

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Лиодинович
Должность: Ректор
Дата подписания: 06.04.2026 13:53:03
Уникальный программный ключ:
5cf0d6f89e80f49a334f6a4ba58e91f3326b9926

Приложение А

(обязательное к рабочей программе дисциплины)

**Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Дагестанский государственный технический университет»**


ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Программирование на языке C++»

Уровень образования	<u>специалитет</u> (бакалавриат/магистратура/специалитет)
Специальность	<u>10.05.03 Информационная безопасность автоматизированных систем</u> (код, наименование специальности)
Специализация	<u>Безопасность открытых информационных систем</u> (наименование)

Разработчик  Качаева Г.И.
подпись (ФИО уч. степень, уч. звание)

Фонд оценочных средств обсужден на заседании кафедры ИБиПИ «15» октября 2025г., протокол № 2

Зав. кафедрой  Качаева Г.И.
подпись (ФИО уч. степень, уч. звание)

г. Махачкала 2025

СОДЕРЖАНИЕ

1. Область применения, цели и задачи фонда оценочных средств	3
2. Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля)	3
2.1. Перечень компетенций с указанием этапов их формирования в процессе освоения ОПОП	4
2.1.2. Этапы формирования компетенций	7
2.2. Показатели уровней сформированности компетенций на этапах их формирования, описание шкал оценивания	9
2.2.1. Показатели уровней сформированности компетенций на этапах их формирования	9
2.2.2. Описание шкал оценивания	11
3. Типовые контрольные задания, иные материалы и методические рекомендации, необходимые для оценки сформированности компетенций в процессе освоения ОПОП	12
3.1. Задания и вопросы для входного контроля	12
3.2. Оценочные средства и критерии сформированности компетенций	12
3.2.1. Аттестационная контрольная работа №1	12
3.2.2. Аттестационная контрольная работа №2	12
3.2.3. Аттестационная контрольная работа №3	13
3.2.4. Список вопросов к зачету	13
3.3. Вопросы по остаточным знаниям	14
3.4. Задания и вопросы для входного контроля	14
3.5. Оценочные средства и критерии сформированности компетенций	15
3.5.1. Аттестационная контрольная работа №1	15
3.5.2. Аттестационная контрольная работа №2	15
3.5.3. Аттестационная контрольная работа №3	15
3.6. Список вопросов к зачету	16
3.7. Вопросы по остаточным знаниям	16

1. Область применения, цели и задачи фонда оценочных средств

Фонд оценочных средств (ФОС) является неотъемлемой частью рабочей программы дисциплины «Программирование на языке С++» и предназначен для контроля и оценки образовательных достижений обучающихся (в т.ч. по самостоятельной работе студентов, далее – СРС), освоивших программу данной дисциплины.

Целью фонда оценочных средств является установление соответствия уровня подготовки обучающихся требованиям ФГОС ВО по специальности 10.05.03 Информационная безопасность автоматизированных систем.

Рабочей программой дисциплины «Программирование на языке С++» предусмотрено формирование следующих компетенций:

ОПК-2 Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности.

ОПК-7 Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ

2. Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля)

Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля), и используемые оценочные средства приведены в таблице 1.

Перечень оценочных средств, рекомендуемых для заполнения таблицы 1 (в ФОС не приводится, используется только для заполнения таблицы)

- *Эссе*
- *Устный опрос*
- *Вопросы для проведения зачета*

2.1. Перечень компетенций с указанием этапов их формирования в процессе освоения ОПОП

Таблица 1

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Критерии оценивания	Наименование контролируемых разделов и тем ¹
<p>ОПК-2 Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности</p>	<p>ОПК-2.1.3 знает состав, назначение аппаратных средств и программного обеспечения персонального компьютера</p>	<p>Знать: Архитектуру персонального компьютера, назначение и взаимодействие основных аппаратных компонентов. Классификацию и назначение программного обеспечения для разработки на C++. Особенности трансляции C++ программ и их связь с аппаратным обеспечением. Уметь: Настраивать среду разработки и инструментарий для компиляции и отладки C++ программ. Использовать системные утилиты для мониторинга ресурсов при выполнении программ. Анализировать влияние аппаратных характеристик на производительность и выбирать оптимальные алгоритмы с учётом архитектуры целевой платформы. Владеть: Навыками работы с компиляторами и системами сборки для создания исполняемых модулей. Методами отладки с использованием интегрированных сред и специализированных инструментов. Приёмами профилирования кода для выявления узких мест и оптимизации под конкретную аппаратную платформу.</p>	<p>№№ 1-17</p>
<p>ОПК-7 Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для</p>	<p>ОПК-7.1.1 знает общие принципы построения, области и особенности применения языков программирования высокого уровня</p>	<p>Знать: Основные парадигмы программирования и их реализацию в языках высокого уровня. Принципы компиляции и интерпретации, этапы трансляции программы. Области применения различных языков, преимущества и недостатки C++ по сравнению с другими языками в контексте системного программирования, разработки высоконагруженных приложений и задач информационной безопасности. Стандарты языка C++ и их эволюцию. Уