

Белорусский государственный университет

УТВЕРЖДАЮ

Проректор по учебной работе и  
образовательным инновациям

 О. Прохоренко

30 июня 2023 г.

Регистрационный № УД –720 /б.

## ОСНОВЫ И МЕТОДОЛОГИИ ПРОГРАММИРОВАНИЯ

Учебная программа учреждения высшего образования  
по учебной дисциплине для специальностей:

6-05-0533-11 Прикладная информатика

6-05-0533-12 Кибербезопасность

2023 г.

Учебная программа составлена на основе ОСВО 6-05-0533-11-2023 специальности 6-05-0533-11 «Прикладная информатика», примерного учебного плана, регистрационный № 6-05-05-029/пр. от 30.01.2023, учебных планов БГУ: №6-5.3-59/03 от 15.05. 2023 г., №6-5.3-59/04 от 15.05. 2023 г., №6-5.3-59/05 от 15.05. 2023 г., №6-5.3-59/01ин. от 31.05. 2023 г., №6-5.3-59/02ин. от 31.05. 2023 г., №6-5.3-59/03ин. от 31.05. 2023 г.;

на основе ОСВО 6-05-0533-12-2023 специальности 6-05-0533-12 «Кибербезопасность», примерного учебного плана, регистрационный № 6-05-05-029/пр. от 30.01.2023, учебного плана БГУ № 6-5.3-60/02 г.

### **СОСТАВИТЕЛИ:**

**В.В. Конах** – старший преподаватель кафедры информационных систем управления Белорусского государственного университета

**Н.К. Рубашко** – старший преподаватель кафедры информационных систем управления Белорусского государственного университета

### **РЕЦЕНЗЕНТЫ:**

**С.В. Абламейко** – профессор кафедры веб-технологий и компьютерного моделирования механико-математического факультета БГУ, академик НАН Беларуси, доктор технических наук, профессор.

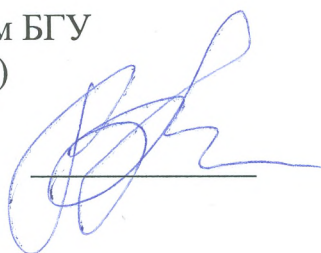
**В.А. Головки** – заведующий кафедрой интеллектуальных информационных технологий Брестского государственного технического университета, доктор технических наук, профессор.

### **РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:**

Кафедрой информационных систем управления Белорусского государственного университета  
(протокол № 18 от 08.06.2023 г.).

Научно-методическим Советом БГУ  
(протокол № 9 от 29.06.2023 г.)

Заведующий кафедрой



В.В. Краснопрошин

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная дисциплина «Основы и методологии программирования», далее «ОиМП», ориентирована на обучение студентов базовым знаниям, умениям и навыкам в области программирования. Изучаемые темы базируются на использовании современных информационных технологий, новейшего программного и технического обеспечения компьютеров.

Основой для обучения программированию является курс информатики, изучаемый в средней школе.

### **Цели и задачи учебной дисциплины**

**Цель** преподавания учебной дисциплины «ОиМП» – подготовка специалиста, умеющего проектировать эффективные алгоритмы решения поставленной задачи, выбирать наиболее подходящие структуры данных, программные и технические средства его реализации и с учетом операционного окружения разрабатывать программные приложения, отвечающие современным требованиям и новейшим компьютерным технологиям. Для этого необходимо формирование систематизированных знаний и навыков в области программирования: изучение классических методов программирования; освоение навыков разработки программ, их отладки, тестирования и документирования; формирование у студентов алгоритмического мышления, системного подхода к основам моделирования и формализации посредством создания компьютерных моделей. В итоге обеспечивается база знаний, необходимая для усвоения материала последующих учебных дисциплин в области информатики и успешной дальнейшей работы. При изложении курса важно показать возможности использования инструментария программирования при решении прикладных задач, возникающих в различных областях науки, техники, экономики и производства.

### **Задачи учебной дисциплины:**

1. Изучение основных конструкций современного языка программирования высокого уровня;
2. Овладение методологией анализа и проектирования типовых алгоритмов (линейных, условных, циклических, рекурсивных) и оценки их быстродействия;
3. Овладение методологией разработки программ на основе императивного и объектно-ориентированного подхода;
4. Приобретение навыков программирования на языке высокого уровня с использованием современных интегрированных сред разработки (IDE) и инструментальных средств;
5. Овладение основами навыков разработчика программного обеспечения (структурирование программного кода, документирование программного кода, реализация принципа модульности, получение общего представления об организации пользовательского интерфейса);

б. Формирование навыков мышления программиста и использование их при разработке ПО для решения различных профессиональных, исследовательских и прикладных задач.

**Место учебной дисциплины** в системе подготовки специалиста с высшим образованием.

Учебная дисциплина относится к **модулю** «Программирование» государственного компонента.

Программа составлена с учётом межпредметных **связей** с учебными дисциплинами.

Дисциплина «Основы и методологии программирования» непосредственно связана с параллельно изучаемыми дисциплинами:

– «Аналитическая геометрия», «Математический анализ» модуля «Высшая математика» для специальностей 6-05-0533-11 «Прикладная информатика» и 6-05-0533-12 «Кибербезопасность»;

– «Алгебра и теория чисел», «Дифференциальные уравнения» модуля «Высшая математика» для специальности 6-05-0533-11 «Прикладная информатика»;

– «Основы высшей алгебры», «Линейная алгебра» модуля «Высшая математика» для специальности 6-05-0533-12 «Кибербезопасность»;

Методы, излагаемые в указанных дисциплинах, используются для проектирования алгоритмов и программных приложений.

– Сформированные при изучении дисциплины «ОиМП» компетенции являются основой для дальнейшего изучения дисциплин: «Разработка кросс-платформенных приложений», «Машинно-ориентированное программирование», «Промышленное программирование», «Дискретная математика и математическая логика», «Алгоритмы и структуры данных», «Операционные системы», «Модели данных и СУБД», «Исследование операций» для обеих специальностей, а также дисциплины «Технологии программирования» для 6-05-0533-12 «Кибербезопасность» и «Технологии программирования для мобильных приложений» для специальности 6-05-0533-11 «Прикладная информатика».

Знания, полученные в учебной дисциплине, используются при изучении всех дисциплин специализации, при выполнении курсовых и дипломных работ, а также используются как инструментарий для моделирования и компьютерного решения задач ряда математических дисциплин, изучаемых на старших курсах.

### **Требования к компетенциям**

Освоение учебной дисциплины «Основы и методологии программирования» должно обеспечить формирование следующих **универсальных компетенций**:

*для специальности 6-05-0533-11 «Прикладная информатика»:*

**УК-5.** Быть способным к саморазвитию и совершенствованию в профессиональной деятельности.

**УК-6.** Проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности.

*для специальности 6-05-0533-12 «Кибербезопасность»:*

**УК-1.** Владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации.

**УК-2.** Решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий.

**УК-4.** Работать в команде, толерантно воспринимать социальные, этнические, конфессиональные, культурные и иные различия.

**УК-5.** Быть способным к саморазвитию и совершенствованию в профессиональной деятельности.

**УК-6.** Проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности.

Освоение учебной дисциплины «Основы и методологии программирования» должно обеспечить формирование следующей **базовой профессиональной компетенции** для обеих специальностей:

**БПК-2.** Строить, анализировать и тестировать алгоритмы и программы решения типовых задач обработки информации с использованием структурного, объектно-ориентированного и иных парадигм программирования.

В результате освоения учебной дисциплины студент должен:

**знать:**

— базовые понятия и принципы обработки информации, этапы решения автоматизируемых задач;

— принципы проектирования алгоритмов и их реализации на языке программирования;

— уровни представления данных, модели данных и методы обработки данных;

— основные методологии и средства эффективной разработки программного обеспечения;

— методы тестирования и отладки программ;

**уметь:**

— проектировать эффективные алгоритмы решения поставленной задачи;

— выбирать наиболее подходящие структуры данных, методологии разработки программ, программные и технические средства реализации алгоритма;

— разрабатывать программные приложения с заданной функциональностью и операционным окружением;

**владеть:**

— основными методами алгоритмизации практических задач;

— навыками тестирования и отладки программ;

— навыками разработки и сопровождения программ в конкретных средах разработки.

### **Структура учебной дисциплины**

Дисциплина изучается в 1-м семестре. Всего на изучение учебной дисциплины «Основы и методологии программирования» отведено:

*для специальности 6-05-0533-11 «Прикладная информатика»:*

— для очной формы получения высшего образования отведено 240 часов, в том числе 132 аудиторных часа, из них: лекции – 64 часа, лабораторные занятия – 60 часов, управляемая самостоятельная работа – 8 часов.

Трудоёмкость учебной дисциплины составляет 6 зачётных единиц.

Форма промежуточной аттестации – зачет и экзамен.

*для специальности 6-05-0533-12 «Кибербезопасность»:*

— для очной формы получения высшего образования отведено 216 часов, в том числе 132 аудиторных часа, из них: лекции – 64 часа, лабораторные занятия – 60 часов, управляемая самостоятельная работа – 8 часов.

Трудоёмкость учебной дисциплины составляет 6 зачётных единиц.

Форма промежуточной аттестации – зачет и экзамен.

## СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

### Раздел 1. Основы программирования

#### *Тема 1.1. Введение в предмет*

Алгоритм и его свойства. Способы записи алгоритмов. Основные этапы разработки и сопровождения программ. Технологии программирования. Классификация языков программирования. Способы описания языков программирования. Формы Бэкуса-Наура и синтаксические диаграммы.

#### *Тема 1.2. Основные понятия языка C++*

Структура программы. Алфавит, синтаксис, семантика языка. Признаки хорошего стиля (основы Code Style).

Типы данных, переменные, константы, выражения. Стандартные типы данных в C++. Приведение типов. Использование auto и decltype. Область действия, область видимости и время жизни переменных.

### Раздел 2. Методологии программирования

#### *Тема 2.1. Структурное программирование*

Основные управляющие структуры и операторы. Простейший ввод-вывод. Механизмы форматирования.

Алгоритмы обработки числовых данных. Вычисления с точностью.

#### *Тема 2.2. Модульное программирование*

Функции. Объявление и определение функции. Формальные и фактические параметры. Способы передачи параметров. Рекурсивные функции. Перегрузка функций. Указатели на функцию. Передача функции в качестве параметра. Callback-функции. Встроенные функции. Шаблоны функций.

Структура программы. Директивы препроцессора. Многофайловые проекты. Условная компиляция. Страж включения.

#### *Тема 2.3. Введение в тестирование*

Типы ошибок и их обработка на этапах проектирования, трансляции, выполнения. Отладка программ в среде разработки. Степпинг, точки останова. Стек вызовов и отслеживание переменных. Использование assert и static\_assert. UNIT-тестирование.

#### *Тема 2.4. Структурированные типы данных*

2.4.1. Массивы. Статические и динамические массивы. Указатели. Адресная арифметика. Одномерные и двумерные массивы. Массивы – параметры. Алгоритмы поиска и сортировки. Контейнеры vector и array. Стандартные функции сортировки и поиска.

2.4.2. Строки, операции над строками. Понятие нуль-терминированных строк. Понятие о командной строке. Передача параметров в функцию main.

2.4.3. Класс string и работа с этим классом.

2.4.4. Структуры. Массивы структур. Объединения. Перечисления. Битовые поля.

### ***Тема 2.5. Ввод-вывод***

2.5.1. Файлы. Текстовые и бинарные файлы. Основные принципы работы с файлами. Механизм чтения данных из файла. Определение конца файла. Открытие и закрытие файлов. Последовательный и прямой метод доступа.

2.5.2. Понятие потока ввода-вывода. Реализация форматированного и неформатированного ввода-вывода для потоковых классов. Строковые потоки.

### ***Тема 2.6. Обработка исключительных ситуаций***

Исключительные ситуации. Системные и пользовательские исключения. Операторы для работы с исключениями. Защищённые блоки. Выброс исключений. Стандартная обработка исключительных ситуаций. Необработанные исключения. Собственная функция аварийного завершения.

### ***Тема 2.7. Проектирование структур данных***

Линейные списки. Кольцевые, двунаправленные списки. Способы реализации списков. Стеки. Линейная и кольцевая очереди. Реализация стеков и очередей на основе массивов и списков. Сборка мусора.

## **Раздел 3. Объектно-ориентированное программирование**

### ***Тема 3.1. Основы объектно-ориентированного программирования***

Класс как абстрактный тип, классы и объекты. Члены класса, управление доступом. Реализация методов. Внутренние и внешние методы. Указатель this. Константные и статические элементы класса. Конструкторы, деструкторы. RAII. Конструктор по умолчанию. Список инициализаторов. Конструктор копирования.

### ***Тема 3.2. Дружественные функции и классы***

Объявление и определение дружественных функций. Вызовы дружественных функций. Дружественные классы. Вложенные классы.

### ***Тема 3.3. Обработка исключений***

Выброс исключений в методах класса. Перехват исключений. Использование стандартного класса exception. Метод what().

### ***Тема 3.4. Перегрузка операторов***

Способы перегрузки операторов для собственных классов. Перегрузка бинарных операторов, перегрузка оператора присваивания, особенности перегрузки унарных операторов, перегрузка операторов ввода-вывода.

***Тема 3.5. Использование классов для реализации пользовательских структур***

Реализация линейных, кольцевых, однонаправленных и двунаправленных списков, стеков и очередей в виде классов.



### ***Тема 3.6. Стандартная библиотека шаблонов***

Структура стандартной библиотеки шаблонов. Контейнерные классы. Последовательные контейнеры. Итераторы. Адаптеры контейнеров. Ассоциативные контейнеры.

### ***Тема 3.7. Элементы функционального программирования***

3.7.1. Алгоритмы STL Функциональные объекты. Отрицатели и связыватели. Примеры использования функциональных объектов при работе с алгоритмами STL. Немодифицирующие и модифицирующие алгоритмы STL.

3.7.2. Лямбда-выражения и их использование для алгоритмов STL.

**УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**Дневная форма получения образования с применением дистанционных образовательных технологий (ДОТ)**

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСП	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
	<b>Основы и методологии программирования</b>	<b>64</b>			<b>60</b>		<b>8</b>	
<b>1.</b>	<b>Основы программирования</b>	<b>4</b>			<b>4</b>			
1.1.	Введение в предмет	2			2			Собеседование
1.2.	Основные понятия языка C++	2			2			Экспресс-опрос
<b>2.</b>	<b>Методологии программирования</b>	<b>32</b>			<b>30</b>			
2.1.	Структурное программирование.	6			8			Отчет по лабораторной работе
2.2.	Модульное программирование.	4			4			Электронный тест Отчет по лабораторной работе
2.3.	Введение в тестирование	2					2	Собеседование
2.4.	Структурированные типы данных	8			8		2	Электронный тест Отчет по лабораторной работе Контрольная работа №1 по темам 2.1-2.4
2.5.	Ввод-вывод	6			4			Отчет по лабораторной работе
2.6.	Обработка исключительных ситуаций	2						Экспресс-опрос
2.7.	Проектирование структур данных	4			6		2	Дискуссия. Отчет по лабораторной работе Контрольная работа №2 по разделам 1-2
<b>3.</b>	<b>Объектно-ориентированное программирование</b>	<b>28</b>			<b>26</b>			
3.1.	Основы объектно-ориентированного программирования	6			4			Отчет по лабораторной

							работе
3.2.	Дружественные функции и классы	1			1		Экспресс-опрос
3.3.	Обработка исключений	1			1		Экспресс-опрос
3.4.	Перегрузка операторов	4			4		Расчетно- графическая работа №1
3.5.	Использование классов для реализации пользовательских структур	4			6	2	Дискуссия. Расчетно- графическая работа №2 Контрольная работа №3 по темам 3.1-3.5
3.6.	Стандартная библиотека шаблонов	6			6		Расчетно- графическая работа №3
3.7.	Элементы функционального программирования	6			4		Расчетно- графическая работа №4 Коллоквиум по разделу 3

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

### Перечень основной литературы

1. Барков, И. А. Объектно-ориентированное программирование : учебник [для вузов] / И. А. Барков. - Изд. 2-е, стер. - Санкт-Петербург : Лань, 2023. - 698 с.- URL: <https://e.lanbook.com/book/329549>.
2. Вайсфельд, М. Объектно-ориентированный подход / М. Вайсфельд. Питер, 2020. - 256 с. - URL: <https://ibooks.ru/bookshelf/371700>
3. Конова, Е. А. Алгоритмы и программы. Язык С++ : учеб. пособие для студ., обуч. по напр. "Прикладная информатика" / Е. А. Конова, Г. А. Поллак. - Изд. 7-е, стер. - Санкт-Петербург: Лань, 2023. - 384 с. — URL: <https://reader.lanbook.com/book/297002>.
4. Лафоре, Р. Объектно-ориентированное программирование в С++ / Р. Лафоре; [пер. с англ.: А. Кузнецов, М. Назаров, В. Шрага]. - 4-е изд. - Санкт-Петербург: Питер, 2022. - 923 с. — URL: <https://ibooks.ru/bookshelf/376836>.
5. Павловская, Т. А. С/С++. Программирование на языке высокого уровня / Т.А. Павловская. - СПб. [и др.]: Питер, 2020. - 464 с. - URL: <https://ibooks.ru/bookshelf/376844>.
6. Пикус Ф. Идиомы и паттерны проектирования в современном С++ /Ф. Пикус. - ДМК, 2020. - 452 с.
7. Программирование. Сборник задач: учебное пособие для вузов / О. Г. Архипов, В. С. Батасова, П. В. Гречкина [и др.]; Под редакцией М. М. Марана. - 2-е изд., стер. - Санкт-Петербург: Лань, 2022. - 140 с. - URL: <https://e.lanbook.com/book/223418>.
8. Шилдт, Г. С/С++: справочник программиста: [пер. с англ.] / Герберт Шилдт. - 3-е изд. - Москва; Санкт-Петербург: Диалектика, 2019. - 429 с.

### Перечень дополнительной литературы

1. Программирование на С++ в примерах и задачах/ А. Васильев. -Эксмо, 2018. -368 с.
2. С/С++ в задачах и примерах/ Н. Культин. - ВHV, 2019. -272 с.
3. С++ за 24 часа/ Р. Кейденхед, Либерти Джесс. -«Вильямс», 2019., 448 с.
4. Технология программирования на С++. Начальный курс/ Н. Литвиненко - ВHV, 2019.
5. Программирование: принципы и практика с использованием С++/ Страуструп, Бьярне. 2-е изд.: Пер. с англ. - М.: ООО "И.Д. Вильямс", 2016. - 1328 с.

6. Эффективный и современный C++: 42 рекомендации по использованию C++11 и C++14 / Мейерс, Скотт.: Пер. с англ. - М.: ООО "И.Д. Вильямс", 2019. - 304 с.

### Электронные ресурсы

1. Образовательный портал БГУ [Электронный ресурс]. – Режим доступа: <https://edufpmi.bsu.by/course/view.php?id=305>. – Дата доступа: 02.06.2022.
2. Образовательная платформа Insight Runner [Электронный ресурс]. – Режим доступа: <https://acm.bsu.by>. – Дата доступа: 02.06.2022.
3. Страуструп Б. - Язык программирования C++ для профессионалов - Национальный Открытый Университет "ИНТУИТ" - 2016 - 670с. - ISBN: - Текст электронный // ЭБС ЛАНЬ - URL: <https://e.lanbook.com/book/100542>.
4. Объектно-ориентированное программирование: учебник / И. А. Барков. - Санкт-Петербург: Лань, 2022. - 700 с. - Текст: электронный // Лань: электронно-библиотечная система. - URL: <https://e.lanbook.com/book/206699>.

### Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Для диагностики компетенций в рамках учебной дисциплины рекомендуется использовать следующие формы:

1. Устная форма: собеседование, экспресс-опрос, дискуссия.
2. Письменная форма: контрольные работы, коллоквиум.
3. Устно-письменная форма: отчеты по лабораторным и расчетно-графическим работам с их устной защитой.
4. Техническая форма: электронные тесты.

В качестве рекомендуемых технических средств диагностики используется Образовательная платформа Insight Runner ([www.acm.bsu.by](http://www.acm.bsu.by)), а также обучение, организованное на платформе Moodle (<https://edufpmi.bsu.by>).

Формой текущей аттестации по дисциплине учебным планом предусмотрен **зачет и экзамен**.

При формировании итоговой отметки используется рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов для текущего контроля знаний и текущей аттестации студентов по дисциплине.

Примерные весовые коэффициенты, определяющие вклад текущего контроля знаний и текущей аттестации в итоговую отметку:

- контрольные работы – 50 %;

- отчеты по лабораторным и расчетно-графическим работам с их устной защитой – 30 %;
- выполнение тестов – 10%;
- коллоквиум – 10 %.

Итоговая отметка по дисциплине рассчитывается на основе отметки текущей успеваемости и экзаменационной отметки с учетом их весовых коэффициентов. Вес отметки по текущей успеваемости составляет 40 %, экзаменационной отметки – 60 %.

Точки контроля по текущей успеваемости формируются из расчета общего количества часов (зачетных единиц), выделенных на изучение дисциплины.

### **Примерный перечень заданий для управляемой самостоятельной работы студентов**

В качестве заданий для управляемой самостоятельной работы могут быть выданы задания для самостоятельного решения задач по следующим темам:

1. **Тема 2.3.** «Введение в тестирование». Типы ошибок и их обработка на этапах проектирования, трансляции, выполнения. Отладка программ в среде разработки. Степпинг, точки останова. Стек вызовов и отслеживание переменных. Использование assert и static\_assert. UNIT-тестирование. (2 ч.)

**Форма контроля** – собеседование.

2. **Тема 2.4.** «Структурированные типы данных». Задачи преобразования одномерных массивов, матриц и строк, задачи поиска и сортировки, задачи обработки динамических массивов (2 ч.)

**Форма контроля** – электронный тест: программная реализация алгоритмов решения задач и проверка их работоспособности на Образовательной платформе Insight Runner.

3. **Тема 2.7.** «Проектирование структур данных». Линейные, кольцевые, двунаправленные списки и способы их реализации. Стеки, линейная и кольцевая очереди и их реализация. (2 ч.)

**Форма контроля** – дискуссия.

4. **Тема 3.5.** «Использование классов для реализации пользовательских структур». Реализация линейных, кольцевых, однонаправленных и двунаправленных списков, стеков и очередей в виде классов. (2 ч.)

**Форма контроля** – дискуссия.

### **Примерная тематика лабораторных занятий**

**Занятие 1-2.** Описание языков программирования: нормальная форма Бекуса, синтаксические диаграммы. Алгоритм и его свойства. Способы

записи алгоритмов. Основные этапы разработки и сопровождения программ. Алфавит, синтаксис, семантика языка программирования.

**Занятие 2-4.** Основные управляющие структуры и операторы. Простейший ввод-вывод. Инструкции выбора, циклов, переходов.

**Занятие 5-6.** Алгоритмы обработки числовых данных. Вычисления с заданной точностью.

**Занятие 7-8.** Модульное программирование. Функции. Объявление и определение функции. Формальные и фактические параметры. Передача параметров по значению, по указателю и ссылке.

**Занятие 9-10.** Структурированные типы данных. Статические и динамические одномерные массивы, vector, array. Алгоритмы сортировки и поиска.

**Занятие 11-12.** Двумерные и многомерные массивы. Обработка матриц.

**Занятие 13-14.** Строки. Класс string и работа с этим классом. Функции стандартной библиотеки для обработки строк.

**Занятие 15-16.** Обработка структурированных данных. Организация ввода/вывода с использованием потоковых классов. Текстовые файлы.

**Занятие 17-18.** Исключительные ситуации. Системные и пользовательские исключения. Оператор try ... catch. Виды блоков catch. Выброс исключений. Оператор throw. Стандартная обработка исключительных ситуаций.

**Занятие 19-20.** Основы объектно-ориентированного программирования. Класс как абстрактный тип. Разработка классов обработки чисел.

**Занятие 21-22.** Дружественные функции и классы. Перегрузка операторов.

**Занятие 23-24.** Реализация пользовательских классов.

**Занятие 25-26.** Классы списки, стеки и очереди.

**Занятие 27-28.** Стандартная библиотека шаблонов STL Использование последовательных и ассоциативных контейнеров.

**Занятие 29-30.** Алгоритмы STL Функциональные объекты. Лямбда-выражения и их использование для алгоритмов STL.

*Рекомендуемая тематика контрольных работ и коллоквиума:*

- 1) Контрольная работа № 1 «Структурированные типы данных».
- 2) Контрольная работа № 2 «Обработка файлов».
- 3) Контрольная работа № 3. «Основы объектно-ориентированного программирования».
- 4) Коллоквиум «Объектно-ориентированное программирование и STL».

### **Описание инновационных подходов и методов к преподаванию учебной дисциплины**

При организации образовательного процесса используются следующие методы:

- **метод учебной дискуссии**, который предполагает участие студентов в целенаправленном обмене мнениями, идеями для предъявления и/или согласования существующих позиций по определенной проблеме. Использование метода обеспечивает появление нового уровня понимания изучаемой темы, применение знаний (теорий, концепций) при решении проблем, определение способов их решения.

- **метод группового обучения**, который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями.

В качестве технических средств для организации работы в рамках учебной дисциплины рекомендуется использовать Образовательную платформу Insight Runner ([www.acm.bsu.by](http://www.acm.bsu.by)), Образовательный портал БГУ (<https://edufpmi.bsu.by>) – инструмент с эффективной функциональностью контроля, тренинга и самостоятельной работы. Для контроля самостоятельности выполнения работ рекомендуется использовать автоматические системы определения несанкционированных материалов, функционирующие в Insight Runner.

- **практико-ориентированный подход**, который предполагает освоение содержания образования через решения практических задач; приобретение навыков эффективного выполнения разных видов профессиональной деятельности; ориентацию на генерирование идей, реализацию групповых студенческих проектов; использование процедур, способов оценивания, фиксирующих профессиональные компетенции.

### **Методические рекомендации по организации самостоятельной работы обучающихся**

Для организации самостоятельной работы студентов по учебной дисциплине следует использовать современные информационные ресурсы: разместить на образовательном портале комплекс учебных и учебно-методических материалов (учебно-программные материалы, учебное издание для теоретического изучения дисциплины, методические указания к лабораторным занятиям, материалы текущего контроля и текущей аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образовательных стандартов высшего образования и учебно-программной документации, в т.ч. вопросы для подготовки к зачету и экзамену, задания, тесты, вопросы для самоконтроля и др., список рекомендуемой литературы, информационных ресурсов и др.).

При составлении **заданий УСР** по учебной дисциплине необходимо предусмотреть возрастание их сложности: от заданий, формирующих достаточные знания по изученному учебному материалу на уровне узнавания, к заданиям, формирующим компетенции на уровне



воспроизведения, и далее к заданиям, формирующим компетенции на уровне применения полученных знаний.

Таким образом, задания УСР по учебной дисциплине рекомендуется делить на три модуля:

- задания, формирующие достаточные знания по изученному учебному материалу на уровне узнавания;
- задания, формирующие компетенции на уровне воспроизведения;
- задания, формирующие компетенции на уровне применения полученных знаний.

### *Примерный перечень заданий*

#### **Задания на уровне узнавания**

Задание 1. Необходимо найти требуемый элемент в одномерном массиве. Какие из изученных алгоритмов можно применить, если известно, что массив а) упорядочен; б) неупорядочен?

Задание 2. Перечислите известные вам алгоритмы внутренней сортировки. Какие из этих алгоритмов реализованы в стандартных функциях языка C++? Как применить стандартные функции сортировки для массива, содержащего объекты нестандартного типа?

Задание 3. Перечислите известные вам функции обработки строк? Какие из них понадобятся, чтобы в строке поменять местами слово максимальной и минимальной длины? Убрать повторяющиеся символы? Найти все слова, являющиеся целыми числами?

#### **Задания на уровне воспроизведения**

Задание 1. Построить линейный однонаправленный список, упорядоченный по возрастанию ключей. Имя входного файла: *input.txt*. Имя выходного файла: *output.txt*.

*Формат входных данных.* Входной файл содержит последовательность чисел – ключи в порядке добавления в список. Ключи задаются в формате по одному в строке.

*Формат выходных данных.* Выходной файл должен содержать последовательность ключей, полученную при проходе по списку в прямом порядке.

Задание 2. Из кольцевого однонаправленного списка удалить все элементы с ключами  $k$ , удовлетворяющими условию  $k < x$ . Имя входного файла: *input.txt*. Имя выходного файла: *output.txt*.

*Формат входных данных.* Входной файл содержит число  $x$ .

*Формат выходных данных.* Выходной файл должен содержать последовательность ключей, полученную при проходе по списку в прямом порядке после выполненного удаления.

*Примечание.* Предварительно нужно построить исходный список.

Задание 3. Объединить два линейных однонаправленных списка, упорядоченных по возрастанию ключей (ключи в списке не повторяются). При объединении списков одинаковые ключи включаются в итоговый список один раз. Имя входного файла: *input.txt*. Имя выходного файла: *output.txt*.

*Формат входных данных.* Входной файл содержит две строки, в каждой – последовательность чисел – ключи в порядке добавления их в первый и второй список соответственно. Числа разделены пробелами.

*Формат выходных данных.* Выходной файл должен содержать последовательность ключей, полученную при проходе по итоговому списку в прямом порядке.

### **Задания на уровне применения полученных знаний**

Задание 1. Имеется план местности, разбитой на квадраты, заданный матрицей размером  $N \times N$ . Каждый квадрат имеет высоту относительно уровня моря, значение которой определяется натуральным числом. Необходимо определить маршрут каравана из стартовой позиции  $(X_s, Y_s)$  в финишную позицию  $(X_f, Y_f)$ . Караван может двигаться только по местности параллельно осям  $Ox$  и  $Oy$  между центрами квадратов и только в соседний квадрат с меньшей высотой.

Задание 2. Из листа клетчатой бумаги размером  $M \times N$  клеток удалили некоторые клетки. На какое количество кусков распадется оставшаяся часть листа, если перед удалением клеток лист склеили в цилиндр высотой  $N$ ?

Пример. Если из шахматной доски удалить все клетки одного цвета, то оставшаяся часть распадется на 32 куска.

Задание 3. Форма тела задана матрицей  $A$  размером  $M \times N$ . Элементы матрицы – натуральные числа. Элемент  $A[i, j]$  соответствует высоте горизонтальной квадратной площадке размером  $1 \times 1$  относительно нижнего основания. Нижнее основание формы горизонтально.

Определить максимально возможный объем невытекшей воды, если тело опускается полностью в воду, затем поднимается.

Пример:

$$A = \begin{pmatrix} 5 & 3 & 1 \\ 5 & 2 & 5 \\ 2 & 5 & 5 \end{pmatrix}$$

После подъема вода останется только во внутреннем углублении, над элементом  $A[2, 2]=1$ . Объем невытекшей воды равен 1.

### **Примерный перечень вопросов к зачету**

1. Определение и свойства алгоритма. Способы записи алгоритмов.

2. Структура программы на языке C++. Типы данных, переменные, константы. Стандартные типы данных в C++. Приведение типов.
3. Унарные и бинарные арифметические операторы в C++. Операторы сравнения. Операторы сдвига. Побитовые и логические операторы C++. Неполное вычисление логических выражений.
4. Операторы присваивания, инкремента и декремента. Условный оператор. Оператор запятая. Инструкция-выражение. Инструкции выбора if и switch. Инструкции передачи управления.
5. Организация циклов в C++.
6. Объявление и определение функции в C++. Формальные и фактические параметры. Способы передачи параметров (по значению и по адресу).
7. Встроенные функции. Стандартные функции.
8. Рекурсия. Формы рекурсивных подпрограмм.
9. Перегрузка функций.
10. Указатели на функции.
11. Шаблоны функций.
12. Статические одномерные и двумерные массивы.
13. Введение в STL, контейнеры vector и array.
14. Динамическое выделение памяти под одномерные и двумерные массивы.
15. Передача одномерных и двумерных массивов в качестве параметров.
16. Алгоритмы поиска и сортировки.
17. Стандартные функции сортировки и поиска.
18. Строки. Операции над строками. Понятие нуль-терминированных строк.
19. Класс string. Функции стандартной библиотеки для обработки строк.
20. Тип struct. Массивы структур.
21. Файлы. Механизм чтения данных из файла. Определение конца файла. Открытие и закрытие файлов. Последовательный и прямой метод доступа.
22. Понятие потока ввода-вывода. Операции над потоками.
23. Реализация форматированного ввода-вывода для потоковых классов. Флаги форматирования. Форматирующие методы. Манипуляторы потоков ввода-вывода и их использование.
24. Текстовые файлы. Форматированный ввод-вывод.
25. Бинарные файлы. Последовательный и прямой методы доступа.
26. Исключительные ситуации.
27. Списки. Линейные и кольцевые списки. Представление этих структур в статической и динамической памяти. Обработка однонаправленных и двунаправленных списков. Сборка мусора.
28. Структура данных очередь. Линейная очередь. Кольцевая очередь. Реализация очереди с использованием массива и списка.
29. Структура данных стек. Реализация стека с использованием массива и списка.

30. Описание класса в C++. Поля и методы. Внутренние и внешние методы. Разграничение доступа к элементам класса. Спецификаторы доступа, set- и get- методы. Константные методы.
31. Конструкторы. Списки инициализаторов в конструкторах. Конструктор по умолчанию. Конструктор копирования.
32. Деструкторы. Деструктор по умолчанию. Необходимость написания собственных деструкторов.
33. Дружественные функции и классы.
34. Выброс исключений в методах класса. Выбросы исключений в конструкторах. Использование стандартного класса exception для выброса исключений.
35. Перегрузка операторов. Способы перегрузки операторов для собственных классов.
36. STL. Контейнерные классы. Итераторы. Типы итераторов. Операции над итераторами. Интервалы.
37. STL. Последовательные контейнеры. Операции над последовательными контейнерами.
38. STL. Адаптеры контейнеров.
39. STL. Ассоциативные контейнеры. Операции над ассоциативными контейнерами.
40. STL. Функциональные объекты. Отрицатели и связыватели.
41. STL. Немодифицирующие и модифицирующие алгоритмы STL (вкл. алгоритмы сортировки).
42. Лямбда-выражения.

### **Примерный перечень вопросов к экзамену**

1. Определение и свойства алгоритма. Способы записи алгоритмов.
2. Основные этапы разработки и сопровождения программ.
3. Технологии программирования. Структурное программирование.
4. Классификация языков программирования. Способы описания языков программирования. Формы Бэкуса-Наура и синтаксические диаграммы.
5. Лексемы, выражения, операторы. Структура программы на языке C++. Типы данных, переменные, константы. Стандартные типы данных в C++. Приведение типов.
6. Переменные в C++. Область действия, область видимости и время жизни переменных.
7. Унарные и бинарные арифметические операторы в C++. Операторы сравнения. Операторы сдвига. Побитовые и логические операторы C++. Неполное вычисление логических выражений.
8. Операторы присваивания, инкремента и декремента. L-value и r-value выражения.

