изучать и **как** обучать информатике.

Задачи учебной дисциплины:

изучение студентами научных и психолого-педагогических основ структуры и содержания предмета информатики учреждений общего среднего образования, понимание методических идей, заложенных в них;

формирование знаний о месте и значении школьной информатики в образовании школьника, о связи этого предмета с другими изучаемыми в школе предметами;

освоение содержания учебников и программ по информатике;

обучение студентов ориентированию в постоянно изменяющемся парке вычислительной техники и программном обеспечении и применению их в учебном процессе.

В результате изучения дисциплины студент должен:

знать:

- основные понятия и принципы дидактики информатики;
- структурные элементы урока и основные требования к ним;
- виды планирования деятельности учителя;
- методы и организационные формы обучения школьной информатике;
- программы по информатике, структуру и содержание учебных пособий и учебников;
- учебно-методическое, техническое и программное обеспечение предмета;

уметь:

- осуществлять планирование работы учителя;
- разрабатывать и составлять план-конспект урока, факультатива, кружка и других видов, и форм учебной деятельности;

- выполнить анализ проведения урока, факультатива, кружка;
- использовать современное программное обеспечение, электронные и бумажные учебные пособия и педагогические технологии в учебном процессе;
- организовать работу в современном кабинете информатики;
- осуществлять внеклассную и внешкольную работу;

владеть:

- классическими и новыми методами обучения;
- организационными формами проведения занятий;
- различными методами контроля и оценки знаний учащихся.

Дисциплина изучается в 5 семестре дневной формы обучения и в 5-6 семестрах заочной формы обучения. Всего на изучение учебной дисциплины «Методика преподавания информатики» отведено:

– для очной формы получения высшего образования – 96 часов, в том числе 36 аудиторных часов, из них: лекции – 24 часа, семинарские занятия – 10 часов, управляемая самостоятельная работа – 2 часа.

– для заочной формы получения высшего образования – 8 аудиторных часов, из них 6 лекций и 2 семинарских занятия, контрольная работа.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма текущей аттестации – экзамен.

В структуру ЭУМК входит:

1. Теоретический раздел, который включает краткий конспект лекций по учебной дисциплине.

2. Практический раздел, который включает планы семинарских занятий и вопросы для обсуждения на них, содержание докладов и рефератов и рекомендуемые темы для них.

3. Раздел контроля знаний, который содержит перечень контрольных мероприятий управляемой самостоятельной работы студентов, типы и примеры тестов, экзаменационные вопросы и билеты.

4. Вспомогательный раздел со списком рекомендуемой литературы.

Работа студента с ЭУМК должна включать ознакомление с тематическим планом учебной дисциплины, представленным в учебной программе учреждения высшего образования, в которой можно получить информацию о тематике лекций, лабораторных занятий и рекомендуемой литературе. Для подготовки к лабораторным занятиям рекомендуется использовать материалы, представленные в теоретическом и практическом разделах ЭУМК, а также материалы для контроля самостоятельной работы студентов. В ходе подготовки к экзамену целесообразно ознакомиться с требованиями к компетенциям по учебной дисциплине, изложенными в учебной программе учреждения высшего образования, а также перечнем вопросов к экзамену.

1. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

1.1. Понятие информатики

Прежде чем говорить о преподавании учебного предмета “Информатика”, попытаемся определить область, именуемую этим словом. Появление и начальное становление информатики как науки относится ко второй половине прошлого века. Терминологические и понятийные трудности, связанные с сущностью самого понятия “информатика” (равно как и производных понятий) не преодолены до сих пор. Его толкование, которое приведено далее, до настоящего времени нельзя считать установившимся и общепринятым. Нет строгого определения этого термина. Необходимо заметить, что в русском языке примерно с середины 60-х годов такой же термин был связан с узкой областью научно-технической информации и документооборота. Подобное определение связывало информатику с библиотковедением, библиографией, методами поиска информации в массивах документов и другими дисциплинами. Эта дисциплина изучала структуру и общие свойства научной информации, а также закономерности ее создания, преобразования, передачи и использования в различных сферах человеческой деятельности. Мы будем рассматривать информатику в другом смысле.

В конце 40-х годов прошлого века появился и широко использовался термин “кибернетика” как общая наука об управлении и связи в системах различной природы – искусственных, биологических, социальных и других. Развиваясь одновременно с прогрессом электронных вычислительных машин, кибернетика со временем превращалась в общую науку о преобразовании информации. Вскоре после появления термина “кибернетика” в мировой науке стало использоваться англоязычное “Computer Science” (компьютерная наука). Этот термин и сейчас в США, Канаде и некоторых странах латиноамериканского континента достаточно широко распространен вместо термина “информатика”.

На рубеже 60-х и 70-х годов XX века во французском языке было введено слово „информатика” (informatique), образованное, скорее всего, как производное от двух французских слов: «ИНФОРмация» (informatione) и «автоМАТИКА» (avtomatique) или «ИНФОРМАция» (informatione) и «автомаТИКА». Новый термин получил распространение в СССР и странах Западной Европы, а позже в России и странах СНГ, включая и Республику Беларусь. Этим словом вначале обозначали область автоматизированной обработки информации, при которой основную часть операций над ней выполняют ЭВМ и другие технические средства, а окончательные решения по управлению принимает человек. Были и другие определения. Например, известный советский ученый в этой области академик А. П. Ершов утверждал, что этот термин вводится в русский язык как название фундаментальной естественной науки, изучающей процессы передачи и обработки информации. При таком толковании информатика оказывается более связанной с

общенаучными категориями и ее можно отнести к фундаментальным наукам, что отражает общенаучный характер понятия информации и процессов ее обработки. Многие ученые подчеркивали, что информатика — это следствие развития ЭВМ.

Область интересов информатики — это структура, общие свойства информации и вопросы, связанные с процессами поиска, сбора, хранения, преобразования, передачи и использования её в различных сферах человеческой деятельности. Обработка огромных объемов и потоков информации немыслима без автоматизации и систем связи. Поэтому ЭВМ и современные информационные и коммуникационные технологии являются и фундаментальным ядром, и материальной базой информатики.

В школьных учебниках можно встретить, например, следующее определение: „Информатика — наука, исследующая законы и методы переработки и накопления информации”. Можно увидеть и следующее: „... Эта наука позволяет не только понять принципы работы и возможности использования ЭВМ, но и даёт представление о законах и методах представления информации при общении людей и в жизни общества”.

Дать определение, отражающее специфику информатики, не просто. Оно является одним из фундаментальных и неопределяемых понятий в строгом смысле слова. Представляется удачным рабочее определение этого термина с помощью трёх компонент:

HARDWARE – “твёрдые”, аппаратные средства (иногда говорят “железо”). Дословно переводится как аппаратура;

SOFTWARE – “мягкие”, программные средства, к которым можно и нужно отнести не только программы, но и обрабатываемые данные. Дословно переводится как программное обеспечение;

BRAINWARE – алгоритмическая или мозговая составляющая (brain — мозг). Это не просто алгоритмы обработки данных, что снова вело бы к SOFT, а интеллектуальная составляющая, знание, которое воплощено и в программах, и в данных, и в аппаратуре;

Это определение информатики удивительно перекликается с наивно-фундаментальным представлением древних о существовании четырёх стихий – первоэлементов: 1) земля – HARD; 2) вода – SOFT; данные и программы не постоянны, текучи, изменчивы; 3) воздух – это та сфера, в которой происходят информационные процессы; 4) огонь – BRAIN; Таким образом, анализ различных определений этого термина позволяет выделить следующие аспекты его содержания:

Совокупность средств автоматизированной информационной техники и технологии;

Особая отрасль экономики, включающая всю сферу автоматизированной обработки и использования информации;

Отрасль научного знания, изучающая процессы передачи и обработки информации и средства её автоматизированной обработки (техническое, математическое и программное обеспечение);

Теория научной информации и научно-информационной деятельности с акцентом на средства автоматизации.

1.2. История преподавания информатики и её современное состояние

1.2.1. Основные этапы преподавания информатики

Необходимо обратить внимание на то, что информатика является одним из немногих общеобразовательных учебных предметов, родившихся в XX веке. Большинство учебных предметов для сферы общего образования начали изучать еще в XIX веке. Рассмотрим основные этапы становления школьной информатики в прошлом веке в странах бывшего Советского Союза (с 1950 до 1991 года) и Республике Беларусь (с 1991 года).

В 50-е годы практиковалось изучение программирования в ряде школ города Новосибирска. Главным инициатором этого нововведения был академик А. П. Ершов. В 60-е годы осуществлялась подготовка программистов в московских школах с математической специализацией. В 70-е годы в Москве, Новосибирске и некоторых других научных центрах в странах бывшего СССР готовили школьников по специальностям, связанным с ЭВМ. В конце 70-х годов началось массовое производство малых ЭВМ, что расширило области применения и доступность ЭВМ. Началась разработка концепции школьной информатики, главным инициатором которой был академик А. П. Ершов. Были разработаны уникальная система программирования «Школьница» и язык Рапира. В 1982 году министерство просвещения СССР приняло решение о введении калькуляторов в учебный процесс школы.

Основные направления реформы общеобразовательной и профессиональной школы, принятые в **1984 году**, одной из основных задач провозгласили обеспечение всеобщей компьютерной грамотности. Была разработана программа нового школьного предмета, выпущены учебники для учащихся по дисциплине «Основы информатики и вычислительной техники» и методические пособия для учителей. Начали издавать новый научно-методический всесоюзный журнал «Информатика и образование». Появились отечественные компьютеры, предназначенные для обучения школьников. В **1985 и 1986** годах проводилась ускоренная подготовка школьных учителей математики и физики к преподаванию нового школьного предмета. На физико-математических факультетах педагогических институтов начала осуществляться подготовка студентов в области информатики (в основном в рамках различных спецкурсов). Но такая подготовка по программированию носила исключительно образовательный характер. Она не была ориентирована на преподавание этого предмета школьникам.

В **1985** г. разработана первая экспериментальная программа предмета «Основы Информатики и Вычислительной Техники», а на её основе были подготовлены и изданы первые пробные учебники по информатике в двух частях под редакцией А. П. Ершова. С **1 сентября 1985 г.** началось преподавание этого

предмета в массовой школе и подготовка учителей информатики в педагогических институтах по новым учебным планам. В **1986** году была опубликована вторая программа преподавания школьного курса информатики в машинном варианте и объявлен конкурс на создание учебника по дисциплине «Основы Информатики и Вычислительной Техники». В соответствии с конкурсной программой разработано первое поколение учебников по информатике авторов В. А. Каймина и др. , А. Г. Кушнеренко и др. , А. Г. Гейна и др.

В **1987** году в педагогических институтах была введена дисциплина «Методика Преподавания Информатики и Вычислительной Техники», которая была предназначена для подготовки будущих преподавателей по специальности «Учитель математики, информатики и вычислительной техники». Анализируя первую программу этого курса, мы убеждаемся, что в результате его изучения будущий учитель должен был понимать значение школьного предмета «Основы Информатики и Вычислительной Техники» в общем образовании, уметь объяснить принципы отбора содержания этого предмета, овладеть основными методическими и дидактическими формами и приемами преподавания этого предмета, понимать взаимосвязи его с другими школьными предметами, осознавать определяющую роль предмета в решении компьютеризации образования.

В первой половине 90-х годов продолжалась оживленная дискуссия по вопросу преподавания информатики в Республике Беларусь, были разработаны новые концепции информатизации образования и новое поколение программ предмета и учебников белорусских авторов, таких, как Ю. А. Быкадоров, В.М. Котов, О. И. Мельников, А. Т. Кузнецов, А. И. Павловский и др. Были переделаны и переизданы первые учебники российских авторов. Всё более очевидной становилась нецелесообразность и недостаточность обучения информатике только в старших классах школы. К началу 90-х годов постепенно в Республике Беларусь начинает складываться новая структура обучения информатике в общем среднем образовании. В **1994** году в Беларуси основной курс информатики перенесли в базовую школу в 8 – 9 классы. Этот двухгодичный курс был обязательным для всех типов учебных заведений. В 10 - 11 классах курс продолжался на углубленном и повышенном уровнях, в специальных классах, школах, лицеях и гимназиях.

В **1998** году была утверждена программа «Информатизация системы образования». В 2000 году был принят документ «Основные направления развития информатизации общего среднего образования», в 2003 г. — «Электронная Беларусь». В 2002 году создан координационный совет по проблемам информатизации образования, переименованный в 2005 году в «Совет по проблемам информатизации образования при Министерстве Образования Республики Беларусь». В 2007 году принята программа «Комплексная информатизация системы образования Республики Беларусь на 2007 – 2010 годы».

Однако в современном мире глобализации дать одни лишь знания на всю жизнь очень трудно даже опытному педагогу. В связи с этим на современном этапе развития методики поставлена несколько иная цель. Она состоит в том, чтобы научить человека самостоятельно добывать информацию из различных источников, производить её грамотный отбор, обработку, хранение и творческое использование. Всё это предполагает грамотное развитие информационной культуры учащихся. И основную роль в её развитии оказывают как раз преподаватели информатики.

С 2006 года информатика изучается с 6-го класса по 11-ый. В рамках этих пяти лет обучения реализованы современные концепции и методы обучения информатике. Также в рамках факультативных занятий введены учебные и методические комплексы для занятий информатикой в младших классах по желанию учеников.

1.2.2. Формирование содержания образования

Общие требования к содержанию образования и принципы его формирования указаны в соответствующих нормативных документах. В обучении любой дисциплины можно выделить три основные функции: образовательную, воспитательную и развивающую. Центральное положение среди них занимает образовательная функция, которая предполагает усвоение научных знаний, формирование предметных навыков. При грамотном формировании содержания мы обратим внимание на две группы основных факторов, традиционно находящихся в диалектическом противоречии:

1. *Научность и практичность.* Содержание учебного предмета информатики должно идти от науки, т.е. не противоречить её современному состоянию и быть методологически цельным. Изучение предмета информатики должно давать такой уровень фундаментальных познаний учащихся, который мог бы обеспечить их подготовку к будущей жизни и профессиональной деятельности.

2. *Доступность и общеобразовательность.* Включаемый в предмет материал должен быть посилен основной массе учащихся, отвечать уровню их умственного развития и запасу знаний, умений и навыков. Этот учебный курс должен содержать все наиболее общезначимые, общеобразовательные и общекультурные сведения из соответствующих разделов науки информатики;

Школьный предмет информатики, с одной стороны, должен быть современным, отвечать усложняющимся требованиям науки и практики, а с другой — быть простым и доступным для изучения. Совмещение этих двух противоречивых требований было и остаётся наиболее сложной методической задачей для педагога. Не удивительно, что формирование содержания образования по информатике — сложный и противоречивый процесс, который охватывает период от начала 60-х годов прошлого века до настоящего времени. При этом надо заметить, что фактическое состояние теоретических разработок

этой проблемы и экспериментальных достижений в этой области слабо коррелируется с фактическим состоянием модели содержания этого предмета, которая находится на вооружении практического учителя информатики.

В основу разработки первой программы школьного предмета «Основы Информатики и Вычислительной Техники» в **1985** году были положены три базовых понятия: информация, алгоритм, ЭВМ. Они и составили концептуальную основу первой версии содержания этого школьного предмета. Ставились следующие задачи нового предмета:

- овладение основными умениями алгоритмизации;
- систематизация и завершение алгоритмической линии курса алгебры;
- формирование представлений о возможности автоматизации выполнения алгоритма;

- усиления прикладной направленности алгоритмической линии, заключающиеся в конкретной реализации алгоритмов решения задач с помощью техники;

- формирование представления об этапах решения задач на электронно-вычислительных машинах;

- ознакомление с основами современной вычислительной техники на примере рассмотрения общих принципов работы микрокомпьютера;

- ознакомление с основными сферами применения вычислительной техники, её ролью в развитии общества.

Этот курс преподавался в двух старших классах средней школы. В 9-м классе на изучение этого предмета отводилось 34 часа (1 час в неделю). В 10-м классе предлагались два варианта. Первый из них, полный курс объемом 68 часов, рекомендовался для школ, располагающих вычислительными машинами или имеющих возможность организовать занятия школьников на вычислительных центрах различных организаций. Краткий курс объемом 34 часа предназначался для школ, не имеющих такой возможности. Теоретическая часть для 10-го класса была одинаковой для обоих вариантов, а отличались объем и содержание практической части. Для школ, имеющих доступ к электронной технике, дополнительные 34 часа рекомендовалось использовать для решения задач в машинном варианте и отработки навыков применения компьютеров и его программного обеспечения.

Кроме определения содержания для информатики, как ни для какого другого предмета, важным и непростым является вопрос построения оптимальной последовательности изучения, соответствующей логике науки и уровню развития учащихся. С учетом этого предмет «Основы Информатики и Вычислительной Техники» в 9-м классе состоял из следующих тем:

- введение — 2 часа;

- алгоритмы. Алгоритмический язык — 6 часов;

- алгоритмы работы с величинами — 10 часов;

- построение алгоритмов для решения задач — 16 часов.

В 10-м классе предлагались по программе следующие темы:

- принципы устройства и работы ЭВМ — 12 часов;

знакомство с программированием — 16 часов;

роль ЭВМ в современном обществе, перспективы развития вычислительной техники — 2 часов;

экскурсия на вычислительный центр — 4 часов.

Анализ первой программы показывает, что был принят алгоритмический подход в изучении информатики. Больше всего часов (48 из 68) отводилось на построение алгоритмов и разработку на их основе программ для различных машин. Избыточность алгоритмизации и программирования подвергалась резкой критике. Но в оправдание разработчиков программы и соответствующих учебников надо учитывать реальное состояние отечественной практики на то время, возможности оснащения школ компьютерами, готовность педагогических кадров и другие обстоятельства. Этим можно объяснить и то, что преподавание информатики на начальном этапе носило в основном «безмашинный» характер.

Основным средством описания алгоритмов являлся специально разработанный под руководством академика А. П. Ершова «учебный алгоритмический русскоязычный» язык, частично напоминающий «*Pascal*». Теперь можно уверенно сказать, что он сыграл большую роль в истории преподавания информатики и полностью себя оправдал. Он обладал рядом привлекательных свойств, которые и объясняют, почему именно ему было отдано предпочтение перед распространенными в то время официальными языками программирования («*Basic*», «*Pascal*» и др.). Язык использовал русскую или национальную лексику, что не создавало дополнительных трудностей, не имеющих никакого отношения к сути предмета. В отличие, например, от языка *Basic*, школьный язык был построен на современных идеях структурного программирования, что наилучшим образом соответствовало операционному мышлению человека. Язык не зависел от машины, в нем не было деталей, связанных с её устройством. Это позволяло основное внимание сосредоточить на алгоритмической сути решаемых задач.

В 1986 году вышла вторая учебная программа. Она, как и первая, была рассчитана на обучение этому предмету в двух старших классах в объеме 102 часа. Между этими первыми двумя вариантами программ существенных различий не было. Основная особенность её в том, что эта программа представляла собой машинный вариант и была ориентирована на обучение информатике в условиях активной работы школьников в кабинете вычислительной техники. Поэтому больше времени отводилось на практическую работу. Важным элементом этой программы являлся впервые официально объявленный примерный перечень программного обеспечения в поддержку курса «Основы Информатики и Вычислительной Техники». Он повторял все разделы программы и состоял из компонентов, которые в условиях объявленного конкурса следовало рассматривать как приглашение к разработке. Позже большинство из них действительно были разработаны, причем в нескольких вариантах. Особое положение в перечне программных средств

занимал интерпретатор учебного языка, первая версия которого уже фактически действовала к моменту объявления конкурса.

Концепция содержания, заложенная в программе машинного варианта, была практически реализована в нескольких подготовленных на ее основе учебных пособиях. Учебные книги по информатике авторов В. А. Каймина и др., А. Г. Кушнеренко и др., А. Г. Гейна и других авторов получили широкое распространение в школах. Появление этих учебников вызвало поток жёсткой критики, которая относилась не столько к самим книгам, сколько к программе, их породившей. Критиковались в основном гипертрофированная линия алгоритмизации и программирования и слабое продвижение в развитии фундаментальных основ школьной информатики.

Необходимость введения нового предмета объяснялась широким внедрением вычислительной техники и особенно персональных компьютеров во все области человеческой деятельности, в том числе в сферу быта и систему народного образования. Бурное развитие микроэлектроники и на её базе производства электронных машин, а позже персональных компьютеров, совершенствование их программного обеспечения и средств общения с ними способствовало компьютеризации школы и других средних учебных заведений и введению предмета «Основы Информатики и Вычислительной Техники». Но освоение нового школьного предмета в первые годы в то время существенно тормозилось общим состоянием процессов информатизации общества, недостаточным развитием материальной базы и современных информационных технологий.

В настоящее время общие вопросы методики преподавания информатики в основном понятны и сформулированы, ответы на них в значительной степени найдены. Детально разработана и методика преподавания ряда конкретных тем. В то же время постоянная открытость информатики новому ставит более сложную и важную задачу – научить человека самостоятельно разрабатывать методику, научить методическому творчеству, передать ему опыт подобной творческой деятельности. Быстро изменчивая среда меняется настолько быстро, что отпадает необходимость в очень строгом понимании содержания самого предмета. На данном этапе развития многие темы привязаны к конкретным программам и приложениям, развитие которых происходит в очень краткие сроки, что не позволяет педагогу строго определять содержательную линию предмета.

1.3. Дисциплина «Методика преподавания информатики» в системе педагогических знаний

1.3.1. Цели и задачи учебной дисциплины

Информатика в учреждениях общего среднего образования разных видов преподаётся более четверти века. Однако методика преподавания этого предмета остается актуальной и в настоящее время. Вопросы и задачи в новой

педагогической науке продолжают возникать и требуют своего разрешения в виде теоретического обоснования и опытной проверки. Поэтому теория и методика обучения информатике в настоящее время продолжает бурно обсуждаться и развиваться.

Методика преподавания информатики формируется не на пустом месте. Исследование целей и содержания общего кибернетического образования, накопленный практический опыт преподавания учащимся элементов кибернетики, основ алгоритмизации и программирования, элементов логики, вычислительной и дискретной математики имеют, как видим из предыдущей темы, более чем полувековую историю. Будучи фундаментальным разделом педагогической науки, методика преподавания информатики опирается в своем развитии на философию, педагогику, психологию, информатику, а также обобщенный практический опыт средней школы.

Одновременно с введением информатики в школьное образование была проведена массовая переподготовка уже работающих учителей и выпускников педагогических университетов, а в **1987** году разработаны новые учебные планы для систематической подготовки учителей информатики. Вскоре появилось и первое учебное пособие по этой вузовской дисциплине, которое позже несколько раз переиздавалось. Методику преподавания информатики как новую дисциплину начали преподавать в педагогических университетах в **1987** году. Так как на тот момент существовало лишь фрагментарное её видение, отражённое в ряде статей, докладов и дискуссий, то преподавание дисциплины проводилось параллельно с его разработкой. Курс этой дисциплины сначала строился как методика решения задач на алгоритмизацию в «безмашинном» варианте.

Основной **целью** данной дисциплины является подготовка студентов к предстоящему преподаванию информатики в учреждениях общего среднего образования. Перед методикой преподавания информатики, как и перед всякой предметной школьной методикой, ставятся следующие вопросы:

Зачем учить информатике?

Что изучать в информатике?

Как обучать информатике?

Поэтому **задачи** дисциплины определены следующим образом:

обеспечение глубокого изучения студентами научных и психолого-педагогических основ структуры и содержания предмета информатики учреждений общего среднего образования, понимание методических идей, заложенных в них.

формирование знаний о месте и значении школьной информатики в образовании школьника, о связи этого предмета с другими изучаемыми в школе предметами.

освоение содержания учебников и программ по информатике.

освоение классических и новых методов обучения и организационных форм занятий с учетом особенностей информатики.

выработка умения организовать работу школьников в современном компьютерном классе.

обучение студентов ориентированию в постоянно изменяющемся мире вычислительной техники и программном обеспечении и применению их в учебном процессе.

Согласно концепции информатизации Республики Беларусь, методика преподавания информатики не может и не должна сводиться к преподаванию только информатики. Она обеспечивает реализацию следующей группы целей этой концепции:

изучение новых областей знаний, связанных с информатикой.

изучение современных информационных технологий, приобретение навыков их использования.

развитие и закрепление логико-алгоритмического стиля мышления, изучение современных методов программирования.

В результате изучения курса будущий учитель должен подготовиться не только к преподаванию информатики, но и к полноценной работе в компьютеризированной школе.

1.3.2. Место дисциплины и её связи с другими ветвями наук

Методика преподавания информатики посвящена изучению способов и методов обучения информатике в учреждениях общего среднего образования, где компьютерная грамотность становится составляющей профессионального уровня учителя. Будущие выпускники соответствующих специальностей будут работать учителями информатики. Более того, они будут не только учить школьников, но и способствовать применению компьютерных технологий в учреждениях общего среднего образования. Поэтому рассматриваемая здесь дисциплина играет важную роль в подготовке преподавателей информатики.

Основная особенность курса этой дисциплины – связь с другими, прежде всего методического цикла, предметами. С методикой преподавания математики она связана, можно сказать, генетически. Понятие алгоритма пришло из математики. Возможна такая аналогия: «множество - элемент = алгоритм - команда». С другой стороны, многие доказательства в математике имеют алгоритмическую структуру, и существует задача научить выявлять эту алгоритмическую составляющую в доказательствах. Время от времени возникает и продолжает возникать тенденция объединить информатику, например, с математикой. В лучшем варианте сильный учитель может реализовать межпредметные связи этих дисциплин. Но сама информатика может проиграть в этом случае в силу математизации задач и связанного с этим неизбежного роста их трудности. Это противоречит тенденции разгрузки начала изучения информатики как раз от математики. При этом могут быть утрачены и связи информатики с другими дисциплинами, и предметы будут по-прежнему

изучаться обособленно. Не связано с математикой и умение пользоваться многими готовыми программами и средствами.

