

Белорусский государственный университет

УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

О.Г. Прохоренко

«05» июля 2023 г.

Регистрационный № УД – 803/б.

ОСНОВЫ И МЕТОДОЛОГИИ ПРОГРАММИРОВАНИЯ

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальности:**

6-05-0533-09 Прикладная математика

2023 г.

Учебная программа составлена на основе ОСВО 6-05-0533-09-2023 специальности 6-05-0533-09 «Прикладная математика», примерного учебного плана, регистрационный № 6-05-05-020/пр. от 20.12.2022, учебных планов БГУ: № 6-5.3-57/01 от 15.05 2023 г., № 6-5.3-57/02 от 15.05 2023 г., № 6-5.3-57/03 от 15.05 2023 г., № 6-5.3-57/04 от 15.05 2023 г.

СОСТАВИТЕЛИ:

Карпович Н.А., старший преподаватель кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета;

Зенько Т.А., старший преподаватель кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета.

Пазюра Е.В., старший преподаватель кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета.

РЕЦЕНЗЕНТ:

П. В. Гляков, профессор кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств», кандидат физико-математических наук, доцент.

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

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой технологий программирования БГУ
(протокол № 16 от 18.05.2023)

Научно-методическим советом БГУ
(протокол № 9 от 29.06.2023)

Заведующий кафедрой _____ А.Н. Курбацкий

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

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 предмет «Информатика», изучаемый в средней школе.

Цели и задачи учебной дисциплины

Цель учебной дисциплины «Основы и методологии программирования» – дать студентам базу, необходимую для усвоения материала последующих учебных дисциплин в области информатики, и сформировать составную часть банка знаний, необходимого студентам для 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. Программирование является основополагающей дисциплиной для освоения технологии программирования, курсов по разработке Интернет-приложений, а также для дальнейшей самостоятельной работы с различными инструментами разработки программных продуктов.

Задачи учебной дисциплины: подготовка специалиста, умеющего проектировать эффективные алгоритмы решения поставленной задачи, выбирать наиболее подходящие структуры данных, использовать современные методы программирования, программные и технические средства его реализации и с учетом операционного окружения разрабатывать программные приложения, отвечающие современным требованиям и новейшим компьютерным технологиям.

При написании программы по учебной дисциплине «ОиМП» использовались современные методологии разработки программ, в частности: структурное программирование, модульное программирование, объектно-ориентированное программирование.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина относится к модулю «Программирование» государственного компонента.

Связи с другими учебными дисциплинами, включая учебные дисциплины компонента учреждения высшего образования, дисциплины специализации и другие.

Учебная дисциплина «Основы и методологии программирования» является основой для дальнейшего изучения дисциплин модуля «Программирование»: «Разработка кросс-платформенных приложений», «Машинно-ориентированное программирование», «Промышленное программирование», «Технологии программирования», которые являются базовыми при изучении всех дисциплин специальности, при выполнении курсовых и дипломных работ, а также используются как инструментарий для моделирования и компьютерного решения задач математических дисциплин.

Требования к компетенциям

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

Универсальные компетенции:

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

Базовые профессиональные компетенции:

БПК-4. Применять навыки построения, анализа и тестирования алгоритмов и программ для решения типовых задач прикладной математики

БПК-5. Применять при проектировании приложений такие парадигмы программирования, как структурное, объектно-ориентированное и функциональное программирование, а также иные парадигмы, разрабатывать программное обеспечение в интегрированных средах разработки.

В результате освоения учебной дисциплины студент должен:

знать:

- современный процедурно-ориентированный алгоритмический язык программирования;
- принципы проектирования алгоритмов и их реализации;
- основные теоретические понятия объектно-ориентированного программирования и механизмы реализации объектно-ориентированного подхода;
- основные методологии и средства эффективной разработки программного обеспечения;
- методы тестирования и отладки программ;
- основные понятия и принципы обработки информации, основы компьютерной обработки информации;

уметь:

- проектировать эффективные алгоритмы решения поставленной задачи;
- выбирать наиболее подходящие структуры данных, методологии разработки программ, выбирать 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 программы;
- навыками разработки и сопровождения программ в конкретных средах разработки.

Структура учебной дисциплины

Дисциплина изучается в первом семестре. Всего на изучение учебной дисциплины «Основы и методологии программирования» 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.

Трудоемкость учебной дисциплины составляет 6 зачетных единиц.

Форма промежуточной аттестации – зачет и экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Методология структурного и процедурного программирования

Тема 1.1. Основы программирования

Введение в программирование

Методологии программирования. История и предназначение C++. Структура программ, заголовочные файлы, подключение библиотек. Комментарии. Переменные и их именование. Пространство имен std. Простой ввод-вывод.

Тема 1.2. Структура языка программирования

Элементы языка. Символы. Ключевые слова. Идентификаторы. Константы. Комментарии. Фундаментальные типы данных. Модификаторы и спецификаторы типов. Переменные. Квалификаторы типов. Примеры простых программ.

Тема 1.3. Структурное программирование

Основные управляющие структуры и операторы. Выражения. Приведение типов, преобразование типов, присваивание. Инструкции выбора, циклов, переходов.

Алгоритмы обработки числовых данных.

Тема 1.4. Модульное программирование

Функции. Объявление и определение функции. Формальные и фактические параметры. Способы передачи параметров. Вызов функции. Область действия имен. Спецификаторы классов памяти.

Перегрузка функций. Шаблоны функций. Функции с переменным числом параметров. Стандартные функции. Функция main.

Тема 1.5. Структурированные типы данных

1.5.1 Массивы. Обработка одномерных и двумерных массивов, понятие vector, array.

1.5.2 Указатели. Арифметические действия с указателями. Передача параметров массивов.

1.5.3 Рекуррентные соотношения и рекурсивные функции.

1.5.4 Алгоритмы сортировки и поиска. Стандартные функции сортировки и поиска.

1.5.5 Функции стандартной библиотеки для обработки строк. Определение и инициализация, функции преобразования строк и чисел.

1.5.6 Функции класса string для обработки строк. Определение и инициализация строк. Функции преобразования строк и чисел.

1.5.7 Структуры. Объединения. Перечисления.

1.5.8 Файлы и потоки. Ввод-вывод. Текстовые и бинарные потоки.

1.5.9 Программирование алгоритмов на основе структурного и модульного подходов.

Тема 1.6. Тестирование программ, Unit-тестирование

Синтаксические, семантические, логические ошибки. UNIT-тестирование.

Раздел 2. Методология объектно-ориентированного программирования

Тема 2.1. Введение в объектно-ориентированное программирование

Класс как абстрактный тип, объекты класса. Конструкторы, деструкторы, методы классов. Константные компоненты и методы класса. Статические компоненты и методы класса.

Классы для обработки числовых объектов.

Тема 2.2. Обработка регулярных выражений

2.2.1 Синтаксис регулярных выражений. Создание шаблонов регулярных выражений.

2.2.2. Основные функции при обработке текста с использованием регулярных выражений.

Тема 2.3. Шаблоны классов

Определение шаблона класса. Специализация шаблона класса. Статические члены шаблона класса. Шаблоны как параметры шаблонов функций и шаблонов класса.

Тема 2.4. Дружественные классы и функции

Объявление и определение дружественных функций. Вызовы дружественных функций. Дружественные классы. Объявление и определение. Конструктор `explicit`.

Тема 2.5. Перегрузка операторов

Общие правила. Унарные операторы. Оператор присваивания. Оператор индексирования. Оператор доступа к членам класса. Операторы инкремента и декремента. Операторы `new` и `delete`.

Бинарные операторы. Перегрузка арифметических операторов. Операторы преобразования типов (конвертеры). Перегрузка операторов `>>` и `<<` для ввода и вывода. Оператор вызова функции.

Тема 2.6. Обработка исключений

Механизм обработки исключений. Вложенные исключения. Классы для обработки исключений. Реализация исключений. Спецификация исключений. Обработка классов с использованием исключений.

Тема 2.7. Использование стандартной библиотеки шаблонных классов STL

2.7.1. Характеристика стандартной библиотеки языка программирования. Структура стандартной библиотеки шаблонов STL. Использование стандартной библиотеки шаблонных классов STL. Итераторы.

`deque`.

`priority_queue`.

2.7.4. Ассоциативные контейнеры.

Тема 2.8. Элементы функционального программирования

2.8.1. Алгоритмы STL. Немодифицирующие алгоритмы: алгоритмы обработки множества, алгоритмы поиска минимальных и максимальных.

2.8.2. Модифицирующие алгоритмы STL: алгоритмы удаления, перестановочные алгоритмы.

2.8.3. Алгоритмы сортировки. Алгоритмы упорядоченных интервалов. Обобщенные численные алгоритмы.

2.8.4. Функциональные объекты. Отрицатели и связыватели. Функторы Лямбда-выражения и их использование для алгоритмов STL.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная форма получения высшего образования с применением
дистанционных образовательных технологий (ДОТ)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
I	Методология структурного и процедурного программирования	34			32		2	
1.1.	<i>Основы программирования</i>	2						
1.2.	<i>Структура языка программирования</i>	4						Письменный опрос на лекции
1.3.	<i>Структурное программирование</i>	4			8			Отчет по лабораторным работам
1.4.	<i>Модульное программирование</i>	4			4			Отчет по лабораторным работам
1.5.	<i>Структурированные типы данных</i>	18			20			Письменный опрос на лекциях
1.5.1	Массивы. Обработка одномерных и двумерных массивов, понятие vector, array.	2			4			Отчет по лабораторным работам
1.5.2	Указатели. Арифметические действия с указателями. Передача параметров массивов	2			4			Отчет по лабораторным работам
1.5.3	Рекуррентные соотношения и рекурсивные функции.	2			2			Отчет по лабораторным работам
1.5.4	Алгоритмы сортировки и поиска. Стандартные функции сортировки и поиска.	2			2			Отчет по лабораторным работам
1.5.5	Функции стандартной библиотеки для обработки строк. Определение и инициализация строк. Функции преобразования строк и чисел.	2			2			Отчет по лабораторным работам
1.5.6	Функции класса string для обработки строк. Определение и инициа-	2			2			Отчет по лабораторным работам

	лизация строк. Функции преобразования строк и чисел.						
1.5.7	Структуры. Объединения. Перечисления.	2		2			Контрольная работа по структурному программированию (раздел 1)
1.5.8	Файлы и потоки. Ввод-вывод данных. Разработка многофайловых проектов.	2		2			Отчет по лабораторным работам
1.5.9	Программирование алгоритмов на основе структурного и модульного подходов.	2					Коллоквиум Тест в системе EDU
1.6.	Тестирование программ. Unit-тестирование.	2				2	Устный опрос
II.	Методология объектно-ориентированного программирования	30		28		6	Письменный опрос на лекциях
2.1.	Введение в объектно-ориентированное программирование	4		4			Отчет по лабораторным работам опрос
2.2.	Обработка регулярных выражений	2		2			Контрольная работа по ООП (раздел 2)
2.3.	Шаблоны классов	2		4			Отчет по лабораторным работам
2.4.	Дружественные классы и функции	2		2			Отчет по лабораторным работам
2.5.	Перегрузка операторов	4		4		2	Отчет по лабораторным работам
2.6.	Обработка исключений	2		2			Контрольная работа по ООП (раздел 2)
2.7.	Использование стандартной библиотеки шаблонных классов STL	8		6		2	Дискуссия, отчет по лабораторным работам
2.8.	Элементы функционального программирования	6		4		2	Дискуссия, отчет по лабораторным работам, Контрольная работа STL (раздел 2)
	Всего	64		60		8	

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. Вайсфельд М. Объектно-ориентированный подход / М. Вайсфельд. Питер, 2020. - 256 с. - URL: <https://ibooks.ru/bookshelf/371700>.
2. 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>.
3. Шилдт, Г.С/С++ : справочник программиста : [пер. с англ.] / Герберт Шилдт. - 3-е изд. - Москва ; Санкт-Петербург : Диалектика, 2019. - 429 с.
4. Пикус Ф. Идиомы и паттерны проектирования в современном С++ /Ф. Пикус. - ДМК Пресс, 2020. - 452 с.
5. 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>.
6. 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>.
7. 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. - Москва: Лань, 2023. - 384 с. - URL: <https://reader.lanbook.com/book/297002>.
8. Солдатенко, И. С. Практическое введение в язык программирования Си : учебное пособие / И. С. Солдатенко, И. В. Попов. - Санкт-Петербург ; Москва ; Краснодар : Лань, 2022. - 132 с. - URL: <https://e.lanbook.com/book/213149>.

Перечень дополнительной литературы

1. Страуструп Б. - Язык программирования С++ для профессионалов - Национальный Открытый Университет "ИНТУИТ" - 2016 - 670с. - ISBN: - Текст электронный // ЭБС ЛАНЬ - URL: <https://e.lanbook.com/book/100542>.
2. Васильев А. Программирование на С++ в примерах и задачах / А. Васильев. -Эксмо, 2018. -368 с.
3. Культин Н. С/С++ в задачах и примерах / Н.Культин. - BHV, 2019. - 272 с.
4. Мейерс С. Эффективный и современный С++. 42 рекомендации по использованию С++11 и С++14 / Скотт Мейерс, - Вильямс, 2019. - 304 с.

5. Кейденхед Р. С++ за 24 часа / Р. Кейденхед, Либерти Джесс. - «Вильямс», 2019., 448 с.
6. Николай Джосаттис. “С++ 2017: полное руководство” / Nicolai M. Josuttis. “С++ 2017: The Complete Guide”
7. Литвиненко Н. Технология программирования на С++. Начальный курс / Н. Литвиненко - BHV, 2019.
8. Язык программирования С = Programming Language C / Брайан Керниган, Денис Ритчи; [пер. с англ. и ред. В. Л. Бродового]. - 2-е изд., перераб. и доп. - Москва; Санкт-Петербург ; Киев : Вильямс, 2006. - 290 с.

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Для оценки достижений студента используется следующий диагностический инструментарий:

- отчет по лабораторным работам (защита выполненных на лабораторных занятиях индивидуальных заданий);
- проведение устных и письменных опросов;
- тестирование в системе EDU;
- проведение контрольных работ по основным разделам дисциплины;
- коллоквиум;
- дискуссия.

Формой промежуточной аттестации по дисциплине «Основы и методологии программирования» 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 в ходе проведения контрольных мероприятий текущей аттестации.

Примерные весовые коэффициенты, определяющие вклад текущей аттестации в отметку при прохождении промежуточной аттестации:

Формирование отметки за текущую аттестацию:

- отчет по лабораторной работе – 30 %;
- тест – 15%;
- письменный и устный опрос на лекциях – 10 %;
- контрольная работа – 30 %;
- коллоквиум – 15 %.

Итоговая отметка по дисциплине рассчитывается на основе отметки текущей аттестации и экзаменационной отметки с учетом их весовых коэффициентов.

Вес отметки по текущей аттестации составляет 40%, экзаменационной отметки – 60 %.

Примерный перечень заданий для управляемой самостоятельной работы студентов

Управляемая самостоятельная работа (консультационно-методическая поддержка и контроль) в форме аудиторных занятий и дополнительно обеспечивается средствами образовательного портала EDU БГУ.

1.6. Тестирование программ. Unit-тестирование. (2 ч)

Поиск синтаксических, семантических, логических ошибок. UNIT-тестирование выполненных лабораторных работ.

Форма контроля – устный опрос.

2.5. Перегрузка операторов. (2 ч.)

Обработка массивов и матриц с использованием операторов перегрузки.

Форма контроля – отчет по лабораторной работе.

2.7. Использование стандартной библиотеки шаблонных классов STL (2 ч)

Решение типовых задач. Для их эффективного решения необходимо использовать контейнеры библиотеки STL (последовательные, ассоциативные, адаптеры контейнеров).

Форма контроля – дискуссия, отчет по лабораторной работе.

2.8. Элементы функционального программирования. (2 ч)

Решение типовых задач с использованием алгоритмов STL (немодифицирующих и модифицирующих и лямбда выражений). Сравнение и анализ классов обработки данных, используя алгоритмы STL.

Форма контроля – отчет по лабораторной работе, дискуссия.

Примерная тематика лабораторных занятий

Лабораторная работа 1. Алгоритмы обработки числовых данных.

Лабораторная работа 2. Вычислительные алгоритмы.

Лабораторная работа 3. Массивы. Обработка одномерных массивов, понятие vector, array. Алгоритмы сортировки и поиска.

Лабораторная работа 4. Двумерные и многомерные массивы. Обработка матриц.

Лабораторная работа 5. Функции стандартной библиотеки и методов класса string для обработки строк

Лабораторная работа 6. Обработка структурированных данных. Использование текстовых файлов.

Лабораторная работа 7. Класс как абстрактный тип. Разработка классов обработки чисел.

Лабораторная работа 8. Использование шаблонов функций и шаблонов классов для обработки массивов и матриц .

Лабораторная работа 9. Дружественные функции и классы. Обработка строк и массивов. Использование регулярных выражений.

Лабораторная работа 10. Перегрузка операторов. Использование классов чисел, массивов и матриц с использованием исключений.