9. Условный оператор. Оператор запятая. Инструкция-выражение. Инструкции выбора if и switch. Инструкции передачи управления.
10. Организация циклов в C++. Цикл foreach. Ключевое слово auto.
11. Подпрограммы. Определение и объявление подпрограмм. Процедуры и функции.
12. Объявление и определение функции в C++. Формальные и фактические параметры. Соответствие типов в формальных и фактических параметрах. Способы передачи параметров (по значению и по адресу). Параметры по умолчанию. Встроенные функции. Стандартные функции. Побочный эффект подпрограмм.
13. Рекурсия. Формы рекурсивных подпрограмм. Глубина и текущий уровень рекурсии. Заикливание рекурсивных подпрограмм. Примеры неэффективности рекурсии.
14. Перегрузка функций. Ошибки, возникающие при перегрузке функций.
15. Указатели на функции. Callback-функции.
16. Макросы. Преимущества и недостатки использования макросов.
17. Шаблоны функций.
18. Директивы условной компиляции. Многофайловые проекты. Страж включения.
19. Массивы. Статические массивы.
20. Введение в STL, контейнеры vector и array.
21. Указатели. Типизированные и безтиповые указатели. Операция разыменования и операция получения адреса. Арифметические операции над указателями. Проблемы и типичные ошибки при работе с указателями. Константные указатели и указатели на константу.
22. Массивы. Реализация одномерных массивов с помощью указателей.
23. Многомерные массивы. Реализация многомерных массивов с помощью указателей. Динамическое выделение памяти под одномерные и двумерные массивы.
24. Указатели-параметры. Массивы указателей.
25. Передача одномерных массивов в качестве параметров. Передача двумерных массивов в качестве параметров.
26. Алгоритмы поиска и сортировки. Стандартные функции сортировки и поиска.
27. Строки. Операции над строками. Понятие нуль-терминированных строк.
28. Функция main. Передача параметров в функцию main.
29. Класс string. Функции стандартной библиотеки для обработки строк.
30. Пользовательские типы данных. Перечислимый тип enum. Тип struct. Массивы структур.
31. Типы ошибок и их обработка на этапах проектирования, трансляции, выполнения. Отладка программ в среде разработки. Степпинг, точки останова. Стек вызовов и отслеживание переменных.
32. Использование assert и static\_assert.
33. UNIT-тестирование.

34. Файлы. Основные принципы работы с файлами. Механизм чтения данных из файла. Определение конца файла. Открытие и закрытие файлов. Последовательный и прямой метод доступа.
35. Понятие потока ввода-вывода. Операции над потоками. Классификация потоков ввода-вывода. Иерархия потоковых классов.
36. Реализация форматированного ввода-вывода для потоковых классов. Флаги форматирования. Форматирующие методы. Манипуляторы потоков ввода-вывода и их использование.
37. Реализация неформатированного ввода-вывода для потоковых классов.
38. Текстовые файлы. Форматированный ввод-вывод.
39. Бинарные файлы. Последовательный метод доступа.
40. Организация прямого доступа к файлу.
41. Исключительные ситуации. Классификация исключительных ситуаций. Операторы для работы с исключениями. Защищённые блоки. Выброс системных и пользовательских исключений. Последовательность действий при обработке исключений. Необработанные исключения. Собственная функция аварийного завершения.
42. Структуры данных. Списки. Типы списков. Представление этих структур в статической и динамической памяти. Обработка однонаправленных и двунаправленных списков. Сборка мусора.
43. Списки. Структура линейного однонаправленного списка. Реализация списка с использованием динамической памяти и с использованием массивов.
44. Структура данных очередь. Линейная очередь. Кольцевая очередь. Реализация очереди с использованием массива и списка.
45. Структура данных стек. Реализация стека с использованием массива и списка.
46. Описание класса в C++. Поля и методы. Внутренние и внешние методы. Разграничение доступа к элементам класса. Спецификаторы доступа, set- и get- методы. Константные методы. Необходимость использования константных методов.
47. Конструкторы. Списки инициализаторов в конструкторах. Конструктор по умолчанию. Конструктор копирования.
48. Деструкторы. Деструктор по умолчанию. Необходимость написания собственных деструкторов. Деструкторы и RAII.
49. Дружественные функции и классы.
50. Указатель this. Константные и статические поля.
51. Выброс исключений в методах класса. Выбросы исключений в конструкторах. Использование стандартного класса exception для выброса исключений.
52. Перегрузка операторов. Способы перегрузки операторов для собственных классов.
53. STL. Контейнерные классы. Итераторы. Типы итераторов. Операции над итераторами. Интервалы.

- 54. STL. Последовательные контейнеры. Операции над последовательными контейнерами. Адаптеры контейнеров.
- 55. STL. Ассоциативные контейнеры. Операции над ассоциативными контейнерами.
- 56. STL. Функциональные объекты. Отрицатели и связыватели.
- 57. STL. Немодифицирующие и модифицирующие алгоритмы STL (вкл. алгоритмы сортировки).
- 58. Лямбда-выражения.

## ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
1. Разработка кросс-платформенных приложений	Информационных систем управления	Нет	Изменений не требуется (протокол № 18 от 08.06.2023 г.).
2. Операционные системы	Информационных систем управления	Нет	Изменений не требуется (протокол № 18 от 08.06.2023 г.).
3. Промышленное программирование	Информационных систем управления	Нет	Изменений не требуется (протокол № 18 от 08.06.2023 г.).
4. Технологии программирования	Информационных систем управления	Нет	Изменений не требуется (протокол № 18 от 08.06.2023 г.).



**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ  
ПО ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**  
на \_\_\_\_ / \_\_\_\_ учебный год

№№ ПП	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры информационных систем управления (протокол № \_\_\_\_ от \_\_\_\_\_ 202\_ г.)

Заведующий кафедрой

\_\_\_\_\_

(степень, звание)

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(И.О.Фамилия)

УТВЕРЖДАЮ  
Декан факультета

\_\_\_\_\_

(степень, звание)

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(И.О.Фамилия)