ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«24» июня 2025 г.
Протокол №2

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

Направление подготовки: *02.03.01 Математика и компьютерные науки*

Направленность (профиль) подготовки: *Разработка*

Квалификация (степень) выпускника: *бакалавр*

Форма обучения: *очная*

Срок освоения программы: *4 года*

Год набора: *2025*

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 0215B0C000D1B3FEB44A37EE631E0C27D4
Владелец: Ивашкевич Евгений Васильевич
Действителен: с 13.01.2026 до 13.04.2027

Электронный документ

Москва
2025

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 направлению 02.03.01 «Математика и компьютерные науки», профиль «Разработка», 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 «Разработка» и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений.

Дисциплина (модуль) является выборной и доступна для изучения на 3 или 4 курсе в 5, 6, или 7 семестре на выбор.

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

Задачи изучения дисциплины (модуля):

— изучить основные концепции и модели облачных вычислений, включая сервисы IaaS, PaaS и SaaS;

— освоить принципы проектирования и развертывания облачных архитектур с учетом безопасности и масштабируемости;

— научиться применять инструменты и платформы облачных провайдеров для решения практических задач в разработке программного обеспечения.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

— конструкции актуальной версии языка C++;

— функции и классы стандартной библиотеки;

уметь:

— решать задачи с помощью языка C++;

— писать поддерживаемый код;

— работать с динамической памятью;

владеть:

— навыком 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)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области разработки, основные принципы критической оценки источников информации и их релевантности
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
ОПК-1.	Способен консультировать и использовать фундаментальные знания в области математического анализа, комплексного и функционального анализа алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов, теоретической механики в профессиональной деятельности	ОПК-1.1.	Знает основные концепции и теории в области математического анализа и смежных дисциплин; методы и подходы, используемые в различных областях математики
		ОПК-1.2.	Умеет применять математические методы для решения профессиональных задач
		ОПК-1.3.	Имеет практический опыт разработки и реализация математических моделей в профессиональной деятельности
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-6.	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-6.1.	Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики
		ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования
		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований
ПК-3.	Способен использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет разрабатывать и применять математические модели и алгоритмы для решения различных задач, анализируя полученные результаты
		ПК-3.3.	Имеет практический опыт использования методов математического и алгоритмического моделирования в реальных проектах или исследованиях

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Контактная работа		Контроль	Самостоятельная работа	
		Лекции	Семинары (практические занятия)			
1	Введение в язык	4	4		30	Домашнее задание
2	Объектно-ориентированное программирование	8	6	2	30	Домашнее задание, Контрольная работа
3	Шаблоны и обобщенное программирование	8	6	2	30	Домашнее задание, Контрольная работа
4	Разработка backend-сервисов на C++	10	8	2	36	Домашнее задание, Проект
	Зачет с оценкой			4		
	Итого:	***30***	***24***	***10***	***126***	
Объем дисциплины (модуля) (в ак. ч.)		***190***				
Объем дисциплины (модуля) (в зач. ед.)		***5***				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Введение в язык	Основы языка. Типы данных. Отличия C++ от Си Указатели и массивы. Функции. Ссылки и константы.
2	Объектно-ориентированное программирование	Структуры и классы. Инкапсуляция. Перегрузка операторов. Конструкторы и деструкторы. Правило трех. Наследование. Приведения типов. Множественное наследование Полиморфизм. Виртуальные функции. RTTI
3	Шаблоны и обобщенное программирование	Шаблоны функций и классов. Специализации шаблонов Исключения. RAII. Умные указатели Стандартные контейнеры. Итераторы. Стандартные алгоритмы Семантика перемещения. Правило пяти
4	Разработка backend-сервисов на C++	Сборка проекта. Линковка. Разделяемые библиотеки Тестирование, отладка и профилирование Сетевое взаимодействие Реализация многопоточных программ Асинхронное программирование

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 C++» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, контрольные работы, 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 удается, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

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, если иное не указано в силлабусе дисциплины:

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, акцентируя внимание на наиболее
9	Отлично	Зачтено	
8	Отлично	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			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	
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	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не
2	Не сдан	Не зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
1	Не сдан	Не зачтено	может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.

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

Активность	Вес	Количество	Описание
Домашнее задание	20%	6	Набор задач по темам недели
Контрольная работа	20%	2	Письменная работа с набором задач, которые нужно решить за ограниченное время
Проект	30%	1	Защита итогового проекта
Зачет с оценкой	30%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по курсу

Формула расчёта итоговой оценки по дисциплине (модулю) «Разработка на C++»: «0,2 × среднее за домашние задания + 0,2 × среднее за контрольные работы + 0,3 × проект + 0,3 × зачет с оценкой».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание по теме «Введение в язык»

Задание: «Консольный менеджер склада»

Разработать консольное приложение для учета товаров на складе.

Требования к реализации:

1. Структуры данных: Описать struct Product (ID, название, количество, цена).

2. Динамическая память: Реализовать динамический массив структур Product вручную (используя new/delete), а не std::vector. Реализовать функции добавления товара и удаления товара со сдвигом элементов.

3. Функции и ссылки: Написать функции для вывода списка товаров и поиска по ID. Аргументы, которые не изменяются внутри функции, должны передаваться по const-ссылке.

4. Указатели: Реализовать функцию сортировки товаров по цене, используя указатели на элементы массива.

5. Сравнение C и C++: В одном из модулей программы использовать C-строки (char*), в другом — std::string. Прокомментировать в коде различия в работе с ними.

Домашнее задание по теме «Объектно-ориентированное программирование»

Задание: «Система расчета заработной платы»

Разработать систему для расчета зарплаты сотрудников разных типов.

Требования к реализации:

1. Иерархия классов: Базовый абстрактный класс Employee. Производные классы: Manager (оклад + премия), Developer (оплата за часы), Consultant (фиксированная ставка за проект).

2. Инкапсуляция: Все поля данных должны быть private. Доступ к ним — через публичные методы (геттеры/сеттеры) или конструктор.

3. Полиморфизм: Виртуальный метод calculateSalary() и виртуальный деструктор. В main использовать массив указателей на базовый класс Employee*.

4. Правило трех: Для класса Manager реализовать явный конструктор копирования, оператор присваивания и деструктор (например, если внутри класса есть динамически выделенный массив подчиненных).

5. Перегрузка операторов: Перегрузить оператор == для сравнения сотрудников по ID и оператор << для вывода информации об сотруднике в поток.

6. Наследование: Реализовать множественное наследование для класса SeniorDeveloper, который наследуется от Developer и интерфейса Mentor (метод mentorJunior()).

Домашнее задание по теме «Шаблоны и обобщенное программирование»

Задание: «Универсальный контейнер и анализ данных»

Часть 1: Реализация собственного шаблонного контейнера.

Часть 2: Использование STL для решения прикладной задачи.

Требования к реализации:

1. Шаблонный класс: Реализовать упрощенный аналог std::vector под названием SimpleVector<T>.

2. Правило пяти: Реализовать конструктор, деструктор, копирующий конструктор, копирующий оператор присваивания, перемещающий конструктор и оператор присваивания (move semantics).

3. Исключения: Бросать исключения std::out_of_range при неверном доступе к элементу.

4. STL и Алгоритмы: Во второй части задания использовать std::vector, std::map, std::string. Написать программу для анализа текста (подсчет частоты слов). Использовать стандартные алгоритмы (std::sort, std::find, std::transform).

5. Умные указатели: Везде, где требуется динамическая память, использовать std::unique_ptr или std::shared_ptr. Raw pointers запрещены.

6. Специализация: Сделать специализацию шаблона SimpleVector<bool> для оптимизации памяти (или просто вывести сообщение о специализации).

Домашнее задание по теме «Разработка backend-сервисов на C++»

Задание: «Многопоточный TCP-эхо сервер»

Разработать серверное приложение, принимающее соединения от клиентов и возвращающее полученные данные обратно.

Требования к реализации:

1. Сборка проекта: Проект должен собираться через CMake. Код разделен на статическую библиотеку (логика сервера) и исполняемый файл (точка входа).

2. Сетевое взаимодействие: Использование сокетов (BSD sockets или кроссплатформенная библиотека, например, Boost.Asio). Сервер должен слушать определенный порт.

3. Многопоточность: Для обработки каждого клиента выделять отдельный поток (std::thread) или использовать пул потоков. Реализовать безопасный доступ к общему логу сервера (мьютексы, блокировки).

4. Асинхронность: Реализовать неблокирующий ввод-вывод или использовать std::async для длительных задач (например, искусственная задержка ответа).

5. Тестирование: Написать модульные тесты с использованием фреймворка (Google Test / Catch2). Покрытие тестами не менее 70% бизнес-логики.

6. Профилирование: Предоставить отчет (скриншоты или лог) работы профилировщика (например, perf, Valgrind --tool=callgrind или встроенных средств IDE), показывающий отсутствие узких мест в критических секциях.

Примерные задания для контрольной работы

Контрольная работа по теме «Объектно-ориентированное программирование»

1. Назовите ключевые различия между class и struct в C++, включая модификаторы доступа по умолчанию и примеры предпочтительного использования каждой конструкции.
2. Дан код иерархии классов Base и Derived. Напишите порядок вывода сообщений в консоль при создании объекта Derived и его последующем уничтожении. Объясните, почему деструктор базового класса должен быть виртуальным.
3. Класс содержит поле-указатель на динамически выделенную память (int* buffer). Напишите реализацию конструктора копирования и оператора присваивания для этого класса, обеспечивающую глубокое копирование (deep copy). Объясните, что произойдет, если использовать реализацию по умолчанию.
4. Реализуйте класс Complex (комплексное число). Перегрузите операторы + (сложение), - (вычитание) и << (вывод в поток ostream). Операторы должны быть реализованы как дружественные функции или методы класса (обоснуйте выбор).
5. Создайте базовый класс Shape с защищенными (protected) полями width и height. Создайте класс Rectangle, наследующий Shape. Реализуйте метод getArea() в классе Rectangle. Объясните, почему поля не должны быть private в данном случае и какие существуют способы доступа к ним при private наследовании.
6. Дан указатель базового класса Animal*, который указывает на объект класса Dog. В базовом классе есть метод makeSound(), в производном он переопределен. Что нужно сделать в объявлении базового класса, чтобы при вызове через указатель базового типа выполнился метод производного класса? Что такое позднее связывание?
7. Объявите абстрактный класс Figure, содержащий чисто виртуальную функцию draw() = 0. Можно ли создать объект типа Figure? Для чего используются такие классы? Приведите пример класса-наследника, который реализует этот интерфейс.
8. Имеется массив указателей на базовый класс Base*. Необходимо найти все объекты, которые фактически являются объектами класса Derived, и вызвать метод специфичный только для Derived. Какой оператор приведения типов необходимо использовать? Напишите фрагмент кода с проверкой типа.
9. Опишите проблему ромбовидного наследования (Diamond Problem). Как ключевое слово virtual при наследовании решает эту проблему? Нарисуйте схему классов и покажите, сколько экземпляров базового класса будет существовать в памяти с virtual наследованием и без него.
10. Разработайте иерархию классов для системы хранения данных: базовый класс Storage (виртуальный деструктор, метод save()), наследники FileStorage и DatabaseStorage. Реализуйте фабричный метод, возвращающий умный указатель (std::unique_ptr) на базовый тип в зависимости от входного параметра. Продемонстрируйте полиморфное поведение метода save().?

Контрольная работа по теме «Шаблоны и обобщенное программирование»

1. Напишите синтаксис объявления шаблонной функции swap для произвольного типа T.
2. В чем разница между полной и частичной специализацией шаблона класса?
3. Что такое стек вызовов (stack unwinding) при возникновении исключения?
4. Расшифруйте аббревиатуру RAII и приведите пример ее использования в стандартной библиотеке.
5. Когда следует использовать std::unique_ptr, а когда std::shared_ptr?
6. Какая сложность доступа по индексу в std::vector и std::list соответственно?

7. Какие пять категорий итераторов существуют в STL?
8. Для чего предназначен алгоритм std::transform и какие аргументы он принимает?
9. Что делает функция std::move и меняет ли она состояние объекта?
10. Перечислите пять специальных методов класса, которые нужно определить при управлении ресурсами вручную?

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	Укажите ключевое слово в C++, используемое для объявления ссылки на неизменяемые данные.	const	ОПК-6
2	В чем заключается основное отличие ссылки от указателя в C++?	ссылка не может быть null / должна быть инициализирована	ОПК-6
3	Какой механизм используется для реализации полиморфизма времени выполнения в C++?	виртуальные функции / vtable	ОПК-6
4	Назовите три компонента, входящие в «Правило трех» для управления ресурсами класса.	деструктор, конструктор копирования, оператор присваивания	ОПК-6
5	Какой умный указатель стандарта C++11 реализует эксклюзивное владение объектом?	std::unique_ptr	ОПК-6
6	Расшифруйте аббревиатуру RAII, описывающую принцип управления ресурсами.	Resource Acquisition Is Initialization	ОПК-6
7	Какое ключевое слово используется для реализации семантики перемещения в функциях?	std::move / && (rvalue reference)	ОПК-6
8	Укажите контейнер STL, обеспечивающий хранение пар «ключ-значение» в отсортированном виде.	std::map	ОПК-4
9	Какой тип наследования может привести к проблеме «ромбовидного наследования»?	множественное наследование	ОПК-6
10	Укажите инструмент сборки, стандартно используемый для управления проектами на C++.	CMake / Make	УК-1
11	Назовите популярный фреймворк для модульного тестирования кода на C++.	Google Test / Catch2	УК-1
12	Какой инструмент используется для обнаружения утечек памяти в Linux-среде?	Valgrind	УК-1
13	Укажите примитив синхронизации для защиты общих данных в многопоточной программе.	std::mutex / мьютекс	ПК-3
14	Какая модель программирования используется для неблокирующего сетевого ввода-вывода?	асинхронная / async	ПК-3
15	Какой тип линковки создает отдельный файл библиотеки, загружаемый во время выполнения?	динамическая / shared library	ОПК-6
16	Укажите ключевое слово для обработки исключений в блоке кода.	try / catch	ОПК-6
17	Какова временная сложность доступа к элементу вектора по индексу?	O(1) / константная	ОПК-4
18	Какой тип данных используется для представления чисел с плавающей запятой	float	ОПК-1

	одинарной точности?		
19	Укажите категорию итераторов, поддерживающих только однонаправленный проход.	input iterator / forward iterator	ОПК-4
20	Как называется механизм проверки типа объекта во время выполнения в иерархии классов?	RTTI / dynamic_cast	ПК-3