

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«07» марта 2024 г.
Протокол №1

**Рабочая программа дисциплины (модуля)
«Разработка на C++»**

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Математика и компьютерные науки

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2024

**Москва
2024**

Содержание

| | |
|--|----------|
| 1. Краткая характеристика дисциплины (модуля) | 3 |
| 2. Перечень планируемых результатов обучения | 4 |
| 3. Тематический план | 6 |
| 4. Содержание дисциплины (модуля) | 6 |
| 5. Учебно-методическое обеспечение | 7 |
| 6. Материально-техническое обеспечение | 7 |
| 7. Методические и оценочные материалы | 9 |

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Разработка на C++» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 Математика и компьютерные науки, профиль Математика и компьютерные науки, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Разработка на C++» важно для освоения эффективного программирования системного и прикладного уровня, а также для понимания принципов работы с памятью и объектно-ориентированного подхода. Это позволяет создавать производительные и надежные приложения, востребованные в различных областях, от игр до встроенных систем.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 Математика и компьютерные науки, профиль Математика и компьютерные науки и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 3 курсе в 5семестре.

Цель изучения дисциплины (модуля): формирование навыков эффективного программирования на языке C++ для создания высокопроизводительных и надежных программных решений.

Задачи изучения дисциплины (модуля) направлены на формирование у студентов следующий знаний, умений и навыков:

- знание конструкций актуальной версии языка C++;
- знание функций и классов стандартной библиотеки;
- умение решать задачи с помощью языка C++;
- умение писать поддерживаемый код;
- умение работать с динамической памятью;
- навык ведения разработки сложных программ на языке C++.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

| Компетенция | Содержание компетенции | Индикатор компетенции | Перечень планируемых результатов обучения по дисциплине (модулю) |
|-------------|---|-----------------------|---|
| ОПК-4. | Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем | ОПК-4.1. | Знает базовые основы современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности |
| | | ОПК-4.2. | Умеет использовать этот математический аппарат в профессиональной деятельности |
| | | ОПК-4.3. | Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности |
| ОПК-5. | Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности | ОПК-5.1. | Знает технологии, необходимые для прикладного программирования, включая современные функциональные языки программирования, а также основные принципы и понятия, применяемыми при использовании компьютерных сетей |
| | | ОПК-5.2. | Умеет пользоваться технологиями прикладного программирования, включая среды высокоуровневого программирования |
| | | ОПК-5.3. | Имеет практический опыт использования технологий прикладного программирования |
| ОПК-6. | Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения | ОПК-6.1. | Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики в области искусственного интеллекта |
| | | ОПК-6.2. | Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования в области искусственного интеллекта |
| | | ОПК-6.3. | Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований в области искусственного интеллекта |

| | | | |
|-------|---|---------|---|
| ПК-4. | Способен разрабатывать программное обеспечение для решения прикладных задач в области искусственного интеллекта под руководством более опытного специалиста | ПК-4.1. | Знает основные принципы разработки программного обеспечения и методы решения прикладных задач в сфере искусственного интеллекта |
| | | ПК-4.2. | Умеет разрабатывать программное обеспечение под руководством специалистов более высокой категории, используя современные технологии и инструменты |
| | | ПК-4.3. | Имеет опыт участия в разработке программного обеспечения для решения прикладных задач в команде под руководством более опытных специалистов |

3. Тематический план

| №п/п | Наименование раздела дисциплины (модуля) | Трудоемкость, академические часы | | | | ТКУ (текущий контроль успеваемости) |
|--------|---|----------------------------------|-----------|----------|------------------------|---|
| | | <i>Очная форма</i> | | | | |
| | | Контактная работа | | Контроль | Самостоятельная работа | |
| Лекции | Семинары (практические занятия) | | | | | |
| 1 | Инструменты для разработки на С++ | 4 | 4 | | 18 | Подготовка к семинару, Домашние задания |
| 2 | ООП на языке С++ и наследование | 4 | 4 | | 19 | Подготовка к семинару, Домашние задания, Контрольная работа |
| 3 | Шаблоны | 4 | 4 | | 18 | Подготовка к семинару, Домашние задания |
| 4 | Полиморфизм и виртуальные функции | 4 | 4 | | 19 | Подготовка к семинару, Домашние задания, Контрольная работа |
| 5 | Исключения | 4 | 4 | | 18 | Подготовка к семинару, Домашние задания |
| 6 | Контейнеры и итераторы | 4 | 4 | | 19 | Подготовка к семинару, Домашние задания, Контрольная работа |
| 7 | Умные указатели | 4 | 4 | | 19 | Подготовка к семинару, Домашние задания |
| | <i>Зачет с оценкой</i> | | | 4 | | Проект |
| | <i>Итого:</i> | <i>28</i> | <i>28</i> | <i>4</i> | <i>130</i> | |
| | <i>Объем дисциплины (модуля) (в ак. ч.)</i> | <i>190</i> | | | | |
| | <i>Объем дисциплины (модуля) (в зач. ед.)</i> | <i>5</i> | | | | |

4. Содержание дисциплины (модуля)

| №п/п | Наименование раздела дисциплины (модуля) | Содержание дисциплины (модуля) по темам |
|------|--|---|
| 1 | Инструменты для разработки на С++ | Компиляторы и сборщики. Среды разработки (IDE). Отладчики и профилировщики. Системы контроля версий. Автоматизация сборки. |
| 2 | ООП на языке С++ и наследование | Классы и объекты. Инкапсуляция данных. Наследование классов. Модификаторы доступа. Конструкторы и деструкторы. |
| 3 | Шаблоны | Функциональные шаблоны. Классовые шаблоны. Параметризация типов. Специализация шаблонов. Обобщённое программирование. |
| 4 | Полиморфизм и виртуальные функции | Виртуальные методы. Позднее связывание. Абстрактные классы. Переопределение функций. Динамическое распределение. |
| 5 | Исключения | Обработка ошибок. Блоки try-catch. Генерация исключений. Пользовательские исключения. Безопасность ресурсов. |
| 6 | Контейнеры и итераторы | Вектор и список. Множества и словари. Итераторы и алгоритмы. Ассоциативные контейнеры. Управление памятью. |
| 7 | Умные указатели | Автоматическое управление памятью. Семантика владения объектов. Счётчик ссылок. Ослабленные ссылки (weak references). Безопасное совместное использование ресурсов. |

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Мейерс, С. Наиболее эффективное использование C++. 35 новых рекомендаций по улучшению ваших программ и проектов : практическое руководство / С. Мейерс ; пер. с англ. Р. В. Павлова. — 2-е изд. - Москва : ДМК Пресс, 2023. - 298 с. - ISBN 978-5-89818-563-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2107916>.

2. Огнева, М. В. Программирование на языке C++: практический курс : учебник для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563618>.

3. Современный C++: безопасное использование : практическое руководство / Д. Лакос, Р. Витторио, Р. Хлебников, М. Алисдар ; пер. с англ. А. В. Снастина. – Москва : ДМК Пресс, 2023. - 1044 с. – ISBN 978-5-93700-134-4. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2204223>.

4. Гримм, Р. Параллельное программирование на современном C++ : практическое руководство / Р. Гримм ; пер. с англ. В. Ю. Винника. - Москва : ДМК Пресс, 2022. - 618 с. - ISBN 978-5-97060-957-6. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2109583>.

5. Павловская, Т. А. C/C++. Структурное и объектно-ориентированное программирование : практикум / Т. А. Павловская, Ю. А. Щупак. - Санкт-Петербург : Питер, 2021. - 352 с. - (Серия «Учебное пособие»). - ISBN 978-5-4461-9799-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1857042>.

Дополнительная литература:

1. Полуэктова, Н. Р. Разработка веб-приложений : учебник для вузов / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 204 с. — (Высшее образование). — ISBN 978-5-534-18645-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567610>.

2. Зараменских, Е. П. Разработка информационных систем : учебник и практикум для вузов / Е. П. Зараменских. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 78 с. — (Высшее образование). — ISBN 978-5-534-21420-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/571333>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

| № | Наименование портала (издания, курса, документа) | Ссылка |
|----|--|---|
| 1. | Научная электронная библиотека elibrary.ru библиотека | https://elibrary.ru/defaultx.asp |
| 2. | База данных для IT-специалистов | https://habr.com |
| 3. | База данных ScienceDirect | https://www.sciencedirect.com |
| 4. | Официальный сайт Министерства науки и высшего образования Российской Федерации | https://minobrnauki.gov.ru/ |
| 5. | Федеральный портал «Российское образование» | https://www.edu.ru/ |
| 6. | Информационная система "Единое окно доступа к образовательным ресурсам" | http://window.edu.ru/ |
| 7. | Единая коллекция цифровых образовательных ресурсов | http://school-collection.edu.ru/ |
| 8. | Федеральный центр информационно - образовательных ресурсов | http://fcior.edu.ru/ |

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

| Наименование ПО | Производство | Лицензионное / свободно распространяемое |
|--|---------------|--|
| Операционные системы: | | |
| Microsoft Imagine (Windows Client, Server) | зарубежное | лицензионное |
| Браузеры: | | |
| Яндекс.Браузер | отечественное | свободно распространяемое |
| Google Chrome | зарубежное | свободно распространяемое |
| Офисные приложения: | | |
| Microsoft Imagine (Visio, OneNote) | зарубежное | лицензионное |
| TeXstudio | зарубежное | свободно распространяемое |

| | | |
|---|---------------|---------------------------|
| Adobe Acrobat Reader | зарубежное | свободно распространяемое |
| Программное обеспечение для планирования и учета времени: | | |
| Toggle app | зарубежное | свободно распространяемое |
| Системы управления проектами: | | |
| Microsoft Imagine (Project) | зарубежное | лицензионное |
| Системы управления базами данных: | | |
| Microsoft Imagine (SQL Server) | зарубежное | лицензионное |
| Системы резервного копирования (backup): | | |
| Acronis Backup Advanced for HyperV | зарубежное | лицензионное |
| Справочно-правовые системы: | | |
| КонсультантПлюс: справочно-правовая система | отечественное | лицензионное |
| Средства антивирусной защиты: | | |
| Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition | отечественное | лицензионное |
| Среды разработки: | | |
| Visual Studio Code | зарубежное | свободно распространяемое |
| Bash (Unix shell) | зарубежное | свободно распространяемое |
| Anaconda | зарубежное | свободно распространяемое |
| Robotic Operating System | зарубежное | свободно распространяемое |
| CopelliaSim | зарубежное | свободно распространяемое |
| Google Colaboratory | зарубежное | свободно распространяемое |
| Пакеты программных средств и библиотек: | | |
| AutoPsy | зарубежное | свободно распространяемое |
| Interactive Disassembler (IDA) | зарубежное | свободно распространяемое |
| Системы управления библиографической информацией: | | |
| Zotero | зарубежное | свободно распространяемое |
| Сервисы и службы: | | |
| Bind | зарубежное | свободно распространяемое |
| Docker | зарубежное | свободно распространяемое |

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Разработка на С++» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, контрольные работы, домашние задания, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Участие в семинаре (аудиторная работа) – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с

темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы - получить специальные знания по одной или нескольким темам дисциплины (модуля) и продемонстрировать навыки их практического применения.

Проект – исследовательская работа по курсу и презентация результатов.

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов, планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Разработка на C++»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме **зачета с оценкой**, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

| Десятибалльная оценка | Пятибалльная оценка | Оценка за зачет | Общая характеристика результата обучения по дисциплине (модулю) |
|-----------------------|---------------------|-----------------|---|
| 10 | Отлично | Зачтено | Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, |
| 9 | Отлично | Зачтено | |
| 8 | Отлично | Зачтено | |

| Десятибалльная оценка | Пятибалльная оценка | Оценка за зачет | Общая характеристика результата обучения по дисциплине (модулю) |
|-----------------------|---------------------|-----------------|--|
| | | | акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами. |
| 7 | Хорошо | Зачтено | Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами. |
| 6 | Хорошо | Зачтено | Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами. |
| 5 | Удовлетворительно | Зачтено | Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования. |
| 4 | Удовлетворительно | Зачтено | Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования. |
| 3 | Не сдан | Не зачтено | Студент не овладел обязательным |

| Десятибалльная оценка | Пятибалльная оценка | Оценка за зачет | Общая характеристика результата обучения по дисциплине (модулю) |
|-----------------------|---------------------|-----------------|---|
| 2 | Не сдан | Не зачтено | минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы. |
| 1 | Не сдан | Не зачтено | |

Дисциплина (модуль) «Разработка на C++» оценивается следующим образом:

| Активность | Вес | Количество | Описание |
|--------------------|-----|------------|---|
| Домашние задания | 20% | 13 | Набор задач по темам недели |
| Аудиторная работа | 15% | 14 | Активная работа студента на семинаре |
| Контрольные работы | 30% | 3 | Письменная работа с набором задач, которые нужно решить за ограниченное время |
| Зачет с оценкой | 35% | 1 | Защита итогового проекта |

Формула расчёта итоговой оценки по дисциплине (модулю) «Разработка на C++»
« $0,2 \times$ среднее за домашние задания + $0,15 \times$ аудиторная работа + $0,3 \times$ среднее за контрольные работы + $0,35 \times$ зачет с оценкой».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание по теме «Компиляторы и сборщики. Среды разработки (IDE)»

1. Установите и настройте компилятор C++ (например, GCC или MSVC) на своей машине.
2. Создайте и соберите простой проект "Hello, World!" с использованием командной строки.
3. Установите и настройте IDE (например, Visual Studio, CLion или Code::Blocks).
4. Настройте проект в IDE и выполните отладку простой программы.
5. Исследуйте и примените базовые параметры оптимизации компилятора.

Домашнее задание по теме «Классы и объекты. Инкапсуляция данных»

1. Создайте класс "Книга" с закрытыми полями (название, автор, год издания) и методами доступа.
2. Реализуйте конструктор и деструктор для класса "Книга".
3. Напишите программу, создающую несколько объектов класса и выводящую их данные.
4. Добавьте методы для изменения и чтения полей с проверкой корректности данных.
5. Реализуйте копирующий конструктор и оператор присваивания для класса.

Домашнее задание по теме «Функциональные и классовые шаблоны»

1. Напишите шаблон функции для обмена значениями двух переменных.
2. Создайте шаблон класса "Пара" для хранения двух значений разных типов.
3. Реализуйте метод для вывода содержимого шаблонного класса "Пара".
4. Напишите программу, демонстрирующую использование шаблонных функций и классов с разными типами.
5. Исследуйте и примените специализацию шаблонов для конкретного типа данных.

Примерные вопросы для подготовки к семинарам

Вопросы к семинару по теме «Виртуальные методы. Позднее связывание»

1. Что такое виртуальный метод в C++ и как он объявляется?
2. Как работает механизм позднего связывания и зачем он нужен?
3. В чём разница между виртуальным и неvirtуальным методом?
4. Как влияет виртуальный метод на размер и структуру объекта?
5. Какие проблемы могут возникнуть при использовании виртуальных методов?

Вопросы к семинару по теме «Обработка ошибок. Блоки try-catch»

1. Как устроена обработка исключений в C++?
2. Как объявляются блоки try, catch и что они делают?
3. Какие типы данных можно использовать для генерации исключений?
4. Как реализовать пользовательское исключение в C++?
5. Как обеспечить безопасное освобождение ресурсов при возникновении исключений?

Вопросы к семинару по теме «Вектор и список. Множества и словари»

1. Как устроена обработка исключений в C++?
2. Как объявляются блоки try, catch и что они делают?
3. Какие типы данных можно использовать для генерации исключений?
4. Как реализовать пользовательское исключение в C++?
5. Как обеспечить безопасное освобождение ресурсов при возникновении исключений?

Примерные задания по контрольным работам

Контрольная работа № 1

1. Установите и настройте компилятор C++ и среду разработки (IDE) на вашем компьютере.
2. Создайте простой проект в IDE с программой «Hello, World!» и выполните его сборку и запуск.
3. Используя отладчик IDE, поставьте точку останова и пошагово выполните программу, наблюдая значения переменных.
4. Настройте систему контроля версий (например, Git) для вашего проекта и выполните базовые операции: инициализацию репозитория, коммит, создание ветки.
5. Напишите простой Makefile или скрипт для автоматизации сборки вашего проекта.
6. Определите класс «Автомобиль» с закрытыми полями (марка, год выпуска), добавьте публичные методы доступа.
7. Реализуйте конструктор и деструктор для класса «Автомобиль», добавьте вывод сообщений при их вызове.
8. Создайте класс-наследник «Электромобиль», расширив класс «Автомобиль» дополнительным полем (ёмкость батареи).
9. Продемонстрируйте работу модификаторов доступа (private, protected, public) на примере классов и наследования.
10. Напишите программу, создающую объекты классов «Автомобиль» и «Электромобиль», и выведите их свойства на экран.

Контрольная работа № 2

1. Напишите шаблонную функцию для обмена значениями двух переменных любого типа.
2. Создайте шаблонный класс «Пара», хранящий два значения произвольных типов, и реализуйте методы доступа к ним.
3. Реализуйте полную специализацию шаблонного класса «Пара» для типа int.

4. Напишите программу, демонстрирующую работу шаблонной функции и класса с различными типами данных.
5. Объясните, что такое виртуальный метод и зачем используется позднее связывание.
6. Создайте базовый класс с виртуальной функцией и класс-наследник, переопределяющий эту функцию.
7. Продемонстрируйте вызов виртуальной функции через указатель на базовый класс, ссылающийся на объект наследника.
8. Определите абстрактный класс с чисто виртуальной функцией и реализуйте наследников, предоставляющих конкретную реализацию.
9. Продемонстрируйте динамическое распределение объектов базового и производных классов с использованием оператора `new`.
10. Объясните разницу между переопределением (`override`) и перегрузкой (`overload`) функций в контексте виртуальных методов.

Контрольная работа № 3

1. Напишите программу, генерирующую исключение при делении на ноль, и обработайте его в блоке `try-catch`.
2. Создайте пользовательский класс исключения с сообщением об ошибке и используйте его в программе.
3. Объясните, как обеспечить безопасность ресурсов при возникновении исключений (RAII).
4. Реализуйте класс с управлением динамической памятью, корректно обрабатывающий исключения в конструкторе.
5. Создайте и заполните контейнер `std::vector` целыми числами, затем выведите их на экран.
6. Используйте `std::list` для хранения строк и реализуйте добавление и удаление элементов.
7. Продемонстрируйте создание и использование `std::set` для хранения уникальных элементов.
8. Создайте `std::map` для хранения пар «ключ-значение» и реализуйте поиск по ключу.
9. Используйте итераторы для обхода элементов `std::vector` и `std::map`, выводя их содержимое.
10. Примените алгоритм `std::sort` к вектору и объясните, как работает сортировка с итераторами.

Примерное описание и критерии оценивания к проекту

Описание проекта:

Итоговый проект представляет собой комплексную программную систему, разработанную на языке C++, демонстрирующую владение ключевыми концепциями и технологиями, изученными в ходе курса. Проект должен включать создание и организацию классов с применением принципов объектно-ориентированного программирования (ООП), использование наследования и полиморфизма через виртуальные функции и абстрактные классы, а также эффективное применение шаблонов для обобщённого программирования.

В реализации должны быть продемонстрированы навыки работы с инструментами разработки: настройка и использование компилятора, сборщика, системы контроля версий, а также автоматизация сборки проекта. Особое внимание уделяется обработке ошибок с помощью исключений, обеспечению безопасности ресурсов и управлению памятью с использованием умных указателей.

Проект должен активно использовать контейнеры стандартной библиотеки (STL) и итераторы для организации и обработки данных, а также включать применение алгоритмов STL для решения практических задач.

Критерии оценивания:

1. Архитектура и структура проекта

- Логическая организация кода, модульность и читаемость.
- Корректное использование классов, инкапсуляции и модификаторов доступа.
- Применение наследования и полиморфизма с правильной реализацией виртуальных функций и абстрактных классов.

2. Использование шаблонов

- Реализация функциональных и классовых шаблонов.
- Корректная параметризация типов и применение специализации шаблонов.
- Обоснованное использование шаблонов для повышения переиспользуемости кода.

3. Обработка ошибок и исключения

- Наличие и корректность блоков try-catch.
- Генерация и обработка пользовательских исключений.
- Обеспечение безопасности ресурсов при возникновении исключений (RAII-подход).

4. Работа с контейнерами STL и алгоритмами

- Использование различных контейнеров (vector, list, set, map) по назначению.
- Применение итераторов для обхода и модификации данных.
- Использование стандартных алгоритмов STL для решения задач.

5. Управление памятью и умные указатели

- Корректное применение умных указателей (unique_ptr, shared_ptr, weak_ptr).
- Отсутствие утечек памяти и ошибок владения ресурсами.
- Обеспечение безопасного совместного использования объектов.

6. Инструменты разработки и автоматизация

- Использование системы контроля версий с понятной историей изменений.
- Наличие автоматизированной сборки проекта (Makefile, CMake или скрипты).
- Использование отладчика и профилировщика для тестирования и оптимизации (при возможности).

7. Качество кода и документация

- Читаемость, стиль кодирования и комментарии.
- Наличие описания архитектуры и инструкций по сборке и запуску проекта.
- Тестовые примеры и демонстрация работы ключевых функциональностей.

8. Функциональность и соответствие требованиям

- Полнота реализации заявленных функциональных возможностей.
- Корректность работы программы без критических ошибок и сбоев.
- Эффективность и оптимальность решений.

Задания для промежуточной аттестации по дисциплине (модулю)

| № п/п | Задание | Ответ | Компетенция |
|-------|--|-------|-------------|
| 1. | Что из перечисленного является системой контроля версий? а) CMake б) GDB в) Valgrind г) Git | г | ОПК-4 |
| 2. | Какой модификатор доступа в C++ позволяет наследникам класса иметь доступ к членам базового класса, но закрывает их от внешнего кода? а) public б) private в) protected | в | ОПК-4 |

| | | | |
|-----|--|---------------|-------|
| | г) friend | | |
| 3. | Какой умный указатель в C++ обеспечивает подсчёт ссылок и автоматическое удаление объекта при отсутствии владельцев? а) std::unique_ptr б) std::shared_ptr в) std::weak_ptr г) raw pointer | б | ОПК-4 |
| 4. | Что обеспечивает позднее связывание в C++? а) Статическая типизация б) Виртуальные функции в) Макросы препроцессора г) Конструкторы | б | ОПК-5 |
| 5. | Какой блок используется для обработки исключений в C++? а) for-loop б) if-else в) switch-case г) try-catch | г | ОПК-5 |
| 6. | Какой контейнер из стандартной библиотеки C++ обеспечивает хранение элементов в отсортированном виде и обеспечивает быстрый поиск? а) std::vector б) std::list в) std::set г) std::deque | в | ОПК-5 |
| 7. | Что из перечисленного является средой разработки (IDE) для C++? а) Visual Studio б) GCC в) Make г) Valgrind | а | ОПК-6 |
| 8. | Какой термин описывает класс, который содержит хотя бы одну чисто виртуальную функцию и не может быть инстанцирован? а) Абстрактный класс б) Интерфейс в) Базовый класс г) Структура | а | ОПК-6 |
| 9. | Как называется параметризация типов в C++? | Шаблоны | ОПК-6 |
| 10. | Как называется функция, вызываемая при создании объекта класса? | Конструктор | ПК-4 |
| 11. | Как называется объект, который позволяет проходить по элементам контейнера? | Итератор | ПК-4 |
| 12. | Какой умный указатель предотвращает циклические ссылки? | std::weak_ptr | ПК-4 |