Сходство данной дисциплины с методиками преподавания физики, химии проявляется в склонности информатики к опыту и эксперименту. Действительно, запуск программы на компьютере – это своеобразный эксперимент. С точки зрения физики компьютер — это прибор, поведение которого можно исследовать. У методики преподавания физики можно позаимствовать методы выполнения подобных опытов. Область пересечения интересов методики преподавания информатики с психологией – это, прежде всего, проблемы общения. Особая форма общения – компьютерная игра. Методика преподавания информатики уже предполагает и использует органическое включение игровых элементов в учебную деятельность с компьютером. Можно, например, рассматривать отладку программы как игру с компьютером. Своеобразной является связь с методикой изучения иностранного языка. Перевод, например, алгоритма на конкретный формальный язык (C, Pascal, и др.) – это вполне языковая, речевая деятельность, иногда и непростая проблема. Методика преподавания информатики» связана также с гуманитарными дисциплинами «Педагогика», «Психология». Кроме этого, имеется связь с учебной вычислительной практикой.

Будучи первой дисциплиной методического цикла, связанного с электронной техникой, методика преподавания информатики играет корректирующую роль в компьютеризации образования в целом. Проблемы информатизации образования лежат не только и не столько в области техники. Намного важнее наличие хороших программных средств. Но даже полное обеспечение системы образования мощными современными компьютерами с дружественным программным обеспечением ещё не означает автоматически, что проблемы информатизации образования решены. Большую роль играет степень психологической готовности и потребность педагогов и управленцев в пересмотре средств и способов своей деятельности в связи с использованием электронно-вычислительных машин.

1.3.3. Требования к усвоению дисциплины

Требования к уровню усвоения содержания дисциплины «Методика преподавания информатики» установлены соответствующим учебным заведением, осуществляемым подготовку преподавателей. В нем определены общенаучные знания и умения, которыми должен овладеть будущий педагог. В результате изучения дисциплины будущий учитель должен **знать**:

- теоретические основы методики преподавания информатики;
- основные понятия и принципы дидактики информатики;
- структурные элементы урока и основные требования к ним;

виды планирования деятельности учителя;
методы обучения школьной информатике;
программы по информатике, структуру и содержание учебных пособий и учебников;

учебно-методическое, техническое и программное обеспечение предмета.

В результате изучения дисциплины будущий учитель должен **уметь**:

разрабатывать и составлять план-конспект урока, факультатива, кружка;
планировать работу учителя;

проводить анализ плана-конспекта урока;

проводить анализ проведения урока, факультатива, кружка;

использовать современное учебно-методическое, техническое и программное обеспечение, электронные учебные пособия и различные педагогические технологии в учебном процессе;

организовать работу школьного кабинета информатики;

применять различные методы контроля и оценки знаний учащихся и осуществлять внеклассную и внешкольную работу;

Таким образом сформулированы основные требования к педагогу, которые ему необходимо постоянно контролировать. Чем прочнее усвоены и применены эти знания и требования к уроку, тем успешнее происходит педагогический процесс.

Для контроля качества образования используются следующие средства диагностики:

индивидуальные типовые задания;

тесты по отдельным разделам и дисциплине в целом;

письменные контрольные работы;

устный опрос во время занятий;

составление рефератов;

выступления по разработанным или данным темам;

письменный или устный экзамен.

Текущий контроль знаний выполняется по каждому разделу программы. В качестве итогового контроля по завершению изучения первого раздела рекомендуется электронное тестирование или письменная контрольная работа. Однако это не единственный вариант проверки. Можно ограничиться и устным опросом.

По второму разделу необходимо выполнить индивидуальные задания, темы которых могут быть заданы преподавателем, но предпочтительнее выбрать следующие темы:

массивы и операции над ними;

строки и их обработка;

графические возможности.

По каждому из них необходимо отладить программу. При этом желательно использовать дополнительно изученную современную систему программирования, отличную от той, которая изучалась на младших курсах в дисциплине «Методы программирования» или в общеобразовательном

учреждении. При оценке каждого из заданий учитывается уровень сложности, своевременность его сдачи, выполнение требований к заданиям, эффективность алгоритма, полноту тестирования.

Второй, а также третий разделы проверяются в форме разработки, презентации и обсуждения на практических занятиях методики изучения конкретной темы школьной информатики. Будущий педагог должен предложить методы изучения и формы занятий, показать, как используется персональный компьютер и другие технические средства, и в качестве примера разработать план-конспект одного урока. По некоторым темам этого раздела предлагается выполнить индивидуальные задания (системы управления базами данных, графический или текстовый редактор, сеть интернет, электронная почта или другие разделы). Итоговый результат работы студента выражен в виде учебно-методического комплекса по определенной теме курса, который включает:

- одну или несколько лекций по теме;
- подробный план проведения практических занятий;
- дидактический материал: упражнения, задачи, задания с решениями и комментариями или без них;
- контрольные вопросы;
- тесты с ответами и комментариями для самостоятельной работы или без них.

По результатам текущего контроля выставляется общая текущая отметка.

Итоговый контроль осуществляется в форме зачёта и/или экзамена. Для зачёта необходимо получить положительную отметку по каждому разделу. Экзамен предлагается проводить в два этапа: письменная контрольная работа с элементами тестирования и выполнение заданий на компьютере по изучаемым в школе темам. В результате получается экзаменационная отметка. Оценка знаний на экзамене производится по 10-балльной шкале. Возможны и другие варианты проведения экзамена.

1.3.4. Другие особенности дисциплины

Компьютер – достаточно дидактически мощное средство, которое заставляет пересматривать и содержание, и формы, и методы обучения, всю систему в конкретных дисциплинах. Рассмотрим некоторые примеры влияния компьютера на курс математики.

Традиционное повышенное внимание школьной математики к логарифмической функции связано с упрощением вычислений вручную при переходе от умножения чисел к сложению их логарифмов. При использовании компьютеров исчезает основание для приоритета логарифмов. Исследование графика функции может быть начато с его быстрого построения на экране и резко упрощается. Если функция в точке не существует, компьютер выдаёт сообщение об ошибке, например деление на нуль. С помощью компьютера легче понять смысл итерационных методов для решения тех или других задач (вычисление квадратного корня, решение уравнений и их систем и др.).

Оптимальным решением является интеграция информатики с другими дисциплинами, но не в учебных часах, а на уровне решаемых задач, системы знаний. Можно ожидать взаимного обогащения понятий, разнообразия связей и как следствие прочности и полезности знаний в целом.

Реальные возможности применения электронно-вычислительных машин должны практически раскрывать «учителя- предметники». Пока этот процесс идет крайне медленно, курс школьной информатики и кабинет вычислительной техники остаются в определенной изоляции. Не будем останавливаться на очевидных причинах этого. В оправдание этого можно сказать, что сфера образования вообще инерционна в большей степени, чем общество. Потери от игнорирования компьютеров при образовании школьников слишком велики, однако с каждым днём наблюдается увеличение количества использования вычислительной техники и компьютеров при изучении различных дисциплин.

Сильные внутрипредметные связи информатики порождают нетривиальную задачу поиска оптимального порядка изучения материала в соответствии с принципами последовательности и доступности изложения. При буквальном понимании последовательности предполагается, что учебный материал выстраивается в логическую цепочку или может быть представлен в виде дерева, где нет логических кругов, и повторение идет лишь как закрепление материала. В информатике это, увы, или к счастью, невозможно. Сильные внутри- предметные связи между различными темами не позволяют «выпрямить материал», как это имеет место, например, при изучении истории, литературы и некоторых других предметов.

В информатике нельзя некоторые темы (например, оператор цикла) изучить сразу полностью, за один или несколько подряд идущих уроков. Особенно это касается раздела «Основы алгоритмизации и программирования». Поэтому еще академиком А. П. Ершовым была предложена реализация принципа последовательности в форме цикличности. Это означает, что сначала изучается некоторый элемент в простейшем, наиболее распространенном на практике виде. Затем через определённый промежуток времени (может быть месяц или даже год и больше) это понятие повторяется, обогащаясь новыми возможностями, дополнительными особенностями. Если для других дисциплин это желательный путь, то для информатики многие разделы или темы по-другому изучить просто невозможно.

Особенностью методики преподавания информатики является динамический, изменяющийся характер самой информатики и как науки, и как учебного предмета, ее нестабильность, постоянное развитие и совершенствование как технических, так и особенно программных средств. В этих условиях вынужденным и плодотворным решением является максимальная опора на результаты общей дидактики, на конкретные методики близких дисциплин – математики и физики. Необходимо руководствоваться принципом единого классического образования школьников в области информатики, не зависящего, или в малой степени зависящего от типов компьютеров и программного обеспечения. Надо больше внимания уделять наиболее общим,

фундаментальным знаниям. Следует по возможности минимизировать или полностью избегать машинно-зависимых знаний и умений, которые могут оказаться бесполезными, а, возможно, и вредными, при работе на другом типе компьютера, с другой операционной системой или другой версией языка.

Основная особенность этой дисциплины – связь предмета с использованием компьютера, который обладает несравненно большей «самостоятельностью», чем любой другой прибор, например, в физике. Общение с компьютером требует развития особых черт мышления и поведения, адекватных методов обучения и воспитания. Достаточно сказать, что цена ошибки в обычном тексте и в программе для компьютера различна. Компьютер является таким средством обучения, мощность, эффективность и дидактические возможности которого потенциально превосходят то, что доступно иным средствам обучения. Это средство не может не вести к обновлению методик преподавания и других дисциплин.

При изучении методики преподавания информатики на занятиях распространены основные три вида деятельности:

роль учащегося – для лучшего понимания материала «изнутри» с позиции ученика и освоения тех учебников, которые появились в школе после поступления студента в учреждение высшего образования;

роль учителя – разработка инструкций, заданий, вопросов для контроля и других материалов для учащегося, управление с рабочего места для преподавателя работой класса;

роль «методиста-предметника» – разработка методических материалов, прежде всего, для себя как учителя, и фактически и для других учителей.

Формы и методы обучения зависят от изучаемых разделов, на которые разделен учебный материал. Теоретические особенности материала можно изучать в форме лекций, бесед, «круглых» столов. Структуру и содержание учебных пособий и учебников предлагается изучать кроме лекций в форме подготовки и обсуждения на семинарских занятиях докладов и рефератов.

Всё многообразие тем школьного предмета «Информатика» можно сгруппировать в отдельные блоки. Примерно третья часть всего отведённого на информатику времени занимают основы алгоритмизации и программирования, которые изучаются все шесть лет. Поэтому далее будут рассмотрены особенности и методика изучения этого раздела, самого сложного для школьников и учителей и важного с точки зрения развития мышления. Наряду с методическими вопросами определенное внимание уделяется здесь более глубокому изучению программирования. Этим самым параллельно с методикой расширяются знания и умения студентов в области современных информационных технологий. Этот раздел изучается в форме лекций и выполнения индивидуальных заданий по разработке программ.

При рассмотрении методики преподавания остальных тем материал предлагается изучать в форме лекций, а также в виде разработки и обсуждения плана-конспекта проведения нескольких уроков по конкретной теме.

При проведении занятий наряду с традиционными методами обучения следует использовать и инновационные методы: лекцию-визуализацию, работу с электронным дидактическим комплексом, видеотренинг, дистанционные занятия, работу с тренажером, компьютерное тестирование. Необходимо также разумно сочетать групповую и индивидуальную работу.

Содержание и формы видов контроля студентов разрабатываются соответствующими учреждениями образования. На основании бюджета времени в соответствии с образовательным стандартом, учебным планом учебной дисциплины устанавливаются виды, объем и содержание заданий по самостоятельной работе студентов. При проведении самостоятельной работы используется электронный вариант лекций, лабораторных занятий и другие материалы.

Для оценки качества самостоятельной работы осуществляется контроль за её выполнением. Формы контроля устанавливаются учреждением высшего образования: собеседование, защита индивидуальных заданий, коллоквиум, контрольная работа, тестирование, проверка рефератов, зачет (итоговый или по отдельной теме), устный и/или письменный экзамен.

При изучении частной методики с целью активизации самостоятельной работы студентов рекомендуется использовать метод проектов, что позволяет реализовать индивидуальный подход к обучению. В ходе работы над проектом студенты активно работают с различными источниками и системами обработки информации. В результате каждый студент по определенной теме создает материалы в виде презентаций, конспектов проведения уроков и других форм. Такая организация учебного процесса способствует развитию как информационной, так и профессионально-методической компетентности будущего учителя информатики.

1.4. Цели преподавания информатики

1.4.1. Значение предмета

Рассмотрим, что означает значение термина школьная информатика. Как известно, одной из областей человеческой деятельности, испытывающей активное влияние информатики, является образование. Ветвь информатики, обслуживающая проблемы учреждений общего среднего образования, получила название школьной информатики. Она занимается исследованием и разработкой программного, технического, учебно-методического и организационного обеспечения применения электронно-вычислительной техники в школьном учебном процессе. Программное (или математическое) обеспечение поддерживает информационную, управляющую и обучающую системы средней школы. Оно включает в себя программистские средства для проектирования и сопровождения таких систем, а также средства общения с ними, ориентированные на школьников, учителей и работников аппарата управления. В области технического обеспечения школьная информатика имеет своей целью экономически обосновать выбор технических средств для сопровождения

учебно-воспитательного процесса школы, определить параметры оборудования школьных компьютерных классов. Учебно-методическое обеспечение школьной информатики состоит в разработке учебных программ, методических пособий и учебников по школьному предмету информатики и другим предметам, при преподавании которых планируется использование средств информатики. К проблемам организационного обеспечения можно отнести следующие: организационно-технические мероприятия по обеспечению и последующему сопровождению технических азов школьной информатики; организация разработки, тиражирования и доставки педагогических программных средств. К организационному обеспечению можно отнести также и подготовку и переподготовку кадров школьных учителей, способных нести в школу информатику как новую научную дисциплину, как инструмент совершенствования преподавания других школьных предметов, как стиль мышления.

Школьный учебный предмет информатики не может включать всего того многообразия сведений, которые составляют содержание быстро развивающейся науки информатики. В то же время этот предмет, выполняя общеобразовательные функции, должен отражать в себе наиболее важные, фундаментальные понятия и сведения, раскрывающие существо науки, вооружать учащихся знаниями, умениями и навыками, необходимыми для изучения основ других наук и будущей жизнедеятельности в современном информационном обществе.

На процесс формирования школьного предмета информатики сказывается чрезвычайно малая временная дистанция между возникновением информатики как самостоятельной отрасли науки и включением в практику общеобразовательной школы соответствующего ей нового учебного предмета. По этой причине определение его содержания является непростой задачей. Более того, долгое время не являлось абсолютно бесспорным, как в целом изучать информатику в общеобразовательной школе — в отдельном предмете или целесообразнее рассредоточить учебный материал по информатике среди ряда учебных дисциплин. Вопрос заключался в следующем: чего должно быть больше — того, что должно составить отдельный предмет, или того, что может или должно быть неразрывно связано с содержанием и методикой изучения остальных школьных предметов. Как известно, на данный момент остановились на следующей дидактической формуле: всякий базовый компонент общего образования, каким является и информатика, включается в содержание образования двояко — в виде отдельного учебного предмета и в виде «вкраплений» в другие учебные предметы.

Школьная информатика должна выполнять роль интегрирующей дисциплины. Если её рассматривать просто как локальный предмет, не связанный с другими, если её изолировать в кабинете вычислительной техники, то такой путь является тупиковым, сковывающим творчество и учителей, и школьников.

Чтобы конкретизировать цели обучения, рассмотрим возможные уровни работы любого человека, не обязательно преподавателя, с компьютером.

1. Пассивный пользователь, не работая непосредственно с электронно-вычислительной машиной, пользуется компьютерными «благами»: получает в банкоматах зарплату, покупает билет на поезд, осуществляет поиск информации в Интернете, пользуется электронной почтой и т.д. Минимальные сведения о компьютерах получает из средств массовой информации. Профессиональные навыки работы на ней отсутствуют.

2. Активный (параметрический) пользователь работает с готовыми программами, подставляя в них свои параметры: добавляет базы данных, осуществляет её корректировку и поиск нужной информации и т.д. Он ориентируется в типах данных, алгоритмов, умеет писать, рисовать с помощью машины. Такой пользователь способен длительное время обходиться без помощи программиста. С позиции школьной информатики это наиболее массовая фигура в настоящем и ближайшем будущем.

3. Программирующий пользователь способен внести небольшие изменения в готовую программу. Например, он может вставить свою формулу в готовую универсальную программу построения графика функции. Он может программировать небольшие задачи в простых средах (например формулы в электронных таблицах *Excel*), способен грамотно поставить задачу программисту, выбрать необходимое готовое программное обеспечение для решения своих задач.

4. «Парапрограммист» («настройщик») работает на языках сверхвысокого уровня: в системах управления баз данных, электронных таблицах и др. Основное его умение: с одной стороны формализовать прикладные задачи пользователя и довести их до состояния работающей программы, и наоборот, видеть конкретные применения готовых программ. Он в состоянии настроить готовые программные продукты общего назначения на конкретные нужды пользователя. Такой специалист перебрасывает мостик между реальностью и компьютером, что требует системно-комбинаторного, прикладного образа мышления. От него в конечном счёте зависит, дойдет ли разработка программиста до практического применения, сможет ли пользователь с ней работать. Такой «настройщик» пока необходим, так как рост дружественности программ происходит в первую очередь по отношению к программистам.

5. Программист разрабатывает средства для «парапрограммистов» и для пользователей. То есть предполагается, что он разрабатывает, прежде всего, общее программное обеспечение.

6. Системный программист обеспечивает эффективность работы на всех предыдущих уровнях.

С позиций школьной информатики и современного её положения видно, что наиболее массовая фигура в настоящем и ближайшем будущем — параметрический пользователь. Но эффективность его деятельности резко возрастает, если он подтягивается до уровня программирующего пользователя, так как уменьшается его зависимость от программистов. Подготовка всех

учащихся до этого уровня — посильная для школы и социально необходимая задача.

1.4.2. Цели преподавания школьной информатики

Определение целей преподавания конкретного предмета — наиболее трудная часть практической методики. Цели обучения в общей дидактике рассматриваются в единстве образования, развития и воспитания. Информатика подчеркивает и практический аспект — подготовку молодого человека к полноценной жизни в компьютеризованном обществе. Общие цели обучения информатике определяются с учетом ее особенностей как науки, ее роли и места в системе наук, в жизни современного общества. Рассмотрим, как общие цели, характерные для школы в целом, могут быть отнесены к образованию школьников в области информатики. Среди основных целей выделяют следующие:

1. Образовательная и развивающая цель;
2. Практическая цель;
3. Воспитательная цель.

Образовательная и развивающая цель обучения информатике в школе — дать каждому школьнику фундаментальные начальные знания науки информатики и на этой основе раскрыть учащимся значение информационных процессов, информационных технологий и вычислительной техники в формировании научной картины мира и в развитии современного общества. Изучение этого предмета призвано также обеспечить учащихся базовыми умениями и навыками, которые необходимы для прочного и сознательного усвоения не только информатики, но и основ других наук, изучаемых в школе. Все это должно также существенно влиять на общее умственное развитие учащихся, развитие их мышления и творческих способностей. Общее образование при изучении информатики заключается в формировании компьютерного мировоззрения, понимании принципов работы, возможностей и ограничений электронно-вычислительной техники и достигается лишь на основе системных знаний, выводящих за рамки прагматичного, потребительского подхода к электронно-вычислительным машинам. Его признаки следующие:

1. Понимание общих принципов работы электронной техники;
2. Представление о сущности деятельности программиста и других специалистов в области информационных технологий;
3. Знания о типах информации и способах её обработки;
4. Представление об алгоритмах, их типах, способах их записи, в том числе о программах;
5. Знания о возможностях и ограничениях автоматизации мышления.

Цели развития в основном сводятся к формированию двух взаимодополняющих стилей мышления: логико-алгоритмического и системно-комбинаторного. Логико-алгоритмическое мышление проявляется в умении

строить логические утверждения о свойствах данных и запросы к поисковым системам, мыслить индуктивно и дедуктивно при анализе своих затруднений, формализовать свои намерения вплоть до записи на алгоритмическом языке или языке программирования. Признаками системно - комбинаторного мышления являются видение предметов и явлений в целостности, взаимосвязях, умение строить несколько взаимодополняющих точек зрения на одну и ту же проблему, умение комбинировать понятийные и орудийные средства из различных дисциплин при построении моделей. Например, с точки зрения алгебры функция есть соответствие, с точки зрения геометрии – кривая, а с точки зрения информатики – алгоритм вычисления результатов по заданным аргументам. Особенностью системно-комбинаторного мышления является также и то, что работа с компьютером — это своеобразная языковая деятельность. Как пользователи, так и особенно программисты излагают свои мысли и намерения в формализованном виде, оперируя правилами специальных языков. Заметим, что системно-комбинаторный стиль мышления, к сожалению, в основном не используется в практике средней и высшей школы. Предметы разделены на высокие и низкие, теоретические и практические. Не всегда видна связь между ними. Поэтому и студенту, и школьнику сквозь обилие частных не видно целостных знаний.

При использовании компьютера, особенно при разработке программ, у школьника развивается умение и склонность к наблюдению за собственным мышлением. В компьютерной деятельности, как ни в одной другой, можно проследить за следующей цепочкой: «что я хотел (алгоритм) – что я сделал (текст программы) – что у меня получилось и почему (результат выполнения программы)». Компьютер овеществляет мышление, делает наглядными наши намерения.

Практическая цель школьной информатики — внести вклад в трудовую и технологическую подготовку учащихся, т. е. вооружить их теми знаниями, умениями и навыками, которые могли бы обеспечить подготовку к трудовой деятельности после окончания школы. Это означает, что школьный курс информатики должен быть практически направленным — обучать школьника работе на компьютере и использованию программных средств. В профессиональной ориентации предмет информатики должен давать учащимся сведения о профессиях, непосредственно связанных с электронно-вычислительной техникой, а также различными приложениями изучаемых в школе предметов, опирающихся на использование этой техники. Практические цели предусматривают также и бытовой аспект — готовить молодых людей к грамотному использованию компьютерной техники в повседневной жизни и быту. Таким образом, прикладной компонент образования проявляется в умении извлекать практическую пользу из диалога с электронно-вычислительными машинами (запросы к поисковым системам, Интернет, небольшие расчёты и т.п.), что определяет уровень и качество жизни.

Воспитательная цель рассматриваемого предмета обеспечивается, прежде всего, тем мощным воздействием на ученика, которое оказывает осознание

возможностей и роли вычислительной техники в развитии общества и цивилизации в целом. При изучении информатики формируется культура умственного труда и такие важные качества, как умение планировать свою работу и рационально её выполнять, настойчивость и целеустремленность, творческая активность и самостоятельность, ответственность и трудолюбие, чёткость и лаконичность мышления, изложения и написания и другие полезные характеристики личности. Сами по себе эти и некоторые другие черты личности не сформируются, требуется с самого начала кропотливая работа учителя в этом направлении.

В результате реализации целей воспитания формируются следующие черты и качества личности:

1. Объективное отношение к результатам компьютерных вычислений, т. е. критичность и самокритичность в оценке своих способностей;

2. Бережное отношение к технике и к информации, как к своей, так и чужой, этическое, нравственное неприятие компьютерного вандализма и создания вредоносных программ и вирусов;

3. Личная ответственность за результаты своей работы на компьютере, за возможные свои ошибки, а также за решения, принимаемые на основе компьютерных данных;

4. Потребность и умение работать в коллективе при решении больших задач, забота о пользователе продуктов своего труда;

Названные выше основные цели обучения информатике взаимосвязаны между собой. Ни одна из них не может быть достигнута изолированно друг от друга.

1.4.3. Компьютерная грамотность и образованность

В соответствии с первой программой предмета, школьные учителя должны были сформировать всеобщую компьютерную грамотность старшеклассникам. Академик А. П. Ершов ввёл понятие компьютерной грамотности как владение навыками решения задач с помощью вычислительной техники, умение планировать действия и предвидеть их последствия, понимание основных идей информатики, представление о роли информационных технологий в жизни общества. При этом он подчеркивал, что основы компьютерной грамотности учащихся перерастут в информационную культуру общества.

Впервые на нормативном уровне понятие об информационной культуре учащихся было введено в 1986 году во второй программе курса основ информатики и вычислительной техники. Оно являлось расширением понятия компьютерной грамотности, введённое А. П. Ершовым. Он считал, что информационную культуру развивает курс школьной информатики главным образом. Этот курс должен формировать следующее:

Навыки грамотной постановки задач и формализованного их описания для решения с помощью электронно-вычислительных машин;

Знания основных алгоритмических структур и умение применять их для разработки алгоритмов решения задач;

Элементарные навыки составления программ для электронно-вычислительных машин на одном из языков программирования высокого уровня.

Понимание устройства и функционирования современной вычислительной техники.

Навыки квалифицированного использования основных типов информационных систем и пакетов прикладных программ общего назначения.

На самом деле ещё в первой программе было введено такое понятие, как компьютерная грамотность учащихся. Это понятие формировалось вместе с новым предметом. Если сравнивать с обычной грамотностью в привычном понимании, то под компьютерной грамотностью можно понимать умение считать, читать, писать, рисовать, искать информацию с помощью электронно-вычислительных машин. Такое определение было дано в одном из первых учебников по школьной информатике. Признаком высокой и сформировавшейся грамотности является самостоятельность и эффективность работы с применением электронно-вычислительных машин. Это первая характеристика качества обучения школьника информатике.

В настоящее время можно выделить следующие группы компонентов компьютерной грамотности школьников.

Понятие об алгоритме, его свойствах, способах описания. Представление о программе, как средстве и методе описания алгоритма для электронно-вычислительных машин, знание основ программирования на одном из языков. С одной стороны, подготовка программистов не является целью общеобразовательной школы. Тем не менее, учащиеся должны понимать основные принципы алгоритмизации и программирования и приобрести начальные навыки составления и отладки самостоятельных программ. Это основывается на алгоритмической культуре, которая может быть сформирована на простых и наглядных средствах. Сначала это были так называемые бытовые алгоритмы, а позже использовались чертежник, робот и другие средства.

Принцип действия и устройство ЭВМ и ее основных элементов. Здесь можно выделить две основные составляющие: структура персонального компьютера и функции его основных устройств (физические основы и принципы действия основных элементов компьютера).

Практические навыки работы на электронно-вычислительных машинах. Это есть умение грамотного обращения с электронно-вычислительными устройствами, а именно: умение запустить и правильно выключить компьютер, умение запустить готовые программы, овладение клавиатурой, печатью и другими устройствами. Сюда же позже были отнесены и навыки работы с простейшими сервисными программами, например, с текстовым и графическим редактором, электронными таблицами, системами управления базами данных, игровыми и обучающими программами, а в последние годы и электронной почтой и ресурсами сети Интернет.

Области применения и роль компьютеров в производстве и других отраслях жизнедеятельности человека. Формирование этой составляющей компьютерной грамотности не является задачей исключительно курса информатики. Эти вопросы целесообразно по возможности раскрывать учащимся в процессе практического использования компьютера для решения различных задач в других учебных предметах. Школьный компьютер можно использовать для вычислительных работ по математике, физике, химии, анализе данных учебного эксперимента при проведении лабораторных работ по этим предметам. В гуманитарных предметах персональный компьютер можно использовать для разработки информационной системы по географии, истории и другим гуманитарным школьным предметам.

Анализ перечисленных нами компонентов показывает, что понятие компьютерной грамотности явилось результатом расширения понятия алгоритмической культуры учащихся путем добавления компонентов, связанных с устройством и использованием электронно-вычислительной техники. Зародившись на начальном этапе преподавания информатики, понятие компьютерной грамотности и в настоящее время активно и успешно работает в методике этого предмета. Сокращённо изложенную выше структуру компьютерной грамотности можно обозначить совокупностью четырёх ключевых слов: программирование, устройство, общение и применение. Усиленное акцентирование внимания на том или другом из них может приводить к изменению цели преподавания предмета. Если, например, преимущество отдать компоненту общения, то курс информатики станет пользовательским. Если доминировать начнет понятие программирования, то цели курса сведутся к подготовке программистов. Эти два направления в смысле целей и содержания информатики долго вели между собой дискуссию о степени важности той или иной ветви. Золотая середина заключается в разумном использовании этих двух ветвей.

Вторая характеристика качества образования школьника по информатике, а, значит, и одна из ее целей — информационная культура. Всего лишь один год, прошедший со времени публикации первой программы показал, что цели преподавания информатики в школе не могут ограничиваться только рамками компьютерной грамотности и что в ближайшей перспективе потребуется развитие и расширение самих целей. Наряду с понятием компьютерной грамотности в новой программе 1986 года впервые на нормативном уровне появляется новое понятие: информационная культура учащихся. Оно было образовано путем добавления к компьютерной грамотности новых компонентов, относящихся в основном к моделированию. В курсе информатики необходимо было сформировать навыки формализованного описания поставленных задач, элементарные знания о методах математического моделирования поставленных задач и умение строить простую математическую модель. Кроме этого, по модели надо было уметь составить алгоритм и реализовать его в виде программы для электронной техники. В информационной культуре были также расширены некоторые прежние компоненты понятия компьютерной грамотности.

Например, компонент, отождествляемый в прежней редакции с развитием упрощённых навыков общения с компьютером, в новой системе целей связывается уже с навыками квалифицированного использования основных типов современных информационных систем и пониманием основных принципов, лежащих в основе их функционирования.

В учебном пособии под редакцией Бочкина А. И. под информационной культурой понимается, прежде всего, этика использования компьютера в контексте общечеловеческих ценностей. Этот термин рассматривается подобно религиозным заповедям, которые веками регулировали нормальное человеческое поведение, а именно:

«Не убий» чужую информацию случайно или умышленно. Отсюда неприятие создания вредоносного программного обеспечения и небрежности;

«Не укради» чужую программу или данные. Минимальное требование культуры здесь — не извлекать для себя прибыль из чужого труда и бесплатно полученное бесплатно же и передавать, причем обязательно с согласия автора. Копирование без разрешения автора и тем более взлом защиты от копирования есть присвоение чужой собственности;

«Не сотвори себе кумира», что означает отказ от компьютерного фанатизма, от излишнего заикливания на программах, от попыток сужения реального мира до компьютерной среды. Особенно это актуально в связи с широким развитием социальных сетей;

«Не лжесвидетельствуй», то есть не обсуждай те программы и данные, которые недостаточно хорошо знаешь.

«Возлюби ближнего как самого себя»: помогай товарищам и коллегам, не занимай лишнего места в общей памяти, удаляй ненужные устаревшие файлы и их копии.

В информационной культуре есть и эстетическое содержание – умение видеть, ценить и создавать красивое: оригинальный, понятный, эффективный алгоритм, красивое оформление сценария работы программы и её результатов (окна, кнопки и другие элементы экрана, использование графики, цветов и палитры). Частью информационной культуры является также самодисциплина и аккуратность: наличие комментариев, структурная запись текстов алгоритмов, стройная система в обозначениях, продуманная организация хранения информации во внешней памяти и другое.

В отличие от российских авторов учебного пособия [13] в книге [14] Бочкин А.И. выделяет еще одну характеристику качества обучения информатике. Он считает, что сложилась традиция трактовать КГ излишне широко, расширяя смысл вплоть до системной грамотности. Видимо, нужна иная, более точная категория — компьютерная образованность. Её основные признаки вытекают из обычных представлений об образованном человеке, взятых в контексте информатики. К ним можно отнести:

регулярное чтение литературы по информатике;

широкий кругозор в компьютерной области, понимание возможностей и ограничений электронно-вычислительной техники;

ориентирование в многообразии программных средств: знать их назначение, качественные характеристики, умение выбрать оптимальные программные средства для конкретной работы, наличие и ведение собственной библиотеки программных средств.

Школа не может обеспечить полностью такую образованность, но заложить основы и сформировать потребность в ней — одна из главных её задач. Таким образом, из выше сказанного можно сделать следующие выводы.

Цели практического образования служат формированию компьютерной грамотности. Цели общего образования и умственного развития связаны с компьютерной образованностью. Цели воспитания служат формированию информационной культуры.

Заметим, что между уровнем освоения работы на электронно-вычислительных машинах, грамотностью, образованностью и культурой не всегда существует однозначная зависимость. Например, пользователь, не умеющий программировать, может быть весьма эрудированным, и наоборот, фанатичный программист может ничего не знать кроме, скажем, своего языка или используемой системы. С этой точки зрения создатель вирусов грамотен и, возможно, образован, но культуры, разумеется, у него нет. Если рост уровня компетентности в информатике не сопровождается повышением уровня культуры, последствия могут быть весьма негативными. Это особенно актуально в связи с развитием и широким использованием компьютерных сетей, включая и Интернет. Поэтому все три указанные цели компьютерного обучения требуют определенных усилий педагогов и не достигаются автоматически при простом усвоении материала.

Проведенный анализ системы целей позволяет предложить ответы на некоторые дискуссионные вопросы преподавания информатики:

Нужно ли всем школьникам изучать программирование? Да, но лишь в той мере, в какой это обеспечивает подготовку программирующих пользователей, формирует логико-алгоритмическое мышление. Чрезмерное увлечение этим чревато потерей времени, нужного для повышения уровня компьютерной образованности. Признаком профессионализма для программиста является избегание программирования в случаях, когда можно использовать готовые программы. С другой стороны, квалифицированный пользователь не должен бояться программирования небольших и нестандартных задач.

Какой язык изучать или использовать? Это не важно. Главное, чтобы с его помощью можно было изучить основные типы алгоритмов, основы современных методов и технологий программирования. Не обязательно использовать простую систему программирования. Можно, например, выбрать профессиональный сложный язык «С++», но некоторые его достаточно непростые возможности не изучать.

Нужно ли будущему пользователю знать устройство электронно-вычислительных машин? Да, в той мере, в какой это способствует компьютерной образованности и успеху в практическом использовании компьютера.

Сложным и различным в плане изучения предметов является следующий вопрос: «Нужно ли в информатике заниматься моделированием?» С одной стороны, не так важно, где моделировать, на уроке информатики или в других предметах. Именно через моделирование формируется системно-комбинаторное мышление, умение решать реальные задачи из разных областей, формируется научная картина мира. С другой стороны, ввиду сложности этой темы, вокруг которой было очень много споров и дискуссий, начиная с 2009 года, она изучается в школьном курсе информатики лишь в классах с углублённым изучением информатики.

1.5. Принципы дидактики и преподавание информатики

1.5.1. Принцип научности

При отборе содержания, средств и методов преподавания курса на помощь учителя информатики должны прийти общие принципы дидактики. Рассмотрим эти принципы. Начнём с принципа **научности**.

Затронем важный с точки зрения дидактики вопрос об объекте и предмете науки информатики. Объект — это область действительности, на которую направлена деятельность исследователя, а предмет — это посредствующее звено между субъектом и объектом исследования. Другими словами, представители разных наук видят один и тот же объект по-разному, в свете разных задач, в разных системах понятий, выделяют в объекте разные стороны, связи и отношения. Например, обучение по-разному рассматривают методист, психолог, кибернетик и другие специалисты.

Применительно к информатике её предмет образуется на основе широких областей своих приложений, а объект — на основе общих закономерностей, свойственных любым информационным процессам в природе и обществе. Предмет информатики определяется многообразием её приложений, которые охватывают различные виды деятельности: производство, управление, науку, образование, медицину, торговлю и другие. Информатика изучает то общее, что свойственно всем многочисленным разновидностям конкретных информационных процессов, которые и являются объектом информатики.

Принцип научности требует, чтобы в содержании образования нашли отражение новейшие достижения соответствующей области знаний с адаптацией на познавательные возможности учащихся. Эта задача упрощается тем, что пока нет деления на высшую информатику и низшую, как это имеет место, например, в математике. Любое понятие из «большой» информатики находит свои аналогии в школьной информатике.

Фундаментальными являются понятия «информация», «алгоритм», «исполнитель». Исполнитель выполняет несколько функций.

1. Дидактическое средство для придания процессу исполнения алгоритмов наглядности (робот, чертёжник и др.);

2. Понятие, позволяющее с единых позиций трактовать многие вопросы: Робот — исполнитель над графикой, редактор — исполнитель над текстами,

операционная система — исполнитель для файлов, принтер – исполнитель для листа бумаги и другие.

Компьютерную модель всякого исполнителя можно понимать в терминах объектно-ориентированного программирования как модуль или объект.

Научность обучения подразумевает также современность методов обучения, что применительно к информатике означает, прежде всего, моделирование в широком смысле, а также исследовательскую деятельность учащегося.

1.5.2. Сознательность усвоения и деятельности, доступность и наглядность, активность и самостоятельность

В традиционном смысле **сознательность** — это полное понимание учащимся содержания и средств деятельности. Но компьютер, будучи сложным устройством, заранее вынуждает ограничивать эту сознательность целями обучения. Можно ли за ограниченное время полно, детально рассказать обо всех процессах, происходящих в компьютере, например, при нажатии клавиши «Enter»? Всё это можно и не знать. Конструктор электронно-вычислительной машины, программист и пользователь имеют различные точки зрения на такое событие. Надо сформировать у учащегося взаимодополняющие точки зрения на подобные ситуации, что и даёт многостороннее знание.

Здесь решающее значение имеет уровень знаний учителя и, самое главное, умение отобрать, ограничить материал.

Принцип доступности реализуется через выделение уровней обучения и работы за компьютером. Например, самый низкий уровень: простое использование готового программного обеспечения. Это доступно всем учащимся. В разделе «алгоритмизация и программирование» предлагаемые задачи можно разделить на несколько уровней, чтобы школьник мог выбрать доступную для себя задачу (простую на 5 баллов, среднего уровня на 7 баллов, сложную задачу на 9 баллов). На 10 баллов надо предложить нестандартную оригинальную задачу. Наглядность достигается с помощью:

блок-схем алгоритмов;

структурной (с отступами) записи текстов программ (алгоритмов);

использование цветов;

демонстрации готовой программы, процесса ее выполнения.

Всё это сделает процесс более наглядным и простым.

Активность учащегося при изучении информатики имеет следующую особенность. Если при изучении других дисциплин педагог работает в прямом контакте с учащимися, видит их реакцию, сам реагирует, то здесь возможна работа ученика один на один с компьютером. В информатике активность учащегося является не только целью, но и необходимым условием успешности обучения.

Формы проявления активности различны, например, самоконтроль, контроль над работой товарища, оказание реальной ему помощи, изменение готовых и разработка собственных алгоритмов и программ. Активность следует

из интереса к предмету, но учителю важно чётко сформулировать, что является контролируемым результатом обучения, то есть, что нужно «сдать». Активизировать работу, особенно в начале обучения, можно, используя работу учащихся по два или более человека за одним компьютером, даже если их достаточное количество в классе. В таком случае уменьшается неуверенность, возникает диалог, происходит взаимное обучение. Но при этом возникают вопросы. Вместе дать возможность работать двум сильным ученикам или нет? Хорошо успевающий должен быть в паре со слабым учеником или нет? Оба ученика должны работать в паре с невысокими способностями или нет?

Самостоятельность учащегося, как цель и условие успешного изучения информатики, следует за активностью. Возможные этапы нарастания самостоятельности: от полного управления учителем, через дозированную помощь к самоуправлению познавательной деятельности с помощью компьютера. Полностью самостоятельность реализуется при переходе к творческой деятельности, при выполнении индивидуальных заданий. Когда ученик обращается за помощью к учителю, к товарищу — это, понятно, проявление не самостоятельности, а активности.

Значение самостоятельности очень велико: она ведёт к большей результативности обучения, учит находить выходы из затруднительных, иногда, казалось бы, тупиковых ситуаций. Особенно это касается отладки программ, поиска ошибок. Самостоятельность учит также искать нужную литературу, пользоваться ею, а также учит использовать компьютерные средства помощи.

Прочность знаний тесно связана с их **системностью**, основанной на поиске, построении и учете внутрипредметных и межпредметных связей и ассоциаций. Обычная, например, для зоологии структура изучаемого предмета в виде дерева обязательна: в информатике должна быть дополнена «паутиной» связей между листьями-понятиями, их взаимным обогащением в комбинациях.

Содержание информатики и как науки, и как учебного предмета в виде одного дерева представить невозможно. Скорее всего, это лес с переплетёнными кронами, растущий из таких фундаментальных для этой дисциплины понятий, как «информация», «алгоритм», «исполнитель» и другими.

1.5.3. Другие принципы дидактики

Индивидуализация и коллективность обучения дополняют друг друга, особенно в информатике. В этом отношении компьютер — дидактически двойственный инструмент. Распространяя обучающие или готовые программы, он способствует организации единообразной, фронтальной групповой деятельности, но способ работы учащегося с программой — все же «один на один», со своим индивидуальным темпом, своими путями преодоления трудностей.

Индивидуализация возможна:

1. Через выполнение индивидуальных, а не общих, одинаковых для всех, заданий, классифицированных по уровню сложности;
2. Через гибкую настройку обучающей программы (например, на тип мышления обучаемого);
3. Через освобождение времени педагога для индивидуальной работы при автоматизации рутинной части педагогического труда;

При работе учащихся вдвоём за компьютером, что может оказаться весьма полезным, могут сложиться устойчивые отношения типа «работник-указчик». Поэтому время от времени учащихся надо менять местами и ролями.

Эффективность предполагает оптимизацию усилий педагога и ученика для обеспечения наибольшего их полезного действия, отношение «результат/усилие». Это требует, прежде всего, отсутствия постороннего содержания в их деятельности. Например, блок-схемы, наглядные и удобные для малых задач, могут превратить информатику в черчение при более сложных алгоритмах. Паскаль и другие учебные языки, безусловно, эффективны для изучения алгоритмизации или программирования, но они мало приспособлены для вычисления выражений типа «693 разделить на 13». Здесь лучше использовать калькулятор.

При дефиците машинного времени, а также для сохранения зрения и здоровья, эффективность работы за дисплеем должна обеспечиваться предварительной подготовкой учащегося, изучением инструкций. Эффективным является непосредственное редактирование текстов программ, отладка программ и др. Эффективность работы за дисплеем также зависит от индивидуальных особенностей развития учащегося.

Путь от теории, от приобретения знаний до их применения, то есть до практики, в информатике очень короткий, короче даже чем, например, на уроках трудового обучения. Учащийся может решить задачу, полезную учителю, классу или школе. Например, он может создать небольшую базу данных, касающуюся успеваемости на уроках, или по школьному оборудованию, для библиотеки, и т.п.

Понятия **теории и практики** в информатике обнаруживают полную параллельность с общенаучными категориями. Как и в «большой» науке, теория есть средство прогноза, предсказания или объяснения свойств, поведения компьютерного мира, а практика является средством проверки теории и, с другой стороны, источником гипотез для неё.

Теоретическим методом выглядит доказательство алгоритма без вычислительной техники. Но такой способ приемлем для небольших по объёму несложных алгоритмов. Эффективнее, конечно, тестирование программы на практике с использованием компьютера. Но минимальная система тестов, которая охватывала бы наибольшее количество возможных ситуаций (ветвей программы) строится даже для конкретной задачи теоретически.

Другим аспектом является связь теории и практики при изучении непосредственно информатики. Как и в других науках, теория объясняет или предсказывает результат опыта (запуск алгоритмов на компьютере), а практика

(работа на компьютере) служит средством проверки теории и источником гипотез, например, о поведении программы. Эти два вида деятельности тесно переплетаются на уровне мышления учащегося.

Комбинированным методом работы является отладка программ. Здесь тесно переплетаются практическое проявление ошибки при компиляции и выполнении программ, теоретический анализ её причин, их поиск с привлечением определённых знаний из соответствующей области и практическая проверка на компьютере внесённых исправлений.

1.6. Методы и организационные формы обучения информатике

1.6.1. Особенность методов и их реализация на практике

Метод – это способ деятельности, направленный на достижения определенной цели. В общей дидактике, как известно, следует различать следующие понятия:

Учение – учебная деятельность учащегося, например, отладка программ самостоятельно;

Преподавание – деятельность учителя, например, разработка инструкций, текстов, заданий для индивидуальной работы и другое;

Обучение – их совместная деятельность, например, защита учеником индивидуальных заданий.

Особенность информатики в этом смысле заключается в следующем. Компьютер как посредник между учителем и учеником увеличивает объём относительно независимых видов деятельности учащегося и учителя и сокращает объём их совместной работы. Это связано с тем, что целью курса является в идеале независимость обучаемого при работе с вычислительной машиной от педагога, а затем и от программиста. Но это не означает выведение педагога из учебной деятельности. Его опыт и знания никогда не заменят никакие обучающие программы.

Как отмечалось раньше, учащийся, работающий за компьютером, более самостоятельно, имеет локальные собственные цели.

Метод преподавания в общем смысле можно понимать, как метод управления познавательной деятельностью учащегося;

Метод обучения можно понимать, как метод познания обучаемым действительности в специально созданной учебной ситуации.

Известно множество оснований для классификации методов обучения:

по содержанию обучения;

по способу восприятия информации: словесные, практические и др.;

по способу получения знаний (теория и(или) практика);

по способу реализации обратной связи (контроль учителя, самоконтроль, контроль с помощью компьютера).

Рассмотрим теперь **традиционные и словесно-фронтальные методы**. В такие методы входят рассказ, лекция, беседа и инструктаж. Рассмотрим их особенности.

Характерный **признак рассказа** – яркое, занимательное, эмоциональное повествование. Оно выполняется, как правило, в следующей последовательности: вступление, изложение, заключение. Знания при этом приобретаются теоретически, логика рассуждений здесь нестрогая с аналогиями и примерами. Обратная связь при этом осуществляется по степени внимания. Можно попросить учащегося воспроизвести услышанное от педагога. Основное назначение рассказа: передача конкретных сведений. Подходящие для такого метода темы: области применения вычислительной техники, история развития вычислительной техники и компьютерных технологий, вредоносное программное обеспечение и вирусы. Это можно поручить и «сильным» ученикам, так как рассказ имеет смысл использовать для изучения простых тем, которые носят информационный характер.

Основные черты **лекции**: объёмность материала, сложность, логичность, взаимосвязь отдельных вопросов. Поэтому критерий выбора тем для лекции: трудность в изучении, отсутствие хорошей, понятной и доступной для школьника литературы и другие. Поэтому подходящими темами могут быть, например, такие: устройство электронно-вычислительных машин, системы счисления и представления информации в памяти компьютера, вспомогательные алгоритмы. Примерами неудачных тем являются, например, следующие: клавиатура, подготовка и редактирование текстов в текстовом редакторе и др. Дидактическая функция аналогична, как и для рассказа: передача знаний и сведений. Логика рассуждений в основном дедуктивная, строгая, но могут использоваться и такие операции, как аналогия и сравнение. Обратная связь во время лекции обычно слабая и чаще всего отсроченная, так как из-за сложности материала надо дать время на обдумывание. Классический вид лекции как диктовка, пересказ основных положений, фактов для информатики имеет малую эффективность. Необходимо, чтобы лекция носила проблемный характер, по возможности использовала раздаточный материал (например, перечень и вид стандартных процедур и функций по определённой теме), компьютерные средства демонстрации.

Особенность **беседы** – системы управляющих вопросов, ведущих обучаемого к заранее намеченной преподавателем цели. Как правило, обсуждаются наиболее важные вопросы до и после беседы. Для беседы можно выбирать также темы, по которым учащиеся уже что-нибудь знают: начальные сведения по использованию компьютера, понятия алгоритма. В тоже время учащиеся получают новые знания, систематизируют то, что было им известно. Поэтому дидактической функцией беседы является и приобретение, и упорядочивание знаний.

Черты **инструктажа** следующие: краткость, повелительная форма. Логика почти не привлекается, развёрнутых рассуждений нет. Примеры тем: техника безопасности в компьютерном кабинете, начальные сведения о работе с клавиатурой, порядок работы со стандартным или другим программным обеспечением. Дидактическая функция заключается в усвоении сведений и некоторых стандартных способов действия. Инструктаж может сопровождаться

показом образца действия. Основная цель такого – подготовка к проведению занятий.

1.6.2. Мыслительные операции и работа с вычислительной техникой

Анализ и синтез в информатике имеют следующие специфические особенности. Например, целью анализа может быть выяснение причин ошибки. При этом процесс выполнения программы разбивается на шаги. Если поиск ошибки выполняется в диалоге с электронно-вычислительной машиной путём трассировки программы, то это «грубый» анализ. Если ошибка ищется за столом, то это мысленное выделение частей.

Примером сложного синтеза может являться создание учащимся модели компьютерной среды (например, электронные таблицы) на основе понимания отдельных команд и наблюдений за реакцией среды на эти команды.

Сравнение имеет место, когда сопоставляются, например, близкие элементы языка программирования (операторы «Если» и «Выбор», на «Pascal» операторы «while» и «repeat», функции и процедуры, статические и динамические массивы и т.п.). Эту же операцию можно использовать при изучении устройства электронной техники (сравнение оперативной и внешней памяти), какой-нибудь системы и т.п.

Сравнение есть достаточно мощный дидактический приём. С его помощью легче вводить новые понятия. С помощью этого метода сначала указываем на сходство нового элемента с уже изученным, а затем на различие. Используя этот прием, можно более объективно проверить знания учащихся. Одно дело, рассказать в отдельности, например, про массив, и отдельно про файл с типом на языке «Pascal». Учащийся должен более глубоко знать материал, если необходимо сравнить эти два типа данных: найти общие характеристики и различия, особенности по отношению друг к другу.

Классификация как мыслительная операция имеет место при освоении учащимся достаточно большого объёма материала. Особенно актуально это в настоящее время, когда необходимо осваивать большое количество стандартных средств (классов и их методов, самостоятельных функций, компонент и свойств в системах типа «Delphi» и т.п.). К сожалению, в книгах часто такого рода элементы описываются в алфавитном порядке. Такая классификация вполне уместна не при изучении, а при использовании такого пособия в качестве справочного при условии, что материал уже освоен. При изучении, особенно на начальном этапе, удобнее, если такие элементы будут классифицированы по назначению, по важности, по частоте использования на практике, или по другим критериям (например, строковые функции в языке C++ по типам возвращаемых значений).

Теперь рассмотрим особенности **обобщения, индукции и дедукции**. Ярким классическим приёмом обобщения в информатике является переход от представления об отдельных повторениях похожих действий к команде цикла.

Развёрнутой формой обобщения является индукция. Имеется в виду не математическая, а неполная индукция. Пример. Учащийся впервые сел за компьютер и в текстовом редакторе нажал клавишу «стрелка вправо». Если спросить, что произойдёт, если нажать эту клавишу, ответ будет однозначным и даже недоуменным. Но не принят во внимание случай, когда курсор уже находится у правого края. В этом примере ход рассуждений учащегося, скорее всего, неосознанный, представляет собой неполную индукцию, причем очень неосторожную.

Несмотря на, казалось бы, неправильный результат в этом примере, такая схема часто используется и ведёт к верным умозаключениям, которые подтверждаются практическими навыками по различным причинам.

Комбинаторно-неисчерпаемое множество возможных состояний компьютера устроено очень регулярно: однотипны ячейки памяти, схемы, регистры, структуры алгоритмов (так называемые базовые управляющие структуры: следование, ветвление и повторение).

Современные программные средства, особенно работающие в среде «*Windows*», всегда придерживаются определенных стандартов (меню, кнопки и другие элементы приложения, панель инструментов, строка состояний и т.д.). При разработке своих программ также рекомендуется соблюдать общепринятые требования.

Индукция проявляется и как обобщение команд: тело цикла получается из серий частных повторяющихся действий при конкретных значениях параметра.

Индуктивным является и умозаключение о правильности программы на основании конечного числа тестов. Несмотря на не полную строгость этого вывода, так программирует весь мир. Популярно в информатике вводить новые понятия, отталкиваясь от примеров. Приведем несколько имен (идентификаторов), например, в языке «*Pascal*»: «*MyMin*», «*R1*», «*r2*», «*Cout_of_Max*». И наоборот, запишем имена, которые не являются идентификаторами: «Число», «*2max*», «*R+1*», и др. Анализ этих примеров позволяет сформулировать общеизвестные правила построения идентификаторов. Следует индуктивное обобщение: считать сделанные утверждения истинными не для данных примеров, а вообще для всех имен.

Для успешности такого индуктивного подхода система примеров должна обладать рядом свойств:

Представительностью: примеры представляют наиболее важные случаи;

Минимальностью: примеры не должны повторять одинаковые правила;

Ортогональностью: примеры должны быть независимы;

Распространенностью;

В самом лучшем случае система примеров должна минимально представлять все частные случаи. Но не рекомендуется оглашать индуктивное заключение, например, «Ученик имеет плохие способности», если он не справился с некоторым заданием. И, наоборот, даже единственный, частный успех является основанием для высказывания гипотезы о том, что он вообще способный.

Роль дедукции в школьной информатике скромнее, чем индукции, в силу опережающей практики возможности тут же проверять верность умозаключения опытом, прямо на компьютере.

Дедуктивным является, например, поиск ошибки в программе, если она ищется пошаговым исполнением. Но и в этом случае заключительный этап понимания, «схватывания» ошибки происходит очень быстро, практически неосознанно. Есть все основания говорить в этом случае о так называемой «свёрнутой» дедукции, идущей от теста, текста программы, дидактического и ожидаемого результата.

Аналогия – это осторожная индукция. Схема рассуждений следующая. Пусть при обстоятельствах *B* имеет место факт *F*. Обстоятельства *B1* в каком-то смысле подобны обстоятельствам *B*. Значит, можно ожидать, что и при них будет иметь место факт *F*.

Конкретизируем это на таком примере. Пусть в некоторой системе сочетание клавиш (например, «*Ctrl+Del*») при редактировании удаляет строку. Пусть есть другая система той же фирмы. Тогда и в ней удаление строки должно выполняться аналогично. Тем самым умение переносится в близкую обстановку более обоснованно, чем при неполной индукции.

Операции **абстракции и конкретизации** связаны, прежде всего, с компьютерным моделированием. Исходная задача всегда ставится конкретно, в терминах области знания, где она возникла, и её перевод на язык информатики – самостоятельная и трудная проблема. Суть дела заключается в переводе с конкретного языка на абстрактный. Затем полученные результаты должны быть представлены пользователю или для себя на языке постановки задачи, т.е. выполнен обратный перевод.

Переводом задачи с формального языка на понятный, содержательный является написание комментариев, что является, в общем, не простой задачей. Это искусство своего рода, творчество, хотя и не очень сложное. Главное здесь, найти «рациональное зерно», чтобы было не много пояснений, но в то же время достаточно для понимания. Особенно это актуально для учебных программ и алгоритмов.

1.6.3. Организационные формы обучения информатике

Организационные формы обучения являются внешним, видимым проявлением структуры информационных связей между педагогом и учащимися.

В общей дидактике конкретные организационные формы обучения различаются по числу участников совместной деятельности (фронтальная, бригадная или звеньевая, индивидуальная, парная работа) и по роли участников учебного процесса, то есть в зависимости от того, кто управляет, учитель или учащиеся.

С древних времён начался монотонный отход от индивидуального обучения. Индивидуальное обучение стало расточительным по отношению к усилиям учителя. Традиционный учебник представлял собой первый шаг к синтезу фронтального и индивидуального обучения.

В информатике новым формообразующим элементом этой структуры является компьютер. С одной стороны, появление компьютера возрождает индивидуальные формы обучения, доступные ранее лишь для детей из высших классов. Благодаря обучающей программе или электронному учебнику ученик может воспринимать заложенные в них знания и опыт в своём индивидуальном темпе. Более того, всё чаще предлагается не жёсткий единообразный алгоритм обучения, а спектр вариантов обучения. Что и в каком порядке познавать, определяет учащийся в зависимости от способностей, уровня знания и других личных качеств.

С другой стороны, за счет распространения информации сохраняется и преимущество фронтальных форм: возможность учиться у лучших учителей.

Методика преподавания информатики указывает на две новые ситуации: работа ученика под управлением обучающей программы либо «обучение» компьютера учеником. И то и другое представляет собой самоуправление познавательной деятельностью при немедленном отражении её результатов на экране.

Организационные формы обучения определяются целями, содержанием и методами преподавания, которые тесно взаимосвязаны. На организационную форму влияет содержание обучения. Так как метод обучения определяется содержанием, то он принимает свою форму. Применительно к информатике демонстрация учителем образца деятельности за компьютером (метод обучения) с целью передачи опыта такой деятельности (содержание обучения) наибольшему числу учащихся приобретает адекватную, фронтальную форму занятий.

В информатике в связи с возможностью использования компьютера в пределах одного урока могут комбинироваться многие формы. Особенно это касается сдвоенных уроков, которые более эффективны, так как сокращаются потери времени на подготовку вычислительной техники и вхождение класса в работ

Рассмотрим **фронтальные формы работы**. Как и в других предметах, фронтальные формы применяются при усвоении всеми учащимися одного и того же содержания или вида деятельности. Примеры: лекция о системах счисления, семинар с выступлением учащегося о применении электронно-вычислительных машин в медицине, демонстрация нового типа алгоритма или программы (отлаженной первым из решивших задачу) на доске или на проекторе.

Достаточно сложной является фронтальная форма работы с использованием всеми учениками компьютеров. Особенно это касается сложных и не очень интересных тем. При этом надо сделать все возможное, чтобы учащиеся не перешли на игру, в интернет и т. п. Учитель должен решительно прервать начавшуюся индивидуальную самодеятельность и восстановить единое

состояние компьютерной среды на всех вычислительных машинах. Надо уметь вовремя перейти к парной или индивидуальной работе.

Обучение в составе бригады (звена) полезна с той точки зрения, что она при умелой организации может отражать реальное разделение труда в коллективе программистов, работающих над одной большой задачей. Типичным примером учебной задачи для использования бригадной формы может являться рисование в графическом режиме большой “картины” состоящей из несколько простых элементов (дом, деревья, солнышко т. д.). Сначала эти части рисуются каждым членом бригады отдельно в виде, например, подпрограмм, а затем собираются на один компьютер.

Основные преимущества бригадной формы: интенсивное взаимное обучение, помощь друг другу а, значит, своевременно убираются пробелы в знаниях. При этом должны быть и задачи разной сложности.

Парная работа на электронно-вычислительных машинах является разновидностью бригадной, и сформировалась, казалось бы, из-за нехватки компьютеров. Но было замечено, что и при достаточном их количестве такая форма может быть полезной в начале обучения при освоении или закреплении новой сложной темы.

Учащийся, работающий один за компьютером, может по разным причинам не обратиться за помощью к товарищу или педагогу. Когда за одним компьютером работают двое, то вероятность незнания одного и того же обоими сразу уменьшается. С другой стороны, в целом класс будет реже отвлекать его и будет возможность учителю больше внимания уделять слабым ученикам.

Заметим, что втроём работать за компьютером менее эффективно и нецелесообразно. Это может быть оправдано в следующем случае. Большой класс (или его половина) являются школьниками «тяжёлыми» в смысле успеваемости. Тогда имеет смысл уменьшить количество одновременно работающих компьютеров, которые используются при объяснении нового материала. Тогда учителю легче руководить синхронной, под диктовку, работой, например, на пяти, а не на десяти компьютерах, даже если их достаточное количество. Надо учитывать, что такая форма занятия очень сложная. Легче сначала объяснить без компьютера, если, конечно, это целесообразно и в принципе возможно, а затем воспроизвести на компьютере.

В некоторых случаях возможна **работа один на один с компьютером**. Отличие этой формы от классической самостоятельной в том, что в виде программы присутствует знание, обладающее собственной активностью. Надо учитывать, что поведение программы зависит от действий учащегося. В этом случае возобновляется фронтальное обучение, но с индивидуальными темпом и способами усвоения.

Для осмысления того, что происходило за компьютером, бывает полезно от него удалиться во время урока, особенно при появлении трудных моментов или неожиданных действий вычислительной техники или программ. **Работа без компьютера** позволяет сравнить полученные результаты с помощью логических действий самостоятельно.

Полезно также и отстранение от компьютера тех учащихся, которые не подготовились к занятию: не изучили инструкцию по выполнению работы в диалоговом режиме, не разработали блок-схему алгоритма или не написали программу, предлагавшуюся в качестве домашнего задания, в бумажном варианте.

Но, с другой стороны, такая форма не обязательно должна быть как наказание. Иногда надо объяснить, что это делается для его же блага (например, чтобы не потерял свой файл, чтобы надежно его сохранил).

Управление со стороны учителя очевидно при фронтальных формах. Как говорилось ранее, сложнее управлять индивидуальной деятельностью по понятным причинам: ситуация за каждым компьютером практически уникальна. Можно предложить следующие варианты выхода из этого:

Привлечь для помощи сильных учеников.

«Автоформализовать собственный педагогический опыт» в виде обучающих программ.

Использовать педагогические знания других учителей, не обязательно в виде обучающих программ.

Иногда **работа под руководством** товарища оказывается эффективнее, доступнее и понятнее, чем помощь учителя. Одна из причин этого в том, что обучаемый не боится спросить у товарища такое, что, по его мнению, спрашивать у учителя было бы стыдно. Другая причина в том, что некоторые учащиеся боятся спрашивать у учителя, так как, по их мнению, существует опасность получить плохую оценку.

Если возникнут общие для класса затруднения, то есть смысл выйти на несколько минут из класса. За это время школьники друг у друга могут выяснить все возникшие вопросы намного быстрее, чем по инструкции или с помощью учителя.

Лекция имеет два смысла: это и форма, и метод. Лекция всегда фронтальная. Из всего многообразия вопросов, связанных с лекцией, обратим внимание на следующее. Она может поддерживаться компьютером как средством наглядности и демонстрации и, если позволяет материальная база, проводится в дисплейном классе. Упражнение при этом выполняет учитель. При наличии подготовленных на компьютере конспектов у учащихся усиливается самоуправление познавательной деятельностью. Полезно использовать конспект, в котором слева отпечатан кратко основной материал, а справа оставлено место для комментариев учащегося, куда записываются пояснения, результаты компьютерных экспериментов и т.д.

Во время компьютерной лекции не следует увлекаться демонстрацией готовых программ. Дело в том, что при изучении алгоритмизации и программирования часто бывает полезным сам процесс разработки алгоритма и программы. При этом важно, как учитель думает и размышляет.

Семинар является переходной формой от фронтальной к индивидуальной работе и поэтому сохраняет своё значение и в изучении информатики.

Прежде чем использовать компьютер, необходимо вырабатывать ряд навыков и умений, так как, например, разрабатывать алгоритм или осваивать новую систему прямо за экраном тяжело и это могут делать не все. Работать без предварительного изучения инструкции расточительно по отношению к машинному времени.

Кроме этого, нужна адекватная форма работы для коллективного осмысления того, что сделано на компьютере, анализ того, что и почему получилось, и анализ типичных ошибок, основных затруднений и путей их преодоления. Это можно делать на семинаре в безмашинном варианте.

Лабораторная работа является основной формой работы в компьютерном классе. Все учащиеся одновременно работают на своих местах с программными средствами, переданными им учителем. Дидактическое назначение этих средств может быть различным. Например, с помощью обучающей программы можно освоить новый материал; а с помощью программы тренажера можно проработать материал, объяснённый учителем. Во время лабораторных работ можно проверить усвоение полученных знаний или операционных навыков, например, с помощью контролирующей программы. Такие занятия или их фрагменты можно также использовать для решения общих, наиболее типичных и сложных примеров по изучаемой теме.

В одних случаях действия школьников во время выполнения лабораторной работы могут быть синхронными, то есть все ученики (или их абсолютное большинство) выполняют одинаковые или похожие задания примерно с одинаковой скоростью. Если в классе есть очень способные ученики, или, наоборот, очень слабые, или пропустившие предыдущие занятия, то таким можно предложить другие задания. Поэтому не исключаются ситуации, когда школьники занимаются в различном темпе или с различными программными средствами.

Роль учителя во время фронтальной лабораторной работы есть наблюдение за работой учащихся и при необходимости оказание им оперативной помощи. При этом можно использовать локальную компьютерную сеть. Ситуация несколько меняется при использовании встроенной помощи, которая рассчитана на обучение (но не с самого начала!) в ходе работы. Учитель должен научить пользоваться такой помощью, чтобы быстро и, главное, разумно находить нужную информацию.

Более высокой формой работы по сравнению с фронтальными лабораторными работами является **индивидуальный практикум**. Учащиеся получают индивидуальные задания для самостоятельной работы. Если в средних классах (6–9) их можно выполнить в течение одного урока, то в 10–11 классах на их выполнение может понадобиться в зависимости от темы и сложности задания два и более урока. Не исключается выполнение части работы дома. Такое задание, как правило, выдаётся для отработки знаний и умений после изучения целого раздела или большой по объёму темы. Учащиеся сами или учитель решают, когда воспользоваться компьютером (в том числе и для поиска в сети), а когда поработать с книгой или обдумать некоторые вопросы без компьютера.

При этом учитель должен следить за тем, чтобы учитывались гигиенические требования к организации работы в кабинете вычислительной техники, чтобы время непрерывной работы за компьютером не превышало рекомендуемых норм. Во время практикума учитель наблюдает за ходом выполнения заданий, оказывая школьникам оперативную помощь. При необходимости приглашает всех к обсуждению общих вопросов, анализу наиболее характерных часто повторяющихся ошибок.

Отметим некоторые характерные черты индивидуального практикума. Задания должны быть разнотипными по сложности. Индивидуальный практикум предполагает большую, по сравнению с лабораторными занятиями, самостоятельность. Связано это, прежде всего, с разнотипностью заданий по содержанию внутри одной темы. Трудно или даже иногда невозможно определить, какую задачу решать в качестве образца. Особенно это имеет место при изучении основ алгоритмизации и программирования. Например, в теме «Циклические алгоритмы» это может быть и вычисление конечных и бесконечных сумм, произведений и факториалов, оригинальный вывод, задачи целочисленной арифметики и другие алгоритмы. Ещё более разнообразны задачи, например, по теме «Одномерные массивы».

Из предыдущей особенности следует необходимость обращаться к учебникам, к справочному материалу при выполнении индивидуальных заданий. Поэтому такая форма занятий, как ни одна другая, учит самостоятельности, работе с литературой и другими источниками информации, включая и компьютерные. Учителю необходимо иметь в виду и быть готовым к тому, что при выполнении индивидуальных заданий со стороны учеников, особенно сильных, могут поступать разнообразные, самые неожиданные, нестандартные вопросы.

Иногда полезно выполнять индивидуальные задания вдвоём. Но в этом вопросе не должно быть двух крайностей: поощрение коллективной парной работы и, наоборот, полный запрет совместной работы. В связи с этим возникает ряд вопросов. Например, предлагать одно задание на двоих или каждому свое? Если давать два задания, то они должны быть одного типа, одного уровня сложности или разные? Оба школьника должны быть одинаковы по способностям или разные? Как вы отнесетесь к тому, если один из двух учеников будет играть, возможно, пассивную роль? Предлагается также подумать над ответами на эти и другие подобного рода вопросы.

1.6.4. Вспомогательные организационные формы

Иногда бывает полезным и не только проводить занятия в кабинете. Рассмотрим основные формы, которые помогут разнообразить уроки информатики. Основные цели **экскурсии** следующие:

показать «живую» информатику, а точнее её какой-нибудь раздел в конкретной сфере;

скорректировать у учащихся «книжные» теоретические представления о настоящей информатике;

сформировать интерес к предмету (разделу), если экскурсия проводится до его изучения;

обобщение знаний, их систематизация, показ связи курса с реальностью, если экскурсия проводится на завершающем этапе;

профориентация на профессии, связанные с использованием ЭВМ.

Экскурсия не обязательно должна быть на крупное предприятие, в большую организацию. Может быть вполне достаточно, чтобы бухгалтер или секретарь, работник отдела кадров, или библиотекарь показали и прокомментировали конкретную систему в действии, на реальных данных.

Экскурсия должна быть тщательно подготовлена. Учитель должен договориться о том, кто, когда и в каком объёме, в каком режиме будет показывать её. Предварительно учителю нужно самому ознакомиться с тем, что будет показано на экскурсии. Нужно заготовить перечень вопросов, на которые учащиеся должны ответить во время экскурсии, по её окончании, или во время уроков. Необходимо также подготовить разъяснения по другим вопросам, на которые не обязательно должны отвечать школьники.

Итог экскурсии есть её коллективное обсуждение и анализ увиденного. Школьники должны выразить своё отношение к проведенному мероприятию, а также возможные перспективы дальнейшего развития по данному вопросу, изучаемому в рамках экскурсии.

Также распространены факультативные занятия и кружки по информатике.

Цели факультативных занятий следующие:

углубление знаний в конкретном разделе информатики;

показать связи информатики с другими дисциплинами;

профессиональная ориентация, в том числе и на профессию педагога.

Особое внимание надо обратить на межпредметный факультатив: информатика и математика, информатика и физика, даже информатика и литература или история и др. Такие курсы полезны по следующей причине. Образование разделено на непересекающиеся дисциплины и темы. Поэтому учащемуся трудно привести знания, полученные при изучении разных предметов, в какую-то единую систему. Такие факультативные занятия помогают увидеть связи между предметами.

Полезен факультативный курс, связанный с управлением школой. С администрацией можно согласовать и разработать подсистемы следующего характера: «Оборудование», «Библиотека», «Документация», «Расписание» и т. п. Такая работа должна заинтересовать учащихся. Кроме этого, она будет носить определённый воспитательный эффект.

Не обязательно ставить целью на факультативных курсах изучить углубленно конкретный язык программирования, что обычно практикуется. Отталкиваться надо от конкретных задач. Работая с библиотекарем, зная её задачи, можно лучше понять, например, необходимость внешней памяти, то есть файлов и средств работы с ними.

Характерные особенности факультативных занятий:
сравнительно большая самостоятельность;
самоуправление познавательной деятельностью;
количество числа учащихся меньше.

Кружок есть наиболее гибкая, глубоко индивидуальная форма работы с разнообразным содержанием. Могут участвовать ученики разных возрастов. Обычно в кружке занимаются те, кто проявил повышенный интерес к предмету. Его могут посещать и не очень способные ученики. Главное, чтобы кружок школьник посещал по собственному желанию и его участие не влияло на оценку по предмету.

Для старшего возраста на кружке можно решать объёмные задачи, связанные целостным содержанием: разработка определенных баз данных для школы, сервисных программ для учителя информатики или обучающих программ по другим предметам.

При работе с младшими школьниками следует избегать уклона в стандартный курс информатики, который они ещё не изучали. Потом это изучать будет не интересно. Возможное содержание кружка: освоение готовых программных средств, способствующих закреплению знаний из других дисциплин, изучаемых параллельно, т. е. обучающих программ. Для поддержания интереса можно рекомендовать простейшие графические пакеты.

Что же касается **олимпиад**, то в нашей стране она проводится в четыре этапа: школьный, городской, областной и республиканский. К участию в первом туре приглашаются все желающие участвовать в олимпиаде. С каждым туром нестандартные задачи становятся сложнее и к самому последнему этапу допускаются самые сильные участники из всей страны. Также в рамках школ имеет место проведение различных конкурсов. Например, конкурс *«Инфомышка»* проводится с целью развития и поддержки интереса школьников к изучению информатики. Он проводится по правилам популярного у школьников международного математического конкурса *«Кенгуру»*.

1.7. Урок информатики

1.7.1. Урок и типы уроков

Основной формой организации учебно-воспитательной работы с учащимися по всем предметам в учреждениях среднего образования является урок. Школьный урок образует основу классно-урочной системы обучения, характерными признаками которой являются:

- постоянный состав учебных групп учащихся;
- строгое определение содержания обучения в каждом классе на каждом уроке;
- определённое расписание учебных занятий;
- сочетание индивидуальной и коллективной форм работ учащихся;
- ведущая роль учителя;

систематическая проверка и оценка знаний учащихся.

Как показывает опыт, который накопила наша школа после введения курса информатики, преподавание основ этого предмета наследует всё дидактическое богатство отечественной школы. С другой стороны, в условиях внедрения в учебный процесс кабинета вычислительной техники и новых информационных и коммуникационных технологий, весь известный опыт должен быть подвергнут критическому анализу, чтобы всё прогрессивное стало достоянием практики в преподавании этого молодого предмета. Применение кабинета вычислительной техники и информационных технологий может существенно изменить характер школьного урока, что делает ещё более актуальным поиск новых и совершенствование старых форм и методов обучения.

Классификацию типов уроков или их фрагментов можно проводить, используя различные критерии. Главный признак урока есть его дидактическая цель, показывающая, к чему должен стремиться учитель. Исходя из этого признака, в дидактике выделяются следующие типы уроков:

урок сообщения новой информации (урок-объяснение);

урок развития и закрепления умений и навыков (тренировочный урок);

урок проверки знаний, умений и навыков.

В большинстве случаев учитель имеет дело не с одной целью, а с несколькими или даже со всеми сразу. Поэтому на практике широко используются *комбинированные уроки*. Они могут иметь разнообразную структуру и обладают рядом достоинств: обеспечивается смена видов деятельности, создаются условия для быстрого применения только что полученных новых знаний, имеет место обратная связь, накопление оценок, индивидуальный подход в обучении и др.

Важнейшая особенность постановки курса информатики на базе кабинета вычислительной техники есть регулярная работа школьников на электронно-вычислительных машинах. Поэтому учебные фрагменты на уроках информатики можно классифицировать по объёму и характеру использования вычислительной техники.

В своё время классическая педагогика осознала недостаточность словесного образования, указав на необходимость приобретения информации не из книг и лекций, а через наблюдение, познание самих предметов и явлений. Поэтому полезным методом преподавания применительно к информатике является наблюдение учащегося за работой товарища или учителя, а также наглядные методы: иллюстрация и демонстрация.

Основная дидактическая цель этих методов — наглядно сообщить новую информацию. Используя демонстрационный экран, учитель показывает различные учебные элементы содержания темы: новые объекты языка или изучаемой системы, фрагменты программы, рисунки, схемы, тексты и т. п. При этом учитель сам работает за компьютером, а учащиеся наблюдают за его действиями, чтобы затем повторить его же самостоятельно на своих компьютерах. Второй синхронный вариант предполагает, что школьники на экранах своих компьютеров повторяют те же действия одновременно,

параллельно с учителем. Учитель может пересылать специальные демонстрационные материалы на ученические компьютеры, а учащиеся работают с ними самостоятельно. Не только учитель, а и учащиеся могут демонстрировать свои результаты. Полезно иллюстрировать, прежде всего, то, что «не видно»: например, модель всей памяти электронно-вычислительной машины или одной ячейки с записью и перезаписью в неё значений. Учитель может демонстрировать работу за компьютером по-разному. Любой современный компилятор демонстрирует место, и даже смысл ошибки. Особенность компьютерной демонстрации выражена в динамичности и управляемости наглядными образами. Есть возможность вмешаться в процесс демонстрации для индивидуализации темпа или повторений. Возрастание роли и дидактических возможностей компьютерных демонстраций объясняется развитием и совершенствованием общих графических возможностей современных компьютеров и специальных технологий, предназначенных для этих целей. Например, к демонстрации можно также отнести современные средства презентации. Динамичность развития компьютерных технологий может существенно изменить сам урок.

1.7.2. Роль учителя и особенности урока по информатике

В связи с распространением технологий компьютерного обучения, которые берут на себя всё больше и больше педагогических функций, становится актуальным вопрос о возможных изменениях роли и обязанностей учителя. Большинство участников всевозможных дискуссий приходят по этому вопросу к следующему мнению: компьютер, вооружённый хорошими педагогическими программами, должен быть надёжным и дружественным помощником учителя и ученика. Он берёт на себя наиболее рутинные функции. Учителю остаются наиболее творческие задачи обучения, воспитания и развития. Ведение дискуссий, поощрение тех или других действий ученика, поддержание дисциплины, личностное общение с учеником и его родителями и другие подобного рода функции ни одна самая лучшая программа, ни один самый мощный компьютер лучше учителя ещё долго, а, может, и всегда, не сделает.

Остановимся на некоторых дидактических особенностях уроков по информатике, связанных со специфическим характером учебного материала и использованием компьютера. Некоторые из них были подмечены ещё до введения предмета в школах.

Обучение школьников в условиях постоянного доступа к вычислительной технике обычно проходит при повышенном эмоциональном состоянии учащихся. Объясняется это частично тем, что школьник очень скоро обнаруживает состояние «власти над умной машиной». Это придаёт ему уверенности, у школьника возникает естественное стремление поделиться своими знаниями с теми, кто ими не обладает. Поэтому наиболее успешно занимающиеся учащиеся могут помогать учителю.

При обучении школьников компьютерным технологиям полезным является деятельностный подход к обучению, и, в частности, так называемый метод проектов. Под разработкой учебного проекта понимается определённым образом организованная целенаправленная деятельность. Таким проектом может быть, например, компьютерный курс изучения определённой темы, логическая игра, смоделированный на компьютере макет лабораторного оборудования, разработка программы на языке программирования и многое другое. В простейшем случае, на начальном этапе изучения информатики в качестве сюжетов для изучения компьютерной графики можно предложить задачи проектирования рисунков животных, строений, симметричных узоров и т.п. При использовании метода проектов необходимо учитывать ряд условий.

Учащимся следует предоставить широкий набор проектов для реального выбора, которые могут быть как индивидуальными, так и коллективными. Школьника необходимо ознакомить с инструкцией по работе над проектом. При этом можно учитывать, что одни лучше усваивают материал, читая текст, другие — слушая объяснение, третьи — непосредственно пробуя, ошибаясь и находя решения в процессе практической работы.

Для ребенка важна практическая значимость полученного результата и оценка со стороны учителя, сверстников, родителей. Поэтому проект должен предполагать для исполнителя законченность и целостность проделанной им работы, желательно по возможности в игровой или имитационной форме. Важно также, чтобы завершённый проект был презентован. Необходимо создать условия, при которых школьники имеют возможность обсуждать друг с другом свои успехи и неудачи.

1.7.3. Технические и программные средства обучения информатике

Введение в учебный план средней школы нового предмета потребовало разрешения проблемы взаимодействия учащихся с электронно-вычислительной техникой. Это затрагивало в конечном итоге и интересы преподавания других школьных дисциплин. Рассматривались различные пути решения этой задачи, например, создание и разработка вычислительных центров коллективного пользования. Основным из них стал следующий: оборудование в школах кабинетов, оснащённых комплексами учебной вычислительной техники на базе программного и электронного оборудования, включённых в локальные и глобальные сети.