Лабораторная работа 11. Стандартная библиотека шаблонов STL. Использование классов-контейнеров для обработки данных, методов и алгоритмов для обработки данных.

Лабораторная работа 12. Использование алгоритмов STL. Функциональные объекты и лямбда-выражения. Алгоритмы сортировки. Обобщенные численные алгоритмы.

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса используется **практико-ориентированный подход**, который предполагает:

- освоение содержание образования через решения практических задач;
- приобретение навыков эффективного выполнения разных видов профессиональной деятельности;
- ориентацию на генерирование идей, реализацию групповых студенческих проектов;
- использованию процедур, способов оценивания, фиксирующих профессиональные компетенции.

Методические рекомендации по организации самостоятельной работы студентов

При преподавании дисциплины используется разработанный сотрудниками БГУ учебно-методический комплекс для студентов вузов «Программирование» (<https://yadi.sk/d/7evugQJMcyLNE>).

Для организации самостоятельной работы студентов по учебной дисциплине используются современные информационные ресурсы: размещённые на образовательном портале edufpmi.bsu.by.

Комплекс учебных и учебно-методических материалов (учебно-программные материалы, учебное издание для теоретического изучения дисциплины, методические указания к лабораторным занятиям, материалы текущего контроля и промежуточной аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям

образовательного стандарта высшего образования и учебно-программной документации, в т.ч. задания, тематика творческих заданий и др., список рекомендуемой литературы, информационных ресурсов и др.).

Примерный перечень вопросов к зачету

1. Понятие алгоритмов и способы их описания.
2. Основные понятия и определения (данные и программа).
3. Базовые элементы языка (символы, лексемы, выражения, операторы).
4. Структура программы. Простой пример.
5. Директивы препроцессора.
6. Базовые типы данных. Декларация данных (определение и объявление).
7. Основные операторы языка, приведение типов.
8. Простейшие объекты ввода - вывода данных.
9. Инструкции ветвления и выбора.
10. Инструкции организации циклов.
11. Инструкции передачи управления.
12. Классы памяти (auto, register, static, extern).
13. Определение и объявление функции. Перегрузка функций.
14. Функции. Параметры функций. Использование ссылок. Параметры-значения.
15. Указатели и ссылки, арифметические операторы с указателями. Параметры-указатели.
16. Одномерные массив и указатели. Алгоритмы обработки. Параметры-массивы.
17. Двумерные массив и указатели. Алгоритмы обработки. Параметры-двумерные массивы.
18. Алгоритм сортировки: Обменная сортировка, метод простого выбора, метод вставок, модификация алгоритмов.
19. Алгоритм сортировки: метод Шелла, метод Хоара.
20. Стандартные сортировка и поиск (qsort и bsearch).
21. Символьная информация и строки, функции для обработки.
22. Строки string. Методы для работы со строками.
23. Массив строк. Сортировка слов и строк.
24. Динамическое распределение памяти.
25. Перечисляемый тип (определение, инициализация, основные правила).
26. Структуры. Описание, объявление, характеристики и обработка.
27. Битовые поля структуры. Описание, объявление, характеристики и обработка.

28. Объединения. Описание, объявление, характеристики и обработка.
29. Поточковый ввод-вывод. Открытие и закрытие потока. Стандартные потоки ввода-вывода. Использование манипуляторов для ввода/вывода.
30. Методология объектно-ориентированного программирования.
31. Базовые конструкции объектно-ориентированных программ.
32. Данные и функции класса. Константные объекты. Указатель this.
33. Конструктор и деструктор. Конструктор по умолчанию.
34. Конструктор копирования. Конструктор с параметрами.
35. Регулярные выражения.
36. Шаблоны функций.
37. Дружественные функции классов.
38. Классы для обработки чисел и строк.
39. Перегрузка унарных и бинарных операторов.
40. Перегрузка специальных операторов.
41. Особенности перегрузки оператора присваивания.
42. STL. Итераторы. Основные операции над итераторами. Виды итераторов.
43. STL. Последовательные контейнеры.
44. STL. Ассоциативные контейнеры.
45. STL. Адаптеры контейнеров.
46. Алгоритмы STL. Методы поиска, преобразования, сортировки.
47. Использование лямбда-выражений.

Примерный перечень вопросов к экзамену

1. Состав языка программирования.
Константы и переменные языка. Типы данных. Выражения. Знаки операций.
2. Основные операторы языка программирования.
3. Инструкции языка программирования.
4. Функции и разработка программ.
Объявление и определение функции. Параметры функции.
Перегрузка функций. Функции с переменным числом параметров.
Решение задач целочисленной арифметики с использованием функций.
Математические функции. Функции общего назначения.
5. Массивы.
Объявление и инициализация массива. Инициализация массива случайными числами. Ввод-вывод массивов. Операции над элементами массива.
Передача массивов функциям. Обработка одномерных и двумерных массивов. Поиск элементов с заданными свойствами.
6. Указатели.
Переменные-указатели. Работа с указателями.

Арифметические действия с указателями. Динамическое распределение памяти (ДРП). Указатели и массивы. Массивы – параметры. Указатели на константы и константные указатели. Указатели на функции. Алгоритмы преобразование и перестановки.

7. Сортировка и поиск.

Сортировка пузырьком. Сортировка вставками.

Сортировка выбором. Линейный и бинарный поиск. Стандартные функции сортировки и поиска.

8. Символьная информация и строки. Функции для работы со строками.

9. Строки string. Методы для работы со строками.

10. Объекты ввод-вывода.

11. Типы данных, определяемые пользователем (переименование типов, перечисление, структуры, объединения).

12. Классы.

Определение класса. Макет для описания класса. Понятие инкапсуляции.

Объекты, доступ к членам класса. Указатель this.

Спецификаторы доступа к членам класса. Друзья класса.

Встроенные функции-члены класса.

Статические члены класса.

Вложенные классы.

13. Конструкторы и деструкторы.

Конструктор класса. Список инициализации. Конструктор по умолчанию.

Конструктор копирования. Явный вызов конструкторов.

Деструкторы в базовых классах и потомках.

14. Регулярные выражения. Синтаксис выражений. Методы и классы для обработки.

15. Шаблоны.

Определение шаблона функции. Конкретизация шаблона функции.

Шаблон класса.

16. Дружественные функции и классы.

Определение. Описание, использование.

17. Перегрузка операторов.

Общие правила перегрузки операторов.

Перегрузка унарных операторов. Перегрузка операторов преобразования типов данных (конвертеры).

Перегрузка оператора присваивания =. Перегрузка составных операторов присваивания.

Перегрузка оператора индексирования[]. Перегрузка оператора вызова функции ().

Перегрузка операторов ++ и --.

Перегрузка бинарных операторов. Перегрузка операторов >> и << для ввода-вывода из потоков.

18. Исключения.

Обработка исключений, генерация исключений, стандартные классы для обработки исключений.

19. Использование библиотеки стандартных шаблонных классов.

Последовательные контейнеры: вектор, список, дек.

Ассоциативные контейнеры: словари (map, multimap), множества (set, multiset, bitset).

Адаптеры контейнеров: очередь, стек.

Приоритетная очередь Класс из библиотеки STL - приоритетная очередь.

Итераторы. Основные операции над итераторами. Виды итераторов. Условия для установки итераторов в недействительное состояние.

12. Функциональные объекты, виды функциональных объектов (унарная и бинарная функция, унарный и бинарный предикат).

Стандартные алгоритмы, не меняющие контейнер.

Стандартные алгоритмы, меняющие контейнер.

Стандартные алгоритмы, связанные с сортировкой и эффективным поиском.

Функторы и 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)
Разработка кросс-платформенных приложений	Технологий программирования	Нет	Оставить содержание учебной дисциплины без изменения (протокол № 16 от 18.05.2023)
Машинно-ориентированное программирование	Технологий программирования	Нет	Оставить содержание учебной дисциплины без изменения (протокол № 16 от 18.05.2023)
Промышленное программирование	Технологий программирования	Нет	Оставить содержание учебной дисциплины без изменения (протокол № 16 от 18.05.2022)
Вычислительные методы алгебры	Высшей математики	Нет	Оставить содержание учебной дисциплины без изменения (протокол № 16 от 18.05.2023)

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ
на ____/____ учебный год

№№ Пп	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры технологий программирования (протокол № _ _ г.)

Заведующий кафедрой

_____ профессор д.т.н. _____

(ученая степень, звание)

(подпись)

_____ А.Н. Курбацкий

(И.О. Фамилия)

УТВЕРЖДАЮ

Декан факультета

(ученая степень, звание)

(подпись)

(И.О. Фамилия)