Кроме компьютеров, к техническим средствам обучения информатике можно отнести также:

Учебное, демонстрационное оборудование, сопрягаемое с вычислительной техникой (например, учебные роботы, управляемые компьютером; модели для демонстрации принципов работы вычислительной техники, её частей, устройств и др.);

Средства телекоммуникаций, обеспечивающие доступность информации для обучаемых, вовлечённость их в учебное взаимодействие, богатое

возможностями и разнообразием видов использования ресурсов всемирной информационной сети;

Используемое в кабинете информатики программное обеспечение должно включать:

Системное программное обеспечение (операционная система, операционные оболочки, антивирусные средства и т. п.);

Программные системы базовых информационных технологий (текстовый и графический редакторы, системы управления базами данных, электронные таблицы, графические системы, системы подготовки компьютерных презентаций и др.);

Инструментальное программное обеспечение — средства разработки программ;

Программное обеспечение учебного назначения;

Программное обеспечение издательской деятельности.

Так как программные средства быстро сменяются и устаревают, то их полный обзор сделать практически невозможно.

Помимо компьютерного оборудования, кабинет информатики рекомендуется оснащать:

Набором учебных программ для изучения информатики и отдельных разделов некоторых других предметов;

Комплектом учебно-методической, научно-популярной и справочной литературы;

Журналом входного и периодического инструктажей учащихся по технике безопасности;

Журналом использования каждого устройства на каждом рабочем месте;

Журналом сведений об отказах вычислительной техники и их ремонте;

Аптечкой первой помощи и средствами пожаротушения;

Инвентарной книгой учёта имеющегося в кабинете учебного оборудования, планами его дооборудования.

При оборудовании и использовании компьютерного класса большое значение имеет строгое соблюдение санитарных норм и правил, предназначенных для предотвращения неблагоприятного воздействия на человека вредных факторов, связанных с использованием техники. Этот вопрос ещё более актуален, так как речь идет о здоровье детей.

Согласно гигиеническим требованиям, для учителей длительность работы в дисплейных классах и кабинетах информатики устанавливается не более 4 часов, а для инженеров — 6 часов в день. Для снижения нагрузки в течение рабочего дня должны устраиваться регламентированные перерывы в работе. Разрешаемое время непрерывной работы учащихся за дисплеями зависит от их возраста и не должно превышать 20 мин. для учащихся 6 – 7 классов, 25 мин. — для 8 – 9 классов и 30 мин. — для 10 – 11 классов. После установленной выше длительности непрерывной работы должен проводиться комплекс специальных упражнений для глаз, а после каждого урока — физические упражнения для профилактики общего утомления. Число уроков для учащихся 10 – 11 классов с

использованием компьютеров должно быть не более двух в неделю, а для остальных классов — не более одного урока. Внеклассная работа с использованием электронно-вычислительных машин должна проводиться не чаще двух раз в неделю общей продолжительностью не более 60 минут для учащихся 2 – 5 классов и 90 мин. — для 6 класса и старше.

Фактор санитарно-гигиенических требований к организации учебного процесса в кабинете вычислительной техники накладывает жёсткие ограничения на структуру каждого урока по информатике, что должно учитываться при их планировании. В частности, это касается, прежде всего, учёта продолжительности любой работы за включёнными компьютерами.

Для обеспечения организации работы кабинета информатики приказом директора школы назначается заведующий кабинетом вычислительной техники из числа учителей информатики. Он является организатором работы учителей и учащихся по применению средств вычислительной техники, информационных технологий в преподавании информатики и других учебных предметов. Кроме этого, заведующий принимает меры по дооборудованию и пополнению кабинета учебно-наглядными пособиями, вычислительной техникой и другими техническими средствами, несёт ответственность за работоспособность и сохранность всего имеющегося в кабинете оборудования, своевременность и тщательность его профилактического технического обслуживания, за поддержание в кабинете санитарно-гигиенических требований и требований техники безопасности.

Вводный и периодический инструктаж по ней должен проводить, как правило, учитель информатики, который проводит занятия в соответствующем классе. На вводном инструктаже, который проводится в виде небольшой лекции, а лучше беседы, учитель знакомит учеников с распорядком в кабинете, правилами техники безопасности и гигиены труда, с потенциально опасными моментами, которые могут возникнуть в процессе работы, и с соответствующими мерами предосторожности. Краткий периодический инструктаж непосредственно перед работой за компьютером дополняет вводный и знакомит с правилами выполнения конкретной работы, с возможными опасными ситуациями и правилами поведения при их возникновении. Все сведения по проведению инструктажа заносятся в специальный журнал, в котором должны быть записаны фамилии, кто и кого инструктировал и их подписи, а также краткое содержание инструктажа.

Помощь в работе заведующему кабинетом оказывает лаборант, который находится в непосредственном подчинении заведующего кабинетом. Он отвечает за правильное хранение и использование учебного оборудования, участвует в его приобретении и ведёт инвентаризацию. По плану преподавателя и под его руководством лаборант готовит оборудование к работе, обеспечивает постоянную готовность противопожарных средств и средств первой помощи. Он должен регистрировать отказы техники во время занятий и проводить по возможности мелкий ремонт вышедшего из строя оборудования. Согласно

санитарным требованиям при кабинете информатики должна быть лаборантская комната определённой площади, соединённая с учебным помещением.

1.8. Методика преподавания основ алгоритмизации и программирования

1.8.1. Методические особенности раздела

Раздел «Основы алгоритмизации и программирования» имеет явно выраженную практическую направленность, что способствует усилению связи обучения с жизнью. В этом разделе, как ни в одном другом, большое внимание должно уделяться решению задач и выполнению упражнений. При решении некоторых задач строится математическая модель и вычислительный алгоритм, требующие обоснований, выходящих за рамки школьной программы. Происходит расширение традиционного, известного из других школьных предметов понятия величина. Вводятся и используются не только числовые, но и литерные величины. Кроме этого, ученики знакомятся с новыми способами организации данных в форме одномерных и двумерных массивов.

При изучении программирования школьника надо приучить абсолютно точно и по определённым фиксированным правилам записывать алгоритм. Поэтому с самого начала необходимо привить школьникам такие качества, как чёткость, ясность, аккуратность в записях, предельную внимательность и сосредоточенность, особенно при работе на компьютере.

Необходимо иметь в виду следующую методическую сложность. Алгоритмический стиль мышления, который мы должны сформировать, отличается от математического, хотя и основывается на нём. Решить задачу и составить алгоритм, с помощью которой вычислительная техника должна решить ту же задачу, это не всегда одно и то же, то есть требует различных способностей и сторон мышления. Например, даже ученики с хорошими математическими способностями не сразу понимают динамический смысл записи алгоритма. Если любое записанное действие при решении математической задачи выполняется всегда, если оно записано, то в программе, например, может выполняться только одна из двух её ветвей. Некоторая последовательность действий может повторяться, динамически может меняться значение некоторой величины и т. п. Эти трудности возрастают в условиях изучения алгоритмов без использования техники. Как преодолеть эту сложность? Для лучшего понимания учениками работы алгоритма можно рекомендовать следующие приемы:

исполнение алгоритма вручную;

использование блок-схем;

выполнение программы по шагам, т. е. трассировка программы, с помощью современных средств программирования.

1.8.2. Общие методические принципы обучения

Принцип многоуровневости связан с тем, что в алгоритмизации и программировании, как ни в одной другой области деятельности, все темы, элементы языка тесно переплетены между собой. Поэтому некоторую трудность представляет определение порядка изучения понятий и конструкций языка, приёмов и методов программирования. Ряд учебников построен таким образом, что сначала рассматриваются элементы выбранного языка программирования (постоянные величины и переменные, все их типы, все операции над ними, правила построения выражений и т. д.), а потом последовательно изучаются операторы, методы и технологии программирования. При этом, как правило, сразу в одном пункте приводятся все возможности изучаемого элемента языка, например оператора, хотя некоторые из них редко используются или достаточно сложные. Аналогично при изучении какой-нибудь другой системы (не обязательно системы программирования) приводится полное описание некоторого элемента, хотя на начальном этапе достаточно знать её простейшую форму, лишь некоторые ее возможности, которых хватает для решения достаточно обширного класса задач.

Такую общепринятую методику можно усовершенствовать следующим образом. Параллельное изучение языка программирования и средств разработки программы делим на этапы (уровни). Первый из них (основы, начало, быстрое введение и т. п.) включает лишь некоторые наиболее простые и часто используемые в дальнейшем элементы (типы данных (вещественные и целочисленные), операции над ними (арифметические), операторы, стандартные функции и процедуры, и т.п.), обеспечивающие возможность быстрой разработки простейших программ. После приобретения первоначальных навыков программирования школьник осваивает второй уровень (основной). На этом этапе изучаются новые элементы языка и операционной системы, а также рассматриваются новые возможности тех же конструкций языка и операционной системы, основы которых были изучены на первом этапе. Первые два уровня должны быть усвоены всеми учениками класса.

Третий, заключительный уровень состоит в изучении дополнительных возможностей системы и языка, которые могут встретиться реже, при разработке более сложных программ. Изучаемые на этом этапе возможности операционной системы и языка программирования позволяют повысить эффективность использования вычислительной техники и оптимизировать программы. Такие дополнительные возможности есть смысл изучать со всеми учениками только в специализированных классах, а в обычной школе лишь на дополнительных занятиях с учениками, проявляющими повышенный интерес и способности к программированию.

Рекомендуется в качестве основного принципа изучения алгоритмизации и программирования использовать **принцип предварительной мотивации** элементов выбранного языка программирования. Это означает, что любую тему лучше изучать «от частного к общему», а не наоборот. Элементы языка рекомендуется вводить по следующей схеме:

1. Наглядная простая задача, которая требует для своего решения введения нового элемента, т. е. показывается необходимость изучаемого элемента.
2. Пример использования нового элемента при решении этой задачи.
3. Общий вид конструкции, её формальное описание.
4. Примеры на закрепление.
5. Выполнение индивидуальных заданий на электронно-вычислительной машине и упражнений без использования компьютеров.

При этом пункты с первого по третий лучше рассмотреть в форме лекции, а пункты четыре и пять на практических занятиях. При этом теоретические сведения желательно подавать в сокращённом виде и основные методы, приёмы программирования должны изучаться на простых наглядных примерах.

Следуя этому принципу, не обязательно всегда соблюдать одинаковый порядок форм занятий: лекция – практические занятия – лабораторные работы. При изучении некоторых тем, например, ввод-вывод данных, эффективнее сначала решить задачу с использованием компьютера или выполнить упражнения без ЭВМ, а потом на лекции обобщить, подвести итоги и сделать выводы, объяснить новые дополнительные возможности изучаемого элемента и обратить внимание на наиболее сложные вопросы темы.

При изучении многих тем следует руководствоваться **принципом сравнения**, который предполагает анализ различных алгоритмов или программ решения одной и той же задачи, выбор из них наилучшего решения.

Этот же принцип можно использовать также для того, чтобы показать необходимость той или иной конструкции языка, быстрее и лучше её понять. Для этого можно записать несколько вариантов некоторого фрагмента программы решения одной и той же задачи, используя разные элементы. Можно привести, например, следующие такого рода элементы:

представление положительных и отрицательных целых чисел в памяти вычислительной техники;

логические операции «и» и «или» (не важно, в каком языке);

операторы цикла «*while*» и «*for*» в языке «*Pascal*»;

вспомогательные алгоритмы и вспомогательные алгоритмы для вычисления функций в «школьном» алгоритмическом языке.

Один из вариантов, как правило, эффективнее и «красивее», хотя он может быть более трудным для изучения.

Благодаря такому методическому приёму реализуется на практике **принцип повторения**. Один из сравниваемых элементов изучается обычно раньше (например, оператор «*if*»). Повторение при изучении основ алгоритмизации и программирования в большей степени, чем в математике, играет важную роль. Это связано с тем, что при изучении нового элемента языка, при написании программ по новой теме надо знать практически почти все ранее изученные темы. Например, выражения, операторы «*if*», ввода-вывода будут использоваться при написании программ по любой новой теме.

Рассмотрим теперь **принцип индивидуальных заданий**. При обучении программированию следует исходить из того, что развитие алгоритмическое

мышление школьника, научить его программировать и решать задачи с применением вычислительной техники можно только при условии, если каждый из них будет регулярно составлять программы, выполнит самостоятельно несколько индивидуальных заданий. Каждое такое задание закрепляет и проверяет знания по одной или нескольким наиболее важным темам и предполагает выполнение следующих этапов:

- изучение постановки задачи;
- разработку алгоритма решения;
- запись алгоритма в виде программы;
- подготовку программы на компьютере в соответствующей системе программирования;
- отладку и тестирование программы;
- анализ результатов и подготовку отчета.

В отличие от традиционной методики, когда всему классу предлагается общая задача, при такой организации работы школьника исключается формальное переписывание друг у друга. При этом можно учесть способности школьников, используя задачи с различным уровнем сложности. Важно также, что при выполнении индивидуальных заданий повышается активность на занятиях и ответственность за его своевременное выполнение. При этом эффективность обучения возрастает, если использовать следующие методические приёмы:

- выполнение одного задания вдвоём;
- доклад наиболее интересных, оригинально и нестандартно выполненных заданий на практическом занятии и его обсуждение;
- обмен готовыми программами друг с другом с объяснением;
- конкурс на лучшую программу решения одной и той же задачи;
- бригадный метод работы (одно задание даётся группе людей от двух до пяти человек). Это позволяет ученикам обмениваться идеями друг с другом, искать вместе ошибки, учить работать в коллективе.

Вопрос остаётся открытым для читателя о возможности и трудностях выбора уровня заданий при выборе варианта индивидуальных заданий. Неправильно поступают те учителя, которые всему классу в качестве домашнего задания предлагают одни и те же или одинаковые по уровню сложности задачи. Более способным ученикам мало пользы от простых заданий, а слабые ученики вынуждены обращаться за помощью при выполнении трудных, заданий.

Важным принципом изучения основ алгоритмизации и программирования в машинном варианте, кроме рассмотренных, должен быть **принцип параллельности** изучения выбранного языка программирования и интегрированной среды разработки программ. Параллельно с этим можно изучать и операционную систему. Это позволит уже с первых уроков часть занятий проводить в дисплейных классах. Необходимо обратить внимание на то, что не должно быть двух крайностей в изучении начал программирования при наличии в школе компьютеров. Первая из них заключается в том, что сначала в течение некоторого времени изучаются основы языка программирования в

безмашинном варианте традиционными методами с мелом у доски и только потом осваиваются методы разработки программ и школьников допускают к компьютерам. При таком подходе теряется интерес к программированию и многие его элементы, динамический смысл выполнения операторов без работы с компьютером остаются непонятными длительное время. Нельзя согласиться и с теми учителями информатики, которые все уроки проводят в дисплейных классах и (или) на несколько минут отключают компьютеры для объяснения нового материала. При такой методике меньше внимания уделяется развитию алгоритмического мышления и наиболее принципиальным общим вопросам и методам программирования, что является самым важным при изучении любого языка программирования.

1.8.3. Формы занятий

С учетом отмеченных принципов рекомендуется организовать изучение основ алгоритмизации и программирования в машинном варианте с использованием следующих форм:

лекция без электронной техники;

практическое занятие без вычислительной техники;

лабораторные занятия в дисплейных классах с использованием электронно-вычислительных машин;

индивидуальные занятия и консультации.

Основным принципом организации занятий в школе по основам алгоритмизации и программирования в условиях систематического использования вычислительных машин должно являться взаимосогласованное изложение теории и практики с опережающим изложением практических вопросов во время работы на машине. При этом приоритет должен быть отдан решению задач, поэтому основными формами занятий должны быть семинарские и лабораторные, во время которых решаются задачи и выполняются упражнения. Такие занятия рекомендуется проводить с половиной класса (до 15 человек), чтобы каждый ученик имел возможность готовить и отлаживать на электронной машине программы.

Лекцию можно проводить со всем классом или с двумя параллельными классами (до 40 человек) в течение не более 45 минут. Двухчасовая лекция в школе малоэффективна, несмотря на то, что такой формой занятий иногда руководители учреждений образования восхищаются. Опыт показывает, что в этом случае после первого часа ученики с трудом воспринимают материал. Сэкономленное за счет объединения двух классов время можно использовать для индивидуальной работы. Школьникам, проявляющим повышенный интерес к программированию, можно рекомендовать предварительно изучить материал лекции по литературе. Тогда на лекции, практическом занятии он может получить ответ на возникшие во время самостоятельного изучения вопросы.

Распределение часов между разными формами занятий зависит от изучаемой темы, количества учеников на лекции и практических занятиях, уровня их подготовленности. Например, если тема имеет теоретическую направленность, то соотношение может быть таким: одна лекция, половина практического содержания и половина индивидуальной работы с электронно-вычислительной машиной. При таком варианте практическое и лабораторное занятия проводятся за 1 урок в дисплейном классе. Если тема имеет практический характер, то можно рекомендовать следующее соотношение: половина лекции, половина практического занятия, 1 занятие индивидуальной работы с вычислительной техникой.

Эффективными являются также «сдвоенные» уроки информатики. Тогда за 2 часа в дисплейном классе можно провести все формы занятий. Два часа подряд без компьютера можно с пользой работать, мастерски используя различные методические приёмы и умело варьируя формами занятий (повторение в форме беседы предыдущей темы, лекция по теоретической части нового материала, выполнение небольших упражнений учителем или учениками, беседа по закреплению изученного материала). Но при любой организации занятий из медицинских соображений школьник не должен работать за дисплеем более 45 минут подряд в течение дня.

1.8.4. Изучение основ алгоритмизации и программирования с использованием компьютеров

Можно предложить несколько разнообразных форм проведения лабораторных занятий с использованием компьютеров. Они выбираются в зависимости от изучаемой темы и поставленной цели занятия, уровня подготовленности учащихся, наличия компьютеров, количества учеников и других факторов. Например, рассмотрим следующие методические приёмы.

Сначала, не включая компьютеры, учитель даёт возможность записать основные команды, порядок работы, а затем на этом же или на следующем занятии каждый ученик, используя конспект, выполняет на компьютере то, о чём было рассказано.

Если учеников не очень много (не более 10 человек), объясняемый материал можно сразу показывать на одном (или двух) дисплеях, а затем каждый ученик закрепляет показанные элементы на своём компьютере.

Этот вариант обучения предполагает синхронное выполнение учениками действий на компьютере по командам, которые учитель записывает на доске. Недостатком этого способа является то, что выполнять команды ученики будут неравномерно, и некоторые из них должны будут ждать следующего указания учителя. Следующую команду учитель может подавать лишь после того, когда предыдущий пункт выполнен всеми учениками.

Эти три методических приёма целесообразно использовать на начальном этапе обучения работе на компьютерах или при демонстрации новых

возможностей вычислительной техники. При этом не следует включать много компьютеров. Чем больше включено компьютеров при объяснении нового материала, тем труднее объяснять его. На первых занятиях, а также когда осваивается новая тема, за одним компьютером можно работать нескольким школьникам.

При закреплении показанных навыков, при выполнении индивидуальных заданий основной является индивидуальная форма работы учителя с учениками, при которой по мере возникновения у них вопросов учитель оказывает необходимую помощь.

1.9. Решение задач по основам алгоритмизации и программирования

1.9.1. Выбор задач

При выборе задач по основам алгоритмизации и программирования необходимо руководствоваться следующими принципами. Важно не только и не столько изучить правила конкретного языка программирования и обучить школьника записывать с его помощью алгоритм. Прежде всего, и это самое главное, задачи должны способствовать развитию алгоритмического стиля мышления, помочь овладеть общими методами и приемами программирования. С помощью задач человек должен освоить основные типы алгоритмов (ветвления, циклы с известным и неизвестным количеством повторений), научиться использовать и обрабатывать данные различных типов (целые, вещественные, логические, символьные, строки, массивы, структуры).

Не существенно, из какой области знаний взяты задачи. Гораздо важнее, можно ли с их помощью научиться разработке качественных программ и освоить фундаментальные понятия информатики и, в частности, алгоритмизации. Но с другой стороны, с целью поддержания межпредметных связей, демонстрации возможностей применения вычислительной техники, для разнообразия примеров желательно решать с помощью электронно-вычислительных машин не только математические задачи. В противном случае школьник, проявляющий способности к программированию, но у которого определены проблемы с математикой, может потерять интерес к информатике.

Задачи должны быть интересными, наглядными, понятными школьнику, разнообразными не только по тематике, но и по методическим приемам, используемым при их решении. Задачи не должны требовать много времени на объяснение их содержательной постановки. Перед выбором каждой задачи должны быть поставлены определенные цели, и её решение должно способствовать их достижению.

Все задачи по программированию можно разделить на два больших класса: «задачи-программы» и упражнения.

Первый из них включает задачи на составление алгоритма и полной программы на выбранном языке и их анализ, а в машинном варианте, кроме

этого, и её отладку. В упражнениях требуется записать один или несколько вариантов элемента языка (записать выражение, оператор) или части программы или проанализировать их. При анализе программы (или её части) надо ответить на ряд вопросов, например, есть ли ошибки, всегда ли ошибка будет проявляться, как повлияет на результат какое-нибудь изменение в программе и т.п. Первый класс задач является более важным, и его необходимо чаще практиковать, особенно для закрепления одной или нескольких тем и при выполнении индивидуальных заданий. Но на начальном этапе изучения темы и при объяснении наиболее сложных принципиальных вопросов нельзя игнорировать упражнения, которые играют подготовительную, вспомогательную роль. Упражнения особенно полезны при работе с отстающими школьниками. Частным случаем упражнений являются тесты.

Одним из этапов обучения данного раздела и важным условием умения составлять качественные программы является понимание готовых программ, умение их «читать». Поэтому полезно выполнять упражнения на восстановление постановки задачи, для которой записана программа или её часть. В случае затруднения можно рекомендовать исполнить её для одного или нескольких тестов, или составить блок-схему.

Необходимо обратить внимание на такие упражнения, в которых надо сравнить несколько вариантов программы или алгоритма. При этом необходимо выбрать правильный вариант среди предложенных вариантов. Полезны упражнения, в которых требуется исследовать на эффективность программу или её часть, выбрать наилучший из правильных вариантов или записать, если можно, более эффективный вариант.

1.9.2. Упражнения на составление и анализ блок-схем задач

При изучении основных типов алгоритмов, наиболее важных и сложных операторов и приёмов программирования полезно использовать задачи и упражнения на составление и анализ блок-схем. Кроме простоты и наглядности, важной отличительной их особенностью и достоинством является то, что они почти не зависят от языка программирования. Научившись записывать алгоритмы в виде блок-схем, достаточно освоить синтаксические правила конкретного языка, чтобы можно было разрабатывать эффективные программы. Блок-схемы способствуют быстрому развитию алгоритмического мышления. Их можно использовать как в машинном варианте, так и в безмашинном варианте изучения основ алгоритмизации и программирования. К сожалению, в последнее время в учебниках по программированию этому уделяется недостаточно внимания для полного понимания сущности программ.

Блок-схемы можно изучать как параллельно с языком программирования, так и последовательно. В первом случае сначала в течение первых двух уроков учащимся дают простейшие элементы блок-схем, правила их разработки и требования к ним. При изучении методов программирования и наиболее сложных

операторов языка составляется при необходимости блок-схема алгоритма, а потом с её помощью разрабатывается программа. При таком способе, хотя и требуется больше времени, школьники лучше понимают динамический смысл выполнения программ.

Последовательное изучение блок-схем и языка программирования заключается в том, что на начальном этапе изучения алгоритмов несколько уроков учащиеся разрабатывают и анализируют блок-схемы различных типов алгоритмов (линейных, разветвляющихся, циклических с известным и неизвестным количеством повторений, итерационных, и др.). Потом эти же типы алгоритмов изучаются на уровне программ. При этом не обязательно решать только математические задачи. Эффективным является составление и анализ блок-схем «бытовых» алгоритмов. Вот примеры таких алгоритмов, возникающих в повседневной жизни:

алгоритм перехода улицы при различных условиях;

алгоритм прохода в метро пассажира, пользующегося (или не пользующегося) льготами;

алгоритм приготовления завтрака;

различные игровые алгоритмы.

На уровне блок-схем полезно использовать также задачи по управлению движением робота на клетчатом поле.

Когда ученик научится разрабатывать блок-схемы основных типов алгоритмов, можно изучать основные конструкции языка программирования, основываясь на полученных навыках и умениях. Это выполняется качественнее и быстрее, чем без предварительного изучения блок-схем. При таком подходе удачно реализуется принцип повторения. Кроме этого, экономится машинное время, так как разработка блок-схем не требует использования компьютеров. На уровне блок-схем легко показать основные принципы и понятия структурного программирования: пошаговая детализация алгоритма и базовые алгоритмические (управляющие) структуры с одним входом и одним выходом.

Кроме обычных блок-схем в виде фигур, соединённых линиями, при разработке алгоритмов, представляющих вложенные или последовательные циклы, полезно составить схему циклов или «набросок программы». Отличным примером при составлении блок-схем являются задачи с параметрами. На примере таких задач хорошо проявляются основные достоинства блок-схем. В качестве примера рассмотрим уравнение вида $a \cdot x = 0$, где переменная a является параметром, а x – неизвестной переменной. Для неё можно составить следующую схему (см. рис. 1).

В наброске программы не обязательно нужно использовать геометрические фигуры. Некоторые принципиальные операторы (цикла, if), но не обязательно все, записываются по синтаксическим правилам конкретного языка. А вместо группы некоторых операторов или других элементов записываем на русском языке, что они делают.

Некоторые из действий можно детализировать, используя в зависимости от сложности этой части программы обычную блок-схему, схему циклов или операторы языка.

При использовании блок-схем алгоритмов традиционным основным типом задач являются задачи: «составить блок-схему (схему циклов, «набросок программы») для решения конкретной задачи...». Кроме этого, на разных этапах изучения программирования можно рекомендовать следующие типы задач и упражнений:

По готовой блок-схеме составить программу (часть программы, записать оператор и т. п.);

По заданной программе или её фрагменту составить блок-схему (схему циклов и т. п.);

По заданной схеме циклов написать одну или несколько из возможных постановок задач на определённую тему. Такие упражнения можно использовать на заключительном этапе изучения основ алгоритмизации и программирования, когда учащиеся получают определённый опыт и будут иметь некоторый набор задач разного типа и их алгоритмов.

Кроме этого, при изучении и использовании блок-схем можно рекомендовать те же типы упражнений и методические приёмы, которые перечислены в других параграфах этой главы.

1.9.3. Комплекс взаимосвязанных задач

Особое внимание следует обратить на решение комплекса взаимосвязанных задач. При этом задачи могут быть как однотипными в алгоритмическом отношении и одинаковыми по трудности, так и с последовательным возрастанием их сложности. Например, пусть дана матрица $A[10;15]$ с элементами, которые являются натуральными числами от 1 по 9. Примером комплекса однотипных задач являются следующие задачи:

- для каждой строки матрицы найти количество троек в ней;
- для каждого столбца матрицы найти количество троек в нём;
- во всей матрице найти количество троек.

Алгоритмы данных задач существенным образом не отличаются, поэтому эти задачи имеют однотипный характер.

Примером последовательности задач с усложнением являются следующие задачи:

- найти наибольшее число из двух данных чисел;
- найти наибольшее число из трёх данных чисел;
- найти наибольшее число в одномерном массиве из конечного данного числа элементов;
- найти наибольшее и наименьшее числа в одномерном массиве данного числа элементов и их номера;

найти второе наибольшее число и количество его повторений: за два просмотра или за один просмотр массива. Рассмотреть варианты в зависимости от количества просмотров;

найти наибольший элемент среди элементов матрицы.

В такой последовательности задач решение следующей задачи основывается на решении предыдущих задач.

Комплекс взаимосвязанных задач можно использовать также при сравнении различных типов алгоритмов, а, значит, и различных операторов. Например, для демонстрации отличия циклов с известным и неизвестным количеством повторений можно предложить следующие три задачи. Найти сумму отрицательных чисел массива, если

массив не рассортирован;

массив рассортирован по возрастанию;

массив рассортирован по убыванию.

В первом из предложенных вариантов условия необходимо проанализировать все элементы массива. В остальных задачах количество повторений цикла зависит от элементов массива. Как только найдём неотрицательное число, можно прекращать цикл.

Для того, чтобы лучше понять отличие двух базовых управляющих структур «цикл пока» (оператор «*while*») и «цикл до» (оператор «*repeat*» языка «*Pascal*»), можно решить две следующие задачи. Задан массив и некоторое число. Найти сумму чисел, предшествующих этому числу, не включая это число и включая его. Для каждой из них можно рассмотреть два случая: заданное число есть в массиве или нет в нём.

Это были примеры последовательности задач, которые используются при изучении различных тем (оператор «*if*», одномерные табличные величины и оператор цикла, сложные циклы, двумерные табличные величины и др.). Такой же комплекс задач можно решать и при изучении одной темы. Например, в теме «строки и символы» можно рекомендовать следующую последовательность задач с возрастающей сложностью:

в слове найти количество заданных букв;

в слове найти, какие буквы встречаются чаще других;

в одной строке, в которой слова разделены одним пробелом, найти слова, в которых чаще всего встречается заданная буква;

в одной строке найти слова, в которых чаще всего встречается заданная буква, если между словами один символ-разделитель без пробелов;

в тексте из нескольких строк найти слова, в которых чаще всего встречается заданная буква, если справа и слева от символа-разделителя слов может быть любое количество пробелов;

в тексте из нескольких строк найти слова, в которых чаще всего встречается заданная буква, если слова разделены любым количеством разделителей в любом сочетании. Также рассмотреть отдельно случай, если участвует перенос слов. Разделитель, обозначающий перенос, задан.

Использование комплекса взаимосвязанных задач позволяет учесть уровень подготовленности обучаемых. В одном классе можно остановиться, например, на третьем этапе, а в другом можно «дойти» и до последней задачи. Начинать также можно с любой задачи. При такой методике можно предоставить ученикам возможность самостоятельно выбирать задачи соответствующего уровня. Это эффективнее случайного набора задач, разделённых по сложности на несколько уровней. Если ученик не может справиться с задачей выбранной сложности, он может, не переключаясь на постановку принципиально другой задачи, «опуститься на один уровень ниже». И, наоборот, если задача для ученика окажется простой, он имеет возможность усложнить её. По каждой наиболее важной теме можно подготовить несколько таких последовательностей задач. И тогда этот принцип можно использовать при подготовке индивидуальных и домашних заданий, при проведении проверочных и контрольных работ, при подготовке к конкурсам, олимпиадам и другим соревнованиям. Тогда ученик в состоянии сам оценить уровень своей подготовки. Оценка в таком случае должна зависеть от того, с задачами какого уровня справился ученик.

1.9.4. Методические приёмы, используемые при решении задач

Для более эффективного обучения разделу «Основы алгоритмизации и программирование» необходимо стремиться разнообразить методические приёмы, используемые при решении задач. Рассмотрим некоторые из них.

Для того чтобы показать важность места ключевых слов, можно использовать следующий методический приём. Записываем два варианта алгоритма, отличающихся местоположением ключевых слов, один из которых правильный. Например, «алгоритм приготовления картошки».

Если картошка не очищена;
То очистить картошку;
Сварить картошку на плите;
Всё;
Съесть картошку;
Конец.

Во втором варианте ключевое слово «Всё» записывается строкой выше. При анализе необходимо найти правильный вариант, объяснить, всегда ли будет работать второй вариант. Также важна роль ключевых слов «Начало» и «Конец» в алгоритме, ибо именно от них может зависеть и то, когда закончится некоторое повторяющееся действие.

Исполнение и анализ следующих вариантов фрагмента программы помогает понять назначение ключевых слов «*Begin*», «*End*», «*else*» и их роль при совместном использовании операторов «*For*» и «*If*» языка «*Pascal*».

<p>1 вариант For i:= 1 to 10 Do If A[i] > 0 then Begin k:=k+1; write (' k= ', k); End</p>	<p>2 вариант For i:= 1 to 10 Do If A[i]>0 then k:=k+1; write (' k= ', k);</p>	<p>3 вариант For i:= 1 to 10 Do Begin If A[i]>0 then k:=k+1; write (' k= ', k); End</p>	<p>4 вариант For i:= 1 to 10 Do If A[i]>0 Then k:=k+1; Else write ('k= ', k);</p>
---	--	---	---

Варианты отличаются наличием и расстановкой рассматриваемых ключевых слов.

Для того чтобы объяснить новый элемент языка, один и тот же пример можно решить разными способами. Например, рассмотрим следующее упражнение из темы «Логические выражения»: переменной y присвоить значение $x - 2$, если $3 < x < 5$ и 0 в остальных случаях. Можно проанализировать следующие варианты, правильно решающие данную задачу:

```

if x <= 3 then y := 0
else if x < 5 then y := x - 2
else y := 0;

y := 0;

if x < 5 then
if x > 3 then y := x - 2;
if (x > 3) and (x < 5) then y := x - 2
else y := 0;

if (x <= 3) or (x >= 5) then y := 0
else y := x - 2;

```

При этом некоторые из вариантов можно предложить школьникам записать самостоятельно.

При решении наиболее сложных задач эффективно использование следующего методического приема. Записываем один или два неправильных варианта решения одной и той же задачи, которые часто приводят школьники. Например, преобразовать массив $A[10]$ следующим образом: на место первого элемента поместить 10-й, на место второго – 9-й и т. д., то есть «перевернуть массив».

<p style="text-align: center;"><i>1 вариант (неправильный)</i></p> <pre>for i := l to 10 Do A[i] := A[11 - i]</pre>	<p style="text-align: center;"><i>2 вариант (неправильный)</i></p> <pre>for i := 1 to 10 Do begin d := a[i]; a[i] := a[11 - i]; a[11 - i] := d end;</pre>
---	---

Проанализировав их выполнение на конкретном числовом массиве, ученики должны найти и исправить ошибку в этих вариантах.

Полезно использовать упражнения на выбор наилучшего оператора или фрагмента программы для решения одной и той же задачи. Этот приём эффективно работает в классах с сильными учениками. Например, поиск первого и второго минимального элемента за один просмотр массива эффективнее, чем за два просмотра.

Если с задачей большинство учеников не может справиться, полезным является следующий методический приём. Необходимо дописать пропущенный оператор или его элемент в указанном месте. Это поможет учащимся справиться с трудностями, возникающими при построении решения задачи.

Почти все рассмотренные выше методические приёмы не зависят от изучаемого языка программирования, что расширяет их круг применения. Более того, такого рода приёмы можно использовать не только в средних учебных заведениях, а и в высших как при чтении лекций, так и при проведении различных практических занятий, а также при подготовке тестов и других материалов для контроля и оценки знаний.

1.10. Методика преподавания некоторых основных тем информатики

Одной из наиболее заметных тенденций в развитии школьной информатики является незначительное увеличение места информационных технологий в её содержании. На начальном этапе преподавания этому уделялось меньше внимания, так как на большинстве доступных школам компьютеров отсутствовало соответствующее программное обеспечение. Из разных подходов к изложению этих вопросов в последние годы всё больше применяется подход, при котором изучаются наиболее популярные, широко используемые на практике конкретные программные продукты: графические и текстовые редакторы, средства презентации, системы управления базами данных, электронные таблицы, сетевые информационные технологии. По каждой из этих систем необходимо изучить следующие группы вопросов:

области применения;

теоретические основы, которые включают в себя вопросы, связанные с различными типами данных и их структурированием, методы решения информационных задач с помощью технологических средств данного типа;

технологическое содержание или прикладные программные средства, которые содержат среду, различные режимы работы, систему команд и данные.

Рассмотрим некоторые методические особенности, связанные с вопросами, касающихся преподавания основных тем.

1.10.1. Преподавание текстовых и графических редакторов

Эти вопросы, как правило, изучаются на начальном этапе изучения информатики. Выбор текстового и простейшего графического редакторов, используемых в учебных целях, учебной программой по школьной информатике, как правило, не регламентируется и зависит от технического и программного обеспечения школьного компьютерного класса. В программе перечисляются только вопросы, которые необходимо изучить, а право выбора остается за учителем. Для учебных целей не обязательно использовать профессиональный, достаточно громоздкий редактор «Word», в котором многие элементы его интерфейса оказываются не востребуемыми и создают определенный информационный шум. Все приёмы работы можно изучить на примере более простых текстовых редакторов (Например, WordPad). Освоив программные вопросы на примере какого-нибудь простого редактора, в конце данной темы можно познакомить учащихся с некоторыми возможностями более сложных редакторов (например, «Word»), которых не было в более простом редакторе (например, проверка на наличие в тексте ошибок, работа с таблицами данных и другие возможности).

Необходимо обратить внимание на то, что компьютер с работающим текстовым редактором является специализированным исполнителем для работы с текстовой информацией, который работает под управлением человека. Все его действия (например, ввод символа, копирование фрагмента текста, сохранение текста и т. п.) можно рассматривать как команды управления исполнителем. Отсюда следует, что изучение текстового редактора, как и любого другого прикладного средства информационных технологий, можно проводить по традиционной методической схеме, свойственной изучению всякого исполнителя: данные, среда редактора, режимы работы, система команд.

Текстовый редактор работает с символьной информацией, в которой можно выделить следующие виды структур: символы, слова, строки символов, фрагменты текста (блоки), файлы. Имеются команды для работы с каждой из этих совокупностей данных. Независимо от выбранного конкретного текстового редактора необходимо, прежде всего, выделить и обратить внимание школьников на общие элементы среды, которые характерны для любых таких редакторов. Это, например, рабочее поле, строка состояния и строка меню.

Под режимом работы в текстовом редакторе понимается определенное состояние исполнителя (в данном случае редактора), в котором возможно выполнение определенного вида работы. Это ввод и редактирование текста, его форматирование, орфографическая проверка, обращение за подсказкой, обмен с внешними устройствами, печать и др.

Систему команд текстового редактора можно условно разделить на следующие группы: команды интерфейса, или работа с меню, команды изменения состояния текстового редактора (нижний/верхний регистры, режим вставки / замены и т. д.), команды перемещения по тексту, редактирования, форматирования и др.

Существует множество прикладных программ, предназначенных для работы с графической информацией. Как и при изучении работы с текстом, нигде в учебной программе по информатике не говорится о конкретном программном средстве. Поэтому возникает естественный вопрос, какой графический редактор выбрать для изучения. Этот термин применяется по отношению к прикладным графическим программам, не имеющим какой-либо специализированной ориентации и используемым для произвольного рисования или редактирования сканированных изображений. В соответствии с двумя принципами представления графической информации — растровым и векторным — графические редакторы делятся на такие же два типа. Профессиональные графические редакторы, такие как «CorelDraw», «Photoshop» — дорогие программные продукты и не всегда доступные. С точки зрения учебных целей, стоящих перед базовым курсом информатики, вполне достаточно редактора, например, «PaintBrush» или «Paint», который стал результатом развития первого из типов редакторов. Такой растровый редактор позволяет ученикам наглядно продемонстрировать дискретную структуру рисунка, даёт возможность воздействовать на каждый отдельный элемент при увеличении масштаба в режиме прорисовки и является сравнительно простым и удобным инструментом для работы с растровыми рисунками. Для получения базовых знаний такого редактора будет вполне достаточно. Однако при наличии технических возможностей для углублённого изучения можно посоветовать и другие редакторы.

Особенность изучения компьютерной графики в том, что это такая область информационных технологий, которую ученикам хочется реально увидеть, а не слушать разговоры о ней. Поэтому большое внимание необходимо уделять демонстрации на компьютере разнообразных красочных рисунков, схем, чертежей, диаграмм, образцов анимационной и трёхмерной графики и других графических продуктов. Учителю надо не забыть сказать и о том, что любимые многими из них компьютерные игры в большинстве имеют графический интерфейс, причем достаточно сложный. Чтобы поддержать интерес к математике, школьникам надо сказать и о том, что для получения на компьютере трёхмерного реалистического изображения часто надо выполнить сложные математические расчёты. К теоретическому содержанию этой темы относятся и вопросы представления изображения в памяти компьютера.

Изучение графического редактора, как и любого другого, можно проводить по той же методической схеме, что и текстового редактора.

Итоговой информацией является созданный рисунок, который с позиций растровой графики представляет собой совокупность разноцветных пикселей. Значит, данными для графического редактора являются цвета. Кроме технологии

работы с ними необходимо объяснить, как они представляются в памяти электронно-вычислительной машины.

Пользовательский интерфейс большинства графических редакторов организуется единообразно. Поэтому необходимо, прежде всего, выделить и обратить внимание школьников на общие элементы среды, которые характерны для любых графических редакторов. Кроме панели инструментов, строки состояния, рабочего поля и меню, есть такие, которые присущи только графическим редакторам. Это панель палитры для выбора цвета, калибровочная шкала для установки ширины кисти, резинки и других рабочих инструментов.

Режим работы в таких редакторах определяет возможные действия пользователя, а также команды, которые он может отдавать редактору в данном режиме. Можно выделить следующие режимы: работа с рисунком (рисование), выбор и настройка инструмента, выбор рабочих цветов, работа с внешними устройствами. Для последнего характерно не только работа с файлами, а и сканирование рисунков, их печать.

Систему команд графического редактора можно условно разделить на следующие группы: команды интерфейса, или работа с меню, команды изменения состояния графического редактора (нижний/верхний регистры, режим вставки / замены и т.д.), команды перемещения по тексту, редактирования, форматирования и др.

1.10.2. Преподавания вопросов, связанных с информацией

Блок вопросов, связанных с информацией, охватывает содержание всего курса, поскольку понятие информации является в нем центральным. Это должно быть очевидным хотя бы потому, что с этим термином связано название предмета. Каждое из понятий, связанных с информацией, рассматривается в информатике в двух аспектах: назовем их условно «бескомпьютерным» и компьютерным. Под первым из них понимается рассмотрение информации без привязки к компьютеру, с общих позиций, по отношению к человеку, обществу, природе. Это такие вопросы, как определение и измерение информации, информационные модели и процессы, а также процессы управления в природе и обществе. Под компьютерным аспектом понимается изучение информационной стороны функционирования самого компьютера, изучение компьютерных технологий хранения, передачи и обработки информации, а также методов программирования.

Однако если проанализировать в историческом плане сначала советские, а позже российские и белорусские учебники и учебные пособия по школьной информатике, то можно сделать вывод, что далеко не в каждом из них информации уделяется главное внимание. Этому есть две причины. В первых учебниках центральными понятиями и объектами изучения являются алгоритм и компьютер, а информация упоминается лишь вскользь и в основном определяется на интуитивном уровне. Вторая причина — в объективной сложности понятия информация. Оно относится к числу фундаментальных в

науке, носит философский характер и является предметом постоянных научных дискуссий. Возникает необходимость обсуждения на уроках сложной проблемы определения информации, но в средних учебных заведениях это можно делать только языком, доступным для учащихся.

Кратко проанализируем варианты определения информации в различных учебниках. В учебнике А. П. Ершова такого определения нет вообще. Почти все его содержание посвящено разбору вопросов, что такое электронно-вычислительная машина и что такое алгоритм. В учебнике второго поколения А. Г. Кушниренко утверждается, что этот термин в информатике является первичным, неопределяемым, подобно тому, как отсутствует строгое определение прямой и точки в планиметрии.

Достаточно много внимания уделено этому понятию в учебнике А. Г. Гейна. Сначала в первой главе читаем следующее: «Информация — постоянный спутник человека. Это те сведения, которые помогают ориентироваться нам в окружающем мире». Далее это понятие лишь используется, а в предпоследней главе авторы снова возвращаются к его определению. «В интуитивном, житейском смысле под информацией понимают сведения, знания и т. п., которые кого-либо интересуют. И чем интереснее сообщаемые сведения, тем больше информации (с житейской точки зрения) в них содержится». Дальше даётся другое определение информации с технической точки зрения. «Когда речь идет об автоматической передаче информации, её хранении и переработке, информация — это произвольная последовательность символов, т. е. любое слово. Каждый новый символ увеличивает количество информации». На универсальность претендует определение информации В. А. Каймина. «Информация в наиболее общем определении — это отражение предметного мира с помощью знаков и символов».

Для истории литературы по школьной информатике значительным событием стал выход в России книги «Информатика. Энциклопедический словарь для начинающих». В ней впервые в современном на то время духе отражено всё разнообразие предметной области информатики, её научное содержание, сделана попытка объединить «человеческую» и «техническую» позиции по отношению к информации. Дано следующее определение: «Информация — это содержание сообщения, сигнала, памяти, а также сведения, содержащееся в сообщении, сигнале или памяти». Недосток в том, что просматривается тавтология: в чем разница между содержанием и содержащимися сведениями? Вряд ли это можно понять самому учителю, а тем более объяснить ученику.

Из всего сказанного можно сделать следующий вывод. Если в центр содержания предмета «информатика» ставить информацию, а не алгоритм, компьютер, то обойти вопрос об определении информации нельзя. Бесспорно и то, что дать единое, универсальное определение информации не так то и просто. И в науке, и на практике известны различные подходы к информации, и в рамках каждого из них даётся определение этого понятия. Ученики должны знать, что в

зависимости от контекста, в котором используется термин «информация», он может нести разный смысл.

При раскрытии понятия «информация» с точки зрения субъективного (бытового, человеческого) подхода, следует отталкиваться от интуитивных представлений об информации, имеющих у детей. Не следует сразу требовать от них определения этого термина. Целесообразно начать с беседы в форме диалога, чтобы подвести их к этому определению с помощью простых понятных вопросов. Например, откуда ученики получают информацию? Далее попросите учеников привести примеры какой-нибудь информации, которую они получили в этот день. Например, кто-нибудь скажет о том, что он услышал по радио прогноз погоды. А учитель в это время подведёт ученика к выводу о том, что, получив информацию, он получил некоторые новые сведения о себе или окружающем мире. Таким образом, учитель вместе с учениками приходит к следующему «определению»: информация для человека — это знания, которые он получает от различных источников. На простых, понятных, известных детям примерах следует закрепить это определение. Аналогично можно получить и то, что информация — это содержимое нашей памяти, которая является средством хранения знаний. Но информацию можно хранить не только в собственной памяти, в голове, но и в записях на бумаге, на магнитных носителях и других объектах. Такую информацию можно назвать внешней. Чтобы ей можно было воспользоваться, он должен сначала её найти, прочитать, т. е. обратить её во внутреннюю форму, а затем производить какие-то действия с ней. Этим самым учитель подводит школьников к пониманию того, что у компьютера, подобно человеку, есть внутренняя и внешняя память. Использование дидактического приёма аналогии между информационной функцией человека и компьютера позволит ученикам лучше понять суть устройства и работы электронно-вычислительной техники.

Проблема измерения информации напрямую связана с проблемой её определения, поскольку сначала надо уяснить, что собираемся измерять, а потом — как это делать, какие единицы использовать. Поэтому просматривается следующая цепочка понятий: информация, сообщение, информативность сообщения, единица измерения информации, информационный объём сообщения. Под сообщением понимается информационный поток, который в процессе передачи информации поступает к принимающему его субъекту. Сообщение — это и речь, которую мы слышим, и то, что видим зрительно (фильм, сигнал светофора и т.п.), и текст, который читаем и т. д. Вопрос об информативности сообщения следует обсуждать на примерах, предлагаемых учителем и учениками. Нельзя отождествлять понятия «информация» и «информативность сообщения». Следующий пример показывает различие этих понятий. Вопрос: «Будет ли информативным текст вузовского учебника по высшей математике для первоклассника, если он попытается его прочитать? Иначе говоря, может ли первоклассник с помощью такой литературы пополнить собственные знания?» Очевидно, что ответ отрицательный, хотя, с другой

стороны, такой учебник содержит определенную информацию, но для первоклассника она непонятна.

При объяснении этой непростой темы можно предложить ученикам поиграть в своеобразную викторину. Учитель предлагает детям вопросы, а ученики отвечают на них письменно или ставят знак вопроса, если ответ не знают. После этого учитель даёт правильные ответы, а ученики отмечают, какие из них оказались информативными, а какие — нет. Причём для последних ответов нужно указать причину отсутствия информации: не новое или непонятное. Например, можно задать следующие вопросы: Какой город является столицей Франции? Что изучает коллоидная химия? Какую высоту и вес имеет Эйфелева башня? Большинство учеников ответят лишь на первый вопрос. Учитель должен дать ответы на остальные два вопроса. Информативным является лишь последнее сообщение. Первое из них не является новым, а второе — непонятное для шестиклассника. Если сообщение неинформативно для человека, то количество информации в нем, с точки зрения этого человека, равно нулю, а в информативном сообщении количество информации больше нуля.

Характерным при изучении этой темы является следующий приём. Обсуждая вопрос об измерении информации, тут же переходят к описанию компьютерного представления информации в форме двоичного кода (бита). Во многих учебниках понятие бита объясняется в основном одинаково. Информация кодируется с помощью последовательности сигналов всего двух видов: намагничено или не намагничено, включено или выключено, высокое или низкое напряжение и т. д. Затем утверждается, что количество информации равно количеству двоичных цифр (битов) в таком коде.

Бит — основная единица измерения информации. Следует обратить внимание учеников на то, что в любой метрической системе существуют основные (эталонные) и производные от них единицы (например, метр — сантиметр, километр и др.). Так подходим к определению байта, килобайта, мегабайта и гигабайта. Обязательно надо заострить внимание, что килобайт больше, чем байт не в 1000 раз, как килограмм больше грамма, а в 1024 раза.

Понятие **информационных процессов**, так же как и понятие информации, является базовым в курсе информатики. Сначала на простых понятных примерах надо показать, что это такое. Это получение информации из средств массовой информации, процесс обучения, принятие управляющих решений, разработка какого-нибудь проекта, документооборот на предприятии, сдача экзамена и др. — вот примеры информационных процессов, с которыми нам приходится постоянно иметь дело. Существуют три основных их типа: хранение, передача и обработка информации. Начинать рекомендуется с рассмотрения этих процессов без привязки к компьютеру, т. е. применительно к человеку. Затем, при изучении архитектуры компьютера, компьютерных информационных технологий речь пойдет о реализации тех же информационных процессов с помощью компьютера.

С процессом **хранения информации** связаны следующие понятия: носитель информации (память), внутренняя и внешняя память, хранилище информации. Их следует изучать, проводя аналогию с человеческой памятью.

Биологическую память (мозг) можно рассматривать внутренней, так как её носитель — мозг — находится внутри нас. Её можно назвать также оперативной или быстрой, так как заученные знания воспроизводятся человеком мгновенно. Все прочие виды носителей информации можно назвать внешними (имеется в виду по отношению к человеку). Необходимо показать их примеры в историческом плане: камень, дерево, папирус, бумага и, наконец, магнитные, оптические и другие современные виды носителей информации.

Приведя примеры **хранилищ** информации (библиотека, справочники, картотеки и др.) и рассказав об их основных информационных единицах (анкета, книга, отчет и т.п.), можно дать следующее определение этого термина. Это определенным образом организованная информация на внешних носителях, предназначенная для длительного хранения и постоянного использования. Надо объяснить, что понимается под организацией хранилища (наличие определенной структуры, т. е. упорядоченность, классификация и т.п.) и для чего она нужна (для удобства добавления, удаления и поиска информации). Организованные определённым образом хранилища данных на устройствах внешней памяти компьютера принято называть базами данных.

Рассматривая человеческую память и то, что хранится, например, в библиотеке, надо привести сравнение внутренней и внешней памяти (объём, время доступа, надежность и др.).

Изучая процессы **обработки информации**, необходимо рассмотреть следующие вопросы: общая схема процесса обработки, постановка задачи обработки, исполнитель обработки, алгоритм обработки, типовые задачи обработки информации.

В процессе обработки информации решается следующая задача: дан некоторый набор исходных данных — исходная информация; требуется получить некоторые результаты — итоговую информацию. Тот объект или субъект, который осуществляет обработку, можно назвать исполнителем обработки. Поэтому общую схему обработки информации можно представить в следующем простом и понятном для школьников виде:

Исходная информация → Исполнитель обработки → Итоговая информация

Описание последовательности действий, которую нужно выполнить, чтобы достичь нужного результата, в информатике принято называть алгоритмом обработки. Подробно понятие алгоритма и вопросы алгоритмизации изучаются в соответствующем разделе. Здесь же обращается внимание на то обстоятельство, что одно из фундаментальных понятий школьной информатики, понятие алгоритма исходит от другого базового понятия — понятия информации и, в частности, понятия информационных процессов.

В форме беседы с учениками необходимо рассмотреть примеры ситуаций, связанных с обработкой информации, и на этих примерах показать выше

перечисленные понятия. Все примеры обработки информации можно разделить на три типа.

Первый тип заключается в получении новой информации, нового содержания знаний. К ним относится, например, решение математических задач, действия следователя, который по набору улик находит преступника, разгадывание учёным тайн древних рукописей;

Ко второму типу обработки информации относится обработка, в результате которой меняется форма, а не содержание информации. К этому типу можно отнести перевод текста с одного языка на другой, кодирование информации, её структурирование, связанное с внесением определенного порядка, определенной организации в хранилище информации (сортировка, группировка по некоторым признакам классификации, табличное или графическое представление и т. п.);

Можно выделить третий тип обработки — поиск информации. Задача поиска формулируется так: в хранилище информации (телефонный справочник, словарь, расписание поездов и т.п.) найти нужную информацию, удовлетворяющую определённым условиям поиска (номер телефона конкретного товарища, английский перевод конкретного слова, время отправления поезда в определенный пункт назначения). Эти и другие примеры поиска, как и примеры других хранилищ информации должны привести сами школьники.

Ключевыми понятиями в описании процесса **передачи информации** являются источник и приёмник информации, информационный канал. Необходимо это изобразить схематически. На естественных примерах разговора, чтения можно обратить внимание на то, что в этом процессе информация представляется и передаётся в форме последовательности сигналов, символов, знаков и показать, что является каналом связи (акустические волны в атмосфере, световые электромагнитные волны). Но надо не забывать, что такие примеры доступны старшеклассникам после изучения соответствующих тем в физике. Для младшего возраста лучше привести более простые примеры.

Если в процессе передачи информации используются технические средства связи, то их называют каналами передачи информации (информационными каналами). К ним относятся телефон, радио, телевидение и др.

В этом вопросе можно рассмотреть также такие понятия, как кодирование и декодирование информации, шум и приёмы защиты от него, скорость передачи информации и пропускная способность канала. При обсуждении последних двух вопросов можно привлечь аналогию с процессом перекачки воды по водопроводным трубам.

1.10.3. Методика изучения систем счисления и представления информации

Тема «системы счисления» является смежной с математикой. Она имеет прямое отношение к математической теории чисел. Но в школьном предмете математики она, как правило, не изучается.

В первых учебниках рассматриваемое понятие не упоминается совсем. Говорится лишь о том, что вся информация представляется в двоичном виде. Среди учебников второго поколения наибольшее внимание системам счисления уделено в книге «Основы информатики и вычислительной техники» под редакцией А. Г. Гейна. В отдельном параграфе дано следующее определение: «Система счисления — способ записи чисел с помощью заданного набора специальных знаков (цифр)». Такое определение затрагивает только алфавит, синтаксис и семантику языка чисел. Более полное определение дано в словаре школьной информатики под редакцией А. П. Ершова: «Система счисления — способ изображения и соответствующие ему правила действий над числами». Под правилами действий понимаются способы выполнения арифметических вычислений в рамках данной системы счисления. Их можно назвать прагматикой языка чисел.

В этой теме необходимо рассмотреть следующие вопросы:

- позиционные и непозиционные системы счисления;
- основные понятия позиционных систем: основание и алфавит;
- развёрнутая форма представления чисел в позиционных системах;
- перевод чисел из одной системы счисления в другую;
- особенности арифметики в двоичной системе счисления;
- связь между двоичной и шестнадцатеричной системами счисления.

С методической точки зрения эффективно подвести учеников к самостоятельному, хотя и маленькому, открытию. Желательно, чтобы ученики сами сформулировали различие между позиционным и непозиционным принципом записи чисел. Для этого пишем на доске два числа, например, \overline{XXX} (римское число тридцать) и 333 (арабское число триста тридцать три). В форме диалога выясняем, что означает каждая цифра в каждом из примеров и как получается значение каждого из чисел. Из этого диалога следуют отличия предложенных двух принципов записи чисел. Закрепив это на других примерах, можно ввести следующее определение: «Система счисления — это определенный способ представления чисел и соответствующие ему правила действий над числами».

Предложите выполнить, например, умножение двух трёхзначных чисел в одной и другой системах счисления. Выполняя это задание, ученики должны понять следующее. В непозиционной римской системе счисления, в отличие от позиционной арабской нет простых и понятных правил для выполнения вычислений с многозначными числами.

Теперь также в форме беседы нужно дать понять ученикам, что позиционных систем существует множество, и отличаются они друг от друга алфавитом — множеством используемых цифр. Размер алфавита или количество используемых цифр — основание системы счисления. Далее желательно

попросить учеников дать значения этих терминов для десятичной системы счисления, вместе разобраться, почему она так называется, привести примеры других позиционных систем счисления. Особое внимание необходимо уделить двоичной системам счисления.

Прежде чем выполнять вычисления в различных системах счисления, рекомендуется сначала научить учеников записывать натуральный ряд чисел в различных позиционных системах счисления. При этом желательно проводить сравнение с десятичной системой счисления. Показываем, как с помощью индексов записать число в различных системах счисления, что существенно. Но здесь нужно сделать следующее важное замечание: ни в коем случае нельзя называть недесятичные числа так же, как десятичные. Например, 36_8 — это не тридцать шесть, а три — шесть восьмеричное; 101_2 — это не сто один, а один — ноль — один двоичное.

1.10.4. Изучение графических возможностей языка «Pascal»

Графические возможности «PascalABC» изучаются в теме «Основы алгоритмизации и программирования» после основных простейших понятий программирования (среда языка, переменные и типы данных, ввод, вывод, присваивание, работа с целым типом и составление линейных программ). Изучать графические возможности его, как и программирование, авторы учебников рекомендуют на конкретных примерах. Примеры графических программ используют только константы в качестве координат, хотя понятие переменной уже было введено и ранее. При объяснении графических программ понятие процедура и функция не используются.

Необходимо обратить внимание на то, что графические команды (вместо термина “процедуры”) можно разделить на несколько видов:

команды, задающие параметры рисунков: для пера это, например, «SetPenColor», «SetPenWidth», «SetPenMode» и др. Для кисти это, например, «SetBrushColor» и «SetBrushStyle»;

команды, рисующие геометрические фигуры: «SetPixel», «MoveTo», «Circle», «Rectangle», «FloodFill» и другие;

команды, задающие параметры шрифта: «SetFontSize», «SetFontColor», «SetFontStyle»;

команда для вывода строки: «TextOut»;

команды для работы с графическим окном: «SetWindowSize», «SetWindowTop», «ClearWindow», «LoadWindow» и другие.

Необходимо заметить, что команды для задания значения цветов, стилей пера, кисти и других параметров, а также команды рисования такие же, как в «Delphi» или в «Builder». Существенное отличие в том, что в названных системах соответствующие процедуры и функции являются не самостоятельными, как в «PascalABC», а членами класса, то есть методами.

При изучении этой темы не обязательно как по основам алгоритмизации и программирования, в которых большое количество элементов, изучить все элементы. Надо уметь рассортировать их. Предлагается выделить то, что является обязательным, наиболее простым и понятным, с чего следовало бы начинать изучение темы. Например, задание размера графического окна есть более важная тема, чем команды, устанавливающие его положение на экране.

Вводятся и объясняются такие понятия, как пиксель, разрешающая способность (разрешение) экрана, координаты пикселя. Необходимо уделить особое внимание компьютерным координатам, так как они отличаются от декартовых. Для этого можно предложить школьникам такие упражнения: показать точку с указанными координатами на экране в компьютерных координатах и в декартовых координатах, указать координаты заданной точки в декартовой системе координат и в компьютерной системе координат. На компьютере полезно синхронно с преподавателем подготовить и выполнить следующую программу рисования точки с заданными координатами.

При объяснении особое внимание надо обратить на команду «*Uses GraphABC*» и провести компьютерный эксперимент, задавая разные разрешающие способности, меняя координаты точки и её цвет. Можно нарисовать несколько точек, не используя команду цикла.

Для более одарённых детей можно эту же программу так. Для координат объявить две переменные (x , y), ввести их, и в левой половине окна точку нарисовать одним цветом (например, красным), а справа нарисовать другим цветом. Окно можно разбить на четыре прямоугольника.

Следуя программе и учебнику, дальше надо научить рисовать линии разным цветом и толщины, треугольник и другие многоугольники из линий, включая и прямоугольник, и окружность. При этом вначале с помощью «*Line*» или «*LineTo*» рисуем только их границы, а затем показываем, как закрасить эти фигуры с помощью «*FloodFill*». Надо учесть, что объективно это очень сложная команда, тем более для школьников. Параллельно с рисованием графические элементы подписываем. Для закрепления этих навыков в учебнике составлена программа для рисования домика, состоящего из треугольника, прямоугольника и круга.

Можно сделать больше программ, если предложить ученикам сразу синхронно с записью на доске готовить программу на компьютере. Это непростой вид занятия, так как школьникам трудно будет сосредоточиться, чтобы слушать объяснения, комментарии учителя и одновременно работать за компьютером. С одной стороны, можно успеть развить творческое мышление, если учащиеся заинтересованы в изучении данных вопросов. Однако, для более слабых учащихся лучше предложить сначала освоить базовые упражнения и лишь потом переходить на более сложные задачи при наличии времени.

Для закрепления материала предлагается выполнить одно общее или два (три) варианта задания одного или разных уровней сложности самостоятельно в классе. При такой форме проведения урока или его части можно обсудить алгоритм или другие общие вопросы, часто встречающиеся ошибки.

1.10.5. Некоторые особенности изучения других разделов

Рассмотрим теперь **основы разработки сайтов**. Итоговой информацией является созданный сайт, который с позиций разработчика представляет собой совокупность разноцветных страниц, снабжённых различной по содержанию информацией. Для полноценного сайта обычных занятий может быть недостаточно ввиду большого количества возможностей самого языка написания. По этой причине достаточно ограничиться лишь изучением определённых из этих возможностей. Также следует обратить внимание на сам язык разметки. Это понятие очень своеобразно, поскольку предполагает взаимосвязь названия тега и некоторого объекта (или образа), который за ним «закреплён».

Пользовательский интерфейс большинства редакторов организуется не единообразно. Множество не только редакторов, но и современных «онлайн-технологий» имеют свою специфику, но сами теги несут в себе определённую структуру, на которую и нужно обратить внимание. Некоторые редакторы «хранят в себе» уже готовые теги, что может существенно упростить разработку сайтов во время занятий.

Режим работы в таких редакторах определяет возможные действия пользователя, а также команды, которые он может отдавать редактору в данном режиме. Можно выделить следующие режимы: разработка сайта, тестирование сайта, выбор тегов для добавления объекта (если эта возможность предусмотрена редактором) и др. Однако, некоторые редакторы (например, блокнот) предполагают знание самих тегов и создание сайта без дополнительных возможностей редактора.

Данная тема представляет собой большое поле для эксперимента не только с графическим дизайном, но и для практической направленности предмета. Современные школьные системы довольно часто предполагают наличие в школе определённого сайта с основной информацией. Разработкой или дополнением таких сайтов могут заняться и учащиеся при наличии времени.

Развитие технологий, связанных с понятием **Интернета**, имеет огромные возможности, касающиеся поиска и обработки информации. Практическая часть здесь имеет больше преимуществ и поэтому эффективность построения занятий в большей степени зависит именно от практической работы учащихся, хотя теоретические понятия никто не отменял. Основные вопросы, поставленные на изучение, связаны с изучением локальной и глобальной сетей, их организацией, развитием, представлением данных в сети интернета, а также использованием его ресурсов.

Теоретическое обоснования понятие о всемирной сети и Интернета представляется частично затруднительным в полной его сущности, поскольку включает в себя такие понятия, как протокол, цифровая адресация, домен, имя сервера и другие. Полный цикл передачи информации или её поиск в сети также сводится лишь к частичному рассмотрению.

При наличии возможности выхода в Интернет, деятельность учащихся можно разнообразить следующими видами работы: обработка и приём электронной почты, обращение к серверам и извлечение файлов, поиск информации в сети Интернет.

Что же касается вопросов, связанных с **базами данных**, то можно выделить следующие вопросы для изучения: организация применения информационных систем и баз данных, их классификация, структура реляционных баз данных, назначение системы управления баз данных.

В курсе базового образования можно отметить тот факт, что основная часть работы происходит в среде Access. И тем не менее теоретические основы данного курса всё-таки требуется рассказать. Основные области применения связаны с работой в архивах, картотеках, где основной задачей является грамотное структурированное хранение различных данных с целью их получения при введении данного запроса. Можно провести небольшую параллель при сравнении некоторых принципов работы баз данных и интернета.

Основные понятия, которые рассматриваются с позиции баз данных, имеют некоторую аналогию с понятием множества в математике, что можно отметить при изучении курса в профильных классах. На конкретных примерах поиска некоторых данных лучше сначала показать возможности, связанные с базами данных, и уже после рассмотрения перейти к постепенному изучению основных понятий.

Практические навыки при изучении данной темы могут быть связаны с задачами теоретического характера, которые имеют цель закрепить основные понятия и термины по данной теме. С другой стороны, практическая работа по поиску необходимых данных в различных средах включает в себя требования, связанные с комплексным владением теоретическим материалом. Индивидуальные задания по данной теме в рамках базового образования не имеют широкого распространения в связи с ограниченностью во времени при изучении данных вопросов.

Что же касается **электронных таблиц**, то можно выделить следующие аспекты: области применения электронных таблиц, системы команд при работе с таблицами, содержимое ячеек и работа с ним. С одной стороны, работа с таблицами связана с вычислениями, которые заменяют ручное управление, что значительно экономит время выполнения работы. С другой стороны, удобное представление данных в таблицах позволяет при умелой сортировке работать с этими данными быстрее. Работая с таблицами, можно указать на некоторую аналогию, которая наблюдается при работе с базами данных.

Важным объектом является ячейка таблицы. Это то, что находится на пересечении столбца и строки таблицы, и хранит в себе некоторую информацию. При работе с ними важно указывать на взаимно-однозначное определение ячейки, которое видно в её имени, состоящем из имени столбца и номера строки. Работа с данными в ячейках, как с некоторыми значениями имеет свои особенности. Интерпретация данных как выражений или чисел даёт нам

возможность использования формул, в которых будут фигурировать имена ячеек и действия, которые будут характеризовать операции, связанные с ними.

Практическая работа с таблицами различна. Деятельность может быть связана, с одной стороны, с обработкой самих данных таблицы, например, обработка ячеек, редактирование данных в ячейках. Но можно проводить обработку и самой таблицы, связанную, например, с удалением строк или столбцов в ней.

Использование таблиц при построении математических моделей является комплексной задачей, которая может быть рассмотрена на факультативных курсах в рамках интегрированных уроков.

Презентации – это одна из тем, которая важна при изучении, поскольку дальнейшее развитие любой сферы включает в себя показ разработок и идей. Презентации являются очень удобным и весьма распространённым способом такой деятельности.

Основные вопросы, касающиеся презентаций, включают в себя создание и обработку презентаций, основные формы представления информации в презентации, структура презентаций её основные элементы.

При проведении практических занятий учащиеся могут освоить не только работу с самим представлением данных на показ, но и ещё раз повторить графические возможности соответствующих редакторов. Эта тема подходит для художественного творчества и развивает эстетическое восприятие у учащихся. Грамотное представление данных на экране развивает у учащихся такие виды качества, как рациональность и аккуратность.

При организации практических занятий имеет смысл использовать и групповую работу при разработке представлений информации или разработке некоторых совместных проектов. Теоретические знания по соответствующей теме играют второстепенную роль ввиду огромной полемики для творчества и поля для фантазий учащихся.

Очень важно указать на то, что далеко не обязательно все данные размещать на слайдах. При чрезмерном обилии повторов отходит необходимость в показе данных в презентации. Также можно указать на возможность в некоторых редакторах указывать заметки к слайдам, которые показаны для рассказчика, но не видны людям, наблюдающим процесс показа.

2. ПРАКТИЧЕСКИЙ РАЗДЕЛ

2.1. Вопросы для управляемой самостоятельной работы

В данном пункте приведены вопросы для обсуждения на *семинарских занятиях*, касающиеся построения теоретического курса в рамках изучения дисциплины «Методика преподавания информатики».

Представьте, что вы в шестом классе рассказываете о современном развитии техники и информации. Какую информацию вы предложили бы ученикам?

Как бы вы своими словами рассказали об основных компонентах, которые играют первостепенную роль при формулировке определения «Информатика»?

Можно ли однозначно сказать, чем именно занимается информатика в современном мире? Относится ли к информатике робототехника, механика, программирование?

Каким образом вы бы сформировали тематику содержания образования? Чем вы бы руководствовались при выборе соответствующих тем?

Какие темы, на ваш взгляд, никогда не потеряют актуальность в информатике? Можно ли однозначно определить разделы, которые должны быть в программе? Какие из них наиболее важны при профессиональной подготовке учащихся?

С какого класса, на ваш взгляд, следует изучать информатику в общеобразовательных учреждениях? Какие темы вы бы внесли в список для изучения на факультативных занятиях или кружках?

Какие темы рефератов наиболее актуальны в современной информатике и методике преподавания информатики?

Какие методы и подходы вы бы использовали при проведении занятий на соответствующие темы?

Есть ли в информатике темы, которые можно изучать без повторения? Если да, то какие? Если нет, то почему?

В современных условиях, какой должен быть уровень работы с компьютером у каждого человека? А в рамках углублённого изучения предмета?

Какие основные цели на примере конкретных тем вы можете предложить при изучении материала в разных классах? Какова на ваш взгляд основная цель при изучении разделов, связанных с алгоритмизацией и программированием?

Какие основные требования по компьютерной грамотности можно предъявлять при обучении школьников информатике в средней школе? А на углублённом уровне? Есть ли однозначная грань между современным состоянием технического развития и уровнем компьютерной грамотности населения?

Какие основные принципы дидактики применимы в различных разделах информатики? Приведите примеры.

Как принципы дидактики проявляются при преподавании различных тем по информатике? Приведите примеры.

Какие методы обучения следует использовать на уроках, посвящённых алгоритмизации и программированию? А основам работы с компьютером и основным устройствам, связанным с ним?

Какое должно быть оптимальное количество членов бригады в бригадной форме работы? Кто должен руководить бригадой? Постоянным ли должен быть состав бригады? Приведите примеры в зависимости от изучаемой темы и состава класса.

Оба ученика должны быть одного уровня при парной работе за электронно-вычислительной машиной или нет? Слабый ученик должен заниматься с сильным учеником или нет? Постоянными ли должны быть пары при такой работе или нет? Как оценивать результат парной работы за вычислительной техникой? Как правильно распределить роли двух учащихся, работающих за одним компьютером?

Какое должно быть соотношение теоретического материала и примеров решения заданий в лекциях по конкретным темам из методики преподавания основных тем? Какой оптимальный объём материала, на ваш взгляд, лучше рассматривать на лекциях? В каком порядке лучше рассматривать вопросы по различным темам? Приведите примеры.

Кто определяет уровень сложности при выборе индивидуальных заданий, сам учащийся или учитель? Как найти такой уровень задания, чтобы задание было не слишком простое и, в то же время, посильным? Как оценивать работу учащихся, если задание сложного уровня он не выполнил, а с более простым заданием смог бы справиться?

Какие бы вопросы об истории информатики вы бы предложили использовать в конкурсе «Инфомышка»?

Насколько актуальным является вопрос о временных ограничениях работы за компьютером? На ваш взгляд, какие меры могут способствовать грамотному обращению с компьютером и вычислительной техникой не в ущерб собственному здоровью? Готовы ли вы выполнять эти меры? Готовы ли люди выполнять такие меры?

Как распределять варианты заданий при распределении индивидуальных заданий учащимся? Всем ли следует давать задания одинаковой сложности внутри одной темы или учитывать индивидуальные способности учеников? Одинаковым ли должно быть количество заданий по темам или нет? Задания по некоторым темам следует ли рекомендовать только наиболее «сильным» ученикам или нет?

Какие методические особенности, на ваш взгляд, вы бы выделили при разработке уроков по теме «Системы управления базами данных»? А по темам «Анимация», «Векторная графика», «Работа с текстом», «Презентации», «Электронная почта», «Вирусы и защита информации», «Робототехника и её современное состояние»?

2.2. Практикум по информатике

В этом пункте приведены практические задачи для *семинарских занятий и самостоятельной работы*, при решении которых можно рассмотреть некоторые особенности конкретной темы по информатике.

1. Придумайте свои вопросы и тесты, которые вы могли бы предложить в качестве контроля знаний по конкретным темам.

2. Разработайте простейшую модель базы данных таким образом, чтобы на её примере можно было бы рассмотреть основные понятия, связанные с этой темой.

3. Разработайте такую электронную таблицу данных, чтобы с её помощью можно было продемонстрировать основные особенности и свойства таблиц.

4. Найти значение многочлена с заданными в виде массива коэффициентами, вычисляя и суммируя значения одночленов. Как, используя данный пример, показать основные достоинства и недостатки работы с массивами?

5. Запрограммировать управляемое движение небольшого текста или объекта с возможностью выбора параметров движения. Показать на данном примере основные особенности при реализации комбинированных задач по программированию.

6. Построить график какой-нибудь функции в любом из изучаемых программных пакетов. Предусмотреть возможности изменения различных параметров, связанных с графиком (изменение масштаба графика вдоль различных осей, изменение толщины графика и др.) и показать особенности редактора, связанные с построением графика.

7. Числовой массив вывести в виде вертикальной диаграммы, предусмотрев возможность изменения масштаба диаграмм. Показать основные достоинства и недостатки построения диаграмм в данном пакете.

8. Проверить программу на правильность расстановок скобок. Показать основные возможности программирования, связанные с данной задачей.

9. Пусть в тексте разделителем между словами является пробел. Найти все слова, которые начинаются с некоторой буквы, и заменить эти слова на цифру 1. Показать на примере данной задачи основные возможности и особенности работы с текстом на примере любого языка программирования.

10. В тексте найти слова, которые начинаются и заканчиваются на одну и ту же букву. Показать на примере данной задачи основные возможности языка при анализе слов в тексте.

11. Для заданной квадратной матрицы найти произведение сумм элементов её строк. Показать на примере данной задачи достоинства и недостатки разработки процедур и функций в языке программирования «*Pascal*».

12. Построить кусочно-непрерывную функцию, используя графические возможности в языке программирования «Pascal». Показать на этом примере достоинства и недостатки различных режимов и способов работы с графикой.

13. Построить непрерывную функцию, используя графические возможности в языке программирования «C++». Показать на этом примере достоинства и недостатки способов работы с графикой.

14. Построить непрерывную функцию, используя графические возможности в языке программирования «Delphi». Показать на этом примере достоинства и недостатки способов работы с графикой.

15. Привести пример использования стандартных функций в одном языке программирования и разработать в другом языке программирования функцию, которая решала бы в этом языке такую же задачу, если такой функции там нет.

16. Показать на примере решения одной задачи использование различных видов циклов и роль изменения структуры алгоритма при различном использовании циклов.

17. Рассмотреть алгоритм одной из сортировок. Можно ли её расщепить на шаги, которые можно будет рассказывать на уроках в общеобразовательных учреждениях? А на углублённом уровне?

18. Для заданной квадратной матрицы определить, существует ли у неё такой столбец, номер которого показывает такую же строку в ней? Если да, то вывести все такие номера, если нет, то вывести текст, показывающий отсутствие таких номеров. Показать на примере данной задачи особенности работы с матрицами на примере любого языка программирования.

19. Построить для двух заданных уравнений множеств точек на плоскости их объединение или пересечение. При рассмотрении полученного множества точек рассмотреть особенности среды, в которой получен результат.

20. Пусть на плоскости дано множество точек и множество окружностей. Найти две различные точки из заданного множества такие, что прямая, проходящая через них пересекает наибольшее число окружностей из множества данных окружностей. Указать сложности, которые возникли при разработке алгоритма решения задачи и класс, в котором данную задачу можно было бы рассмотреть на факультативном занятии.

2.3. Содержание реферата (доклада с презентацией) по частной методике

Во введении необходимо дать общую характеристику одной или нескольких тем согласно действующей в настоящее время учебной программе:
в каких классах (в каком классе) изучается;

в каких четвертях (четверти);
общее количество часов и распределение по классам;
что по разделу, к которому принадлежит рассматриваемая тема, или по родственной теме школьник должен знать перед изучением темы;
желательно дать кратко историю изучения темы.

1. Содержание, то есть ЧТО необходимо изучить. Перечислить вопросы и объём в часах по каждому из них согласно учебной программе. Указать, есть ли этот материал в учебном пособии, соответствует ли он программе. Пересказать с необходимыми комментариями содержание учебника, какие понятия и в какой последовательности вводятся и как они объясняются, какие используются для этого примеры. Если вопросы тривиальные (например, операторы языка PascalABC, работа в текстовом редакторе), то эта часть должна быть предельно краткой. Для более сложных тем (векторная графика, некоторые вопросы Excel, СУБД, Интернет и т.п.), которые не изучались студентами ни в средних учебных заведениях, ни в университетах, необходимо дать более подробное описание.

2. Методика изучения темы, то есть КАК изучать:

организационные формы проведения каждого урока (беседа, лекция, выполнение общих или индивидуальных заданий и т.п.);

индивидуальная или групповая работа;

используется ли и как, для каких целей *компьютер* и другие технические средства обучения, сколько времени, количество одновременно используемых компьютеров и др.

Некоторые основные варианты использования компьютера:

1) последовательное использование компьютера., то есть учитель сначала объясняет без компьютера, а затем школьники выполняют это же (или с изменениями) в электронном виде;

2) синхронное использование компьютера вместе с учителем;

3) объяснение новой темы, например, с использованием презентаций;

4) объяснение условия задачи, например, программирование графики;

5) выполнение индивидуальных заданий;

6) контроль и оценка знаний;

7) выполнение заданий с использованием учебника или других пособий;
и др.

Не обязательно все рекомендованные варианты использовать.

Можно добавить свои.

3. Дидактический материал (примеры). Если надо, привести конкретные примеры, которые будете использовать

1) для объяснения нового материала;

2) для его закрепления;

3) для самостоятельной работы с указанием, общие примеры, несколько (2-3) вариантов или у каждого школьника свой пример;

4) примеры для домашнего задания.

Необходимо указать, достаточно ли примеров в учебнике или надо придумывать свои. Для более высокой оценки желательно привести примеры, которых нет в действующем учебнике.

Желательно по возможности предусмотреть задания разных уровней.

4. Система контроля и оценки знаний.

Когда, сколько времени проводите текущий опрос, устно или письменно, привести конкретные вопросы, задания, если надо, примеры для проверки.

Тесты и как их будете оценивать.

Самостоятельные работы: когда, время, конкретные задания и критерии оценок.

Контрольная работа согласно календарно-тематического планирования с конкретным наполнением, если она предусмотрена по программе.

Не надо приводить из интернета стандартные критерии оценок для всех предметов, а указать, как вы будете оценивать знания и умения по вашей конкретной теме.

5. Для одного урока привести подробный план конспект его проведения с соответствующими требованиями.

Замечания. Приведенные выше четыре пункта можно рассказывать (описывать) последовательно или параллельно. Обратит внимание на вопросы, которые, возможно, некоторая часть школьников знает.

2.4. Темы рефератов (докладов с презентацией) по частной методике

Подготовить реферат (доклад с презентацией) по следующим темам: "Методика преподавания указанных тем по информатике для учреждений общего среднего образования".

1. Обработка растровых изображений в 6-м классе (5 час).
2. Создание текстовых документов в 6-м классе (5 час).
3. Компьютерные презентации в 6-м классе (8 час).
4. ОАиП (основы алгоритмизации и программирования) в 6-м классе: алгоритмы и исполнители (8 час).
5. *Представление о логике высказываний. Множество и операции над ними в 7 классе. (5 час).*
6. *ОАиП в 7-м классе: основные алгоритмические конструкции (первые уроки для изучения алгоритмизации с использованием Робота).*
7. ОАиП в 7-м классе: основные алгоритмические конструкции (остальные уроки для изучения алгоритмизации без использования Робота).
8. Аппаратное и программное обеспечение компьютера в 7-м классе (5 час).
9. *Работа с векторной графикой в 7-м классе.*
10. Технология обработки аудио- и видеоинформации в 8 кл. (5 час).
11. Основы анимации в 8 кл. (8 час).
12. Основы алгоритмизации и программирования в 8 кл

13. Технология обработки текстовых документов в 8 кл. (8 час).
14. *ОАиП в 9 кл.: алгоритмы обработки строковых величин (8 час).*
15. Обработка информации в электронных таблицах в 9 кл.
16. *Компьютерные информационные модели в 9 кл.*
17. *ОАиП в 10 кл. на базовом уровне: алгоритмы обработки массивов.*
18. Хранение и обработка информации в базах данных в 10 кл. на базовом уровне.
19. *Компьютер как универсальное устройство обработки информации. в 10 кл. на базовом уровне (8час).*
20. ОАиП в 11-м кл. на базовом уровне: введение в объектно-событийное программирование (8 час).
21. Основы веб- конструирования в 11-м кл. на базовом уровне.
22. *Компьютерное моделирование в 11-м кл. на базовом уровне.*
23. *Содержание и методика преподавания информатики в 10-м классе на повышенном уровне.*
24. *Содержание и методика преподавания информатики в 11-м классе на повышенном уровне.*
25. Методика преподавания вопросов, связанных с информацией
26. Методика преподавания вопросов, связанных с коммуникационными технологиями

3. РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ

3.1. Типы и примеры тестов

1. Термин “Информатика” возник в начале 20-го века;
одновременно с появлением больших ЭВМ;
в начале 60-х годов 20-го века;
одновременно с появлением персональных ЭВМ;
одновременно с появлением объектно-ориентированного программирования.
2. Когда началось массовое преподавание в школах основ информатики и вычислительной техники в бывшем Советском Союзе?
3. Когда в Республике Беларусь дисциплину “Информатика” перенесли в базовую школу в 8 – 9 классы?
4. С какого года информатика изучается, начиная с 6-го класса? Введите четыре цифры года.
5. В каком году в педагогических институтах начали преподавать методику преподавания информатики? Введите четыре цифры года.
6. Выберите словесно-фронтальные методы обучения информатике из предложенных.
7. Расположите следующие уровни работы с компьютером по возрастанию уровня подготовки, указав без пробелов номера в нужном порядке: 1) программирующий пользователь, 2) пассивный пользователь 3) системный программист 4) программист.
8. Установите соответствие, например:
понимание общих принципов работы ЭВМ (а не частностей!);
умение записывать алгоритм, включая написание и отладку программы;
объективное отношение к результатам компьютерных вычислений, т. е. критичность и самокритичность в оценке своих способностей.
А. Признак логико-алгоритмического мышления.
В. Признак общего образования при изучении информатики.
С. Признак логико-алгоритмического мышления.
Другие тесты получены аналогично на основе теоретических экзаменационных вопросов.
9. В каком классе вводится и объясняется термин, понятие “...”?
Например, системный блок, системная (материнская) плата, процессор и его разрядность, оперативная память (ОЗУ), постоянное запоминающее устройство (ПЗУ), винчестер, адаптер, шина, сканер, веб-камера;
пиксель, разрешающая способность (разрешение) экрана;
презентация, компьютерная презентация, слайд;

интерфейс, операционная система, рабочий стол, панель задач, ярлык, панель управления, панель инструментов, файл, расширение файлов, путь к файлу, папка, файловый менеджер, проводник;

идентификатор, переменная, константа, массив, одномерный массив, двумерный массив, матрица, индекс одномерного массива, размер одномерного массива;

и др.

10. Что означает термин "...” (см. 9)?

11. В каком классе по программе должны начинать изучать "...”? Например: вставку картинок из коллекции клипов в презентацию, вставку рисунков из файла в презентацию, использование декоративного (фигурного) текста в презентации, использование фигур (линии, звёзды, ленты, выноски и т.п.) в презентации, использование элементов мультимедиа в презентации, использование эффектов в презентации;

создание папок, поиск папок и файлов, копирование папок и файлов, перемещение папок и файлов, удаление папок и файлов;

тип *Real*, *Integer*, перечислимый тип, записной тип, символьный тип, строки и работа с ними; задание одномерного массива в виде констант при объявлении;

операторы присваивания, *read*, *readln*, *Write*, *Writeln* безформатного вывода, *Write*, *Writeln* с форматным выводом, *if*(полная и сокращённая форма), *case*, *for*, *while*, *repeat*, вложенные циклы;

операции логические, арифметические обычные +, -, *, /, ^, целочисленного деления *div*, *mod*, операции сравнения, битовые операции;

стандартные функции и процедуры (*abs*, *sqr*, *sqrt*, *sin*, *cos* и др.), процедуры и функции пользователя, классы, объекты и другие элементы объектно-ориентированного программирования;

ввод массива с экрана; вывод массива на экран;

алгоритмы сортировки массива; нахождения количества, суммы чисел массива, удовлетворяющих некоторому простейшему условию; нахождения наибольшего (наименьшего) числа массива; преобразования массива (например, положительные числа увеличить в 2 раза, а отрицательные уменьшить в 10 раз); построения вертикальной диаграммы по заданному одномерному массиву; поиск в массиве; задание массива случайным образом.

Могут быть и другие алгоритмы, понятия, вопросы.

12. В каком классе изучается конкретная тема и (или) каков её примерный объём в часах? Смотри программу по информатике для учреждений среднего образования для школьников.

Тесты могут быть открытыми или с вариантами ответов.

При ответах на тесты необходимо иметь в виду, что некоторые вопросы, термины, понятия, элементы языка *PascalABC* и других систем в учреждениях среднего образования согласно действующей сквозной шестилетней программе могут не изучаться на базовом уровне или не существуют вообще.

Тестирование можно использовать не только во время проверки знаний (контрольные работы, зачёты, экзамены и т.п.), но и во время обычных занятий,

как правило, без использования компьютера. Основная их особенность в том, что проверку правильности ответов на них можно формализовать и поэтому поручить компьютеру. Т

3.2. Экзаменационные вопросы

1. Что такое информатика?
2. История преподавания информатики в СССР и РФ.
3. Цели и задачи учебной дисциплины «Методика преподавания информатики».
4. Каковы особенности информатики и методики её преподавания по сравнению с другими дисциплинами?
5. Назовите уровни работы с компьютером и дайте им краткую характеристику.
6. Цели преподавания информатики в учреждениях среднего образования.
7. Назовите характерные особенности компьютерной грамотности, образованности, информационной культуры.
8. Что такое принцип научности в преподавании информатики? Объясните фундаментальные понятия информатики: “информация”, “алгоритм”, “исполнитель”.
9. Какие особенности сознательности усвоения и деятельности в информатике?
10. Как достигается доступность и наглядность в информатике?
11. Особенности активности и самостоятельности при изучении информатики.
12. Как реализуется принцип коллективности и индивидуализации?
13. Связь теории и практики, проблема эффективности учебной деятельности.
14. Обзор методов обучения информатике, их классификация.
15. Характерные особенности словесно-фронтальных методов в информатике.
16. Как реализуются анализ, синтез, сравнение и классификация? Приведите примеры обобщения, индукции и дедукции. Особенности аналогии и переноса, абстракции и конкретности в информатике.
17. Как влияет использование компьютера на организационные формы обучения?
18. Какие формы обучения и их особенности при изучении информатики?
19. Какие особенности основных организационных форм обучения в информатике?
20. Какие особенности вспомогательных организационных форм обучения?
21. Требования к компьютерным классам. Специфика работы учителя в условиях локальной компьютерной сети, наличия электронной почты и Интернета.

22. Преимущества и недостатки программированного обучения.
23. Подготовка учителя к уроку, варианты занятий без компьютеров и с их использованием. Методические аспекты повышения эффективности урока.
24. Дифференцированный и индивидуальный подходы при обучении информатике. Принципы подбора индивидуальных и групповых заданий.
25. Управление и контроль за работой обучаемых.
26. Методы контроля знаний, их особенности.
27. Электронное и бумажное (безмашинное) тестирование, преимущества и недостатки тестирования.
28. Особенности построения сквозной программы по информатике для VI — XI классов.
29. — 34. Какие основные разделы информатики, и в каком объёме должны изучаться по действующей в настоящее время программе в VI классе? в VII классе? в VIII классе? в IX классе? в X классе? в XI классе? Любой раздел опишите подробно.
- 35 — 40. Что и в каком объёме должно изучаться по действующей в настоящее время программе в разделе ОАиП в VI классе? в VII классе? в VIII классе? в IX классе? в X классе? в XI классе? Приведите примеры задач и упражнений.
41. Какие методические особенности преподавания раздела “Основы алгоритмизации и программирования”?
42. Какие методические принципы используются при изучении ОА и П?:
43. Методика преподавания алгоритмов.
44. Какие формы занятий и в каких пропорциях целесообразно применять при изучении ОА и П?
45. Предложите несколько форм занятий при изучении ОАиП с использованием компьютеров.
46. Каким требованиям, на ваш взгляд, должны удовлетворять задачи при изучении ОАиП?
47. Ваше отношение к вспомогательным упражнениям и тестам в программировании.
48. Значение блок-схем алгоритмов и программ, их виды и типы упражнений. Когда их использовать при изучении ОАиП? Ваше отношение к блок-схемам.
49. Приведите примеры комплекса взаимосвязанных задач. Их значение, преимущества и недостатки. Когда целесообразно использовать этот методический приём?
50. Какие приведенные в электронных лекциях методические приемы, используемые при решении задач в разделе ОАиП, вам понравились, какие нет и почему? Предложите другие методические приёмы.
51. Методика преподавания начал программирования
52. Методика преподавания ветвления и циклов
53. Методика преподавания массивов
54. Методика преподавания символов и строк

55. Методика преподавания методов программирования на кружках, факультативах, в лицеях, специализированных классах и школах
56. Методика преподавания основ веб-конструирования.
57. Методика преподавания вопросов, связанных с информацией.
58. Методика преподавания аппаратного и системного программного обеспечения
59. Методика преподавания текстовых и графических редакторов.
60. Методика преподавания компьютерных презентаций.
61. Методика преподавания электронных таблиц и систем управления базами данных.
62. Методика преподавания коммуникационных технологий.
63. Методика преподавания сети Интернет.

3.3. Перечень рекомендуемых средств диагностики

Для оценки профессиональных компетенций студентов используются следующие средства диагностики:

- тестирование;
- коллоквиум;
- подготовка реферата (доклада с презентацией).

4. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

4.1. Содержание учебного материала

Введение. Понятие информатики. История преподавания школьной информатики, её современное состояние и краткое содержание.

Раздел 1. Теоретические основы методики преподавания информатики.

Тема 1.1. Дисциплина «Методика преподавания информатики» в системе педагогических знаний. Цели и задачи, место дисциплины, связи с другими дисциплинами. Требования к усвоению дисциплины. Объём дисциплины. Средства диагностики и критерии оценок. Другие особенности дисциплины.

Тема 1.2. Цели преподавания информатики. Значение предмета в учреждениях общего среднего образования. Уровни работы с компьютером. Цели преподавания информатики как единство образования, развития и воспитания. Компьютерная грамотность, образованность и информационная культура.

Тема 1.3. Принципы дидактики и преподавание информатики. Принцип научности. Сознательность усвоения и деятельности. Доступность и наглядность. Активность и самостоятельность. Индивидуализация и коллективность обучения. Эффективность учебной деятельности. Связь теории и практики.

Тема 1.4. Методы и организационные формы обучения информатике. Традиционные и словесно-фронтальные методы. Мыслительные операции и работа на ЭВМ. Индукция, дедукция и аналогия. Анализ и синтез. Организационные формы в зависимости от числа участников и от того, кто управляет обучением. Основные и вспомогательные организационные формы.

Тема 1.5. Технические и программные средства обучения информатике. Компьютерные классы, программное, техническое и информационное обеспечение. Программированное обучение. Специфика работы учителя в условиях локальной компьютерной сети, наличия электронной почты и Интернета.

Тема 1.6. Урок информатики. Методические аспекты повышения эффективности урока. Подготовка учителя к уроку. Дифференцированный и индивидуальный подходы, принципы подбора заданий. Варианты использования компьютеров на уроке. Управление и контроль за работой обучающихся.

Тема 1.7. Контроль и оценка знаний по информатике. Методы контроля знаний, их особенности. Текущий контроль, проведение и проверка самостоятельных и контрольных работ. Итоговые занятия по теме. Электронное тестирование.

Тема 1.8. Содержательная характеристика учебных программ, учебников и учебных пособий по информатике. Исторический анализ

содержания предмета «Информатика». Современное состояние, основные направления модернизации образования в области информатики. Анализ учебников и учебных пособий по информатике для учреждений общего среднего образования: исторический взгляд, действующие учебники и пособия, перспективы.

Раздел 2. Методика преподавания основ алгоритмизации и программирования.

Тема 2.1. Общие методические особенности преподавания основ алгоритмизации и программирования (ОАиП). Связь раздела с другими школьными предметами. Многоуровневость, параллельность, цикличность, сравнение и повторение. Как ввести и объяснить основные понятия, типы, способы описания алгоритмов, бытовые алгоритмы. Анализ основных языков и систем программирования. Требования к задачам по программированию. Комплекс взаимосвязанных задач, вспомогательные упражнения, тренировочные тесты, общие и индивидуальные задания, Методические приёмы, используемые при решении задач.

Тема 2.2. Методика преподавания линейных, разветвляющихся и циклических алгоритмов и программ. Линейные программы, методика их изучения. Методика преподавания присваивания, ввода, вывода, простых и составных условий, логических операций, условного оператора, операторов цикла, Сравнение ветвления и цикла в разных языках программирования.

Тема 2.3. Методика преподавания массивов, символов и строк. Как ввести и объяснить следующие понятия: массив, его объявление, индекс, размер массива, символ и строка, ввод и вывод. Методика решения задач для работы с одномерными массивами и строками. Методика использования стандартных функций и процедур для работы с символами и строками

Тема 2.4. Преподавание программирования на кружках, факультативах, в лицеях, специализированных классах и школах. Как ввести и объяснить следующие понятия: двумерный массив (матрица), процедура, функция, модуль, класс, поля и методы класса, объект. Методика преподавания *Builder* или *Delphi*: окна, работа с компонентами, событийное программирование.

Тема 2.5. Методика преподавания основ веб-конструирования. Методика преподавания методов разработки веб-сайтов: использование офисных приложений, использование языка разметки HTML, веб-конструирование в редакторе *FrontPage*, подготовка изображений для Интернета.

Раздел 3. Методика преподавания основных тем

Тема 3.1. Методика преподавания вопросов, связанных с информацией. Изучение видов и свойств информации, ее измерения и представления в памяти, носителей информации, вирусов, информационных моделей, систем и процессов.

Тема 3.2. Методика преподавания аппаратного и системного программного обеспечения. Методика изучения устройства компьютера и приемов работы на нем. Методика изучения операционной системы *Windows* и работы с файлами.

Тема 3.3. Методика преподавания текстовых и графических редакторов. Методика изучения простейших приемов работы: ввод, сохранение и загрузка, редактирование и форматирование текста. Особенности изучения технологии обработки текстовых документов. Методика обучения школьников работе в графическом редакторе *Paint* и в векторном графическом редакторе.

Тема 3.4. Методика преподавания компьютерных презентаций. Методика обучения школьников созданию и воспроизведению презентаций с использованием декоративного текста, изображений, фигур, эффектов. Особенности создания и использования презентаций на уроках по другим предметам

Тема 3.5. Методика преподавания электронных таблиц и систем управления базами данных. Методика обучения школьников простейшим приемам работы с таблицей, выполнению расчетов с помощью формул и функций, разработке диаграмм и сортировке. Методика изучения систем управления базами данных: создание таблицы базы данных, связывание таблиц, создание и заполнение формы, поиск и сортировка данных, создание отчетов.

Тема 3.6. Методика преподавания коммуникационных технологий и сети Интернет. Как ввести и объяснить основные понятия. Методика обучения работе в локальной компьютерной сети, сети Интернет и с электронной почтой. Использование образовательных и других информационных ресурсов сети Интернет при изучении школьных предметов.

4.2. Электронные ресурсы

1. Электронное приложение к учебному пособию 6 класса [Электронный ресурс]. — Режим доступа: <http://informatica6.adu.by>. — Дата доступа: 20.06.2024.
2. Электронное приложение к учебному пособию 7 класса [Электронный ресурс]. — Режим доступа: <http://informatica7.adu.by>. — Дата доступа: 20.06.2024.
3. Электронное приложение к учебному пособию 8 класса [Электронный ресурс]. — Режим доступа: <http://informatica8.adu.by>. — Дата доступа: 20.06.2024.
4. Электронное приложение к учебному пособию 10 класса (базовый уровень) [Электронный ресурс]. — Режим доступа: <http://profi.adu.by/course/view.php?id=35>. — Дата доступа: 21.06.2024..
5. Электронное приложение к учебному пособию 10 класса (повышенный уровень) [Электронный ресурс]. — Режим доступа: <http://profi.adu.by/course/view.php?id=36>. — Дата доступа: 21.06.2024.
6. Электронное приложение к учебному пособию 11 класса (базовый уровень) [Электронный ресурс]. — Режим доступа: <http://profi.adu.by/course/view.php?id=37>. — Дата доступа: 21.06.2024.

7. Электронное приложение к учебному пособию 11 класса (повышенный уровень) [Электронный ресурс]. — Режим доступа: <http://profi.adu.by/course/view.php?id=38>. — Дата доступа: 21.06.2024..
8. Материалы к учебному пособию для 6 класса [Электронный ресурс]. — Режим доступа: <http://e-vedy.adu.by/course/view.php?id=439>. — Дата доступа: 21.06.2024.
9. Материалы к учебному пособию для 7 класса [Электронный ресурс]. — Режим доступа: <http://e-vedy.adu.by/course/view.php?id=423> — Дата доступа: 22.06.2024.
10. Материалы к учебному пособию для 8 класса [Электронный ресурс]. — Режим доступа: <http://e-vedy.adu.by/course/view.php?id=64>. — Дата доступа: 22.06.2024.
11. Материалы к учебному пособию для 9 класса [Электронный ресурс]. — Режим доступа: <http://e-vedy.adu.by/course/view.php?id=63>. — Дата доступа: 22.06.2024.
12. Материалы к учебному пособию для 10 класса [Электронный ресурс]. — Режим доступа: <http://e-vedy.adu.by/course/view.php?id=62>. — Дата доступа: 24.06.2024.
13. Материалы к учебному пособию для 11 класса [Электронный ресурс]. — Режим доступа: <http://e-vedy.adu.by/course/view.php?id=61>. — Дата доступа: 24.06.2024.
14. Методика преподавания информатики. Учебная программа учреждения высшего образования по учебной дисциплине для специальности первой ступени высшего образования 1-31 03 01-02 «Математика (научно-педагогическая деятельность)» Н.А. Аленский. № УД-12152/уч 2023 г. [Электронный ресурс]. — Режим доступа: <https://elib.bsu.by/handle/123456789/304460> . — Дата доступа: 24.06.2024.

4.3. Список рекомендуемой литературы **Основная.**

1. Аленский Н.А., Травин В.В. Методика преподавания информатики. Уч.-мет. пособие с грифом УМО вузов РБ по естественно-научному образованию. — Минск, «Адукацыя і выхаванне». 2019 --- 104с.
2. Зенько, С. И. Реализация современных образовательных технологий при обучении программированию будущих учителей информатики: пособие для студентов учреждений высшего образования, обучающихся по специальности 1-02 05 01 Математика и информатика / С. И. Зенько, А. З. Кутыш ; М-во образования Республики Беларусь, Бел. гос. педагогический ун-т им. М. Танка. - 2-е изд. - Минск: БГПУ, 2020. - 318 с.
3. Математика. Физика. Информационные технологии: эвристические (открытые) задания участников оргдеятельностного семинара "Методика обучения через открытие: как обучать всех по-разному, но одинаково»: практикум / Белорусский государственный университет; [под ред. и

с предисл. А. Д. Короля; редкол.: Д. И. Губаревич и др.]. - Минск: БГУ, 2018. - 55 с.

4. Францкевич, А. А. От алгоритма к программе: методы и способы решения задач по информатике: пособие для студентов учреждений высшего образования, обучающихся по специальности 1-02 05 01 "Математика и информатика" / А. А. Францкевич; М-во образования Республики Беларусь, Бел. гос. педагогический ун-т им. М. Танка. - Минск: БГТУ, 2022. - 147 с.

Дополнительная.

15. Информатика. 6—7 классы. Дидактические и диагностические материалы. Пособие для учителей учреждений общ. среднего образования с бел. и рус. языками обучения. [С И. Зенько, Ю.А. Быкадоров, В.В. Казаченко и др.]: под ред. С.И. Зенько. — Мозырь: Выснова, 2018. — 171. [1] с.; ил. — (Компетентностный подход).

16. Информатика. 8 —9 классы. Дидактические и диагностические материалы. Пособие для учителей учреждений общ. среднего образования с бел. и рус. языками обучения. [С И. Зенько, Ю.А. Быкадоров, В.В. Казаченко и др.]: под ред. С.И. Зенько. — Мозырь: Выснова, 2018. — 191. [1] с.; ил. — (Компетентностный подход).

17. Информатика. 10 —11 классы. Дидактические и диагностические материалы. Пособие для учителей учреждений общ. среднего образования с бел. и рус. языками обучения. [С И. Зенько, И. С. Войтешенко, В.В. Казаченко и др.]: под ред. С.И. Зенько. — Мозырь: Выснова, 2020. — 231. [1] с.; цв. ил. — (Компетентностный подход).

18. Макарова, Н. П. Информатика: учеб. пособие для 6 кл. учреждений общ. сред. образования с белорус. и рус. яз. обучения/ Н. П. Макарова, А. И. Лапо, Е. Н. Войтехович. — Минск; Нар. асвета, 2018.

19. Котов В. М. Информатика: учеб. пособие для 7 кл. учреждений общ. сред. образования с белорус. и рус. яз. обучения / В. М. Котов, А. И. Лапо, Е. Н. Войтехович. — Минск; Нар. асвета, 2017.

20. Информатика: учеб. пособие для 8 кл. учреждений общ. сред. образования с белорус. и рус. яз. обучения/ В. М. Котов [и др.]. — Минск; Нар. асвета, 2018.

21. Информатика: учеб. пособие для 9 кл. учреждений общ. сред. образования с белорус. и рус. яз. обучения/ В. М. Котов [и др.]. — Минск: Нар. асвета, 2019.

22. Информатика: учеб. пособие для 10 кл. учреждений общ. сред. образования с белорус. и рус. яз. обучения/ В. М. Котов [и др.]. — Минск; Нар. асвета, 2020.

23. Информатика: учеб. пособие для 11 кл. учреждений общ. сред. образования с белорус. и рус. яз. обучения/ В. М. Котов [и др.]. — Минск; Нар. асвета, 2021.

24. Зенько С. И. Дневник педагогической практики по информатике / С. И. Зенько, С. В. Вабищевич, С. Л. Глухарева. — Минск: БГПУ, 2019. — 72 с.

25. Зенько С. И. Дневник преддипломной практики по информатике / С. И. Зенько, С. В. Вабищевич, С. Л. Глухарева. — Минск: БГПУ, 2019. — 72 с.
26. Лапо А. Е. Информатика. 6 --- 11 классы. Примерное календарно-тематическое планирование. Минск: Аверсэв, 2023. -- 63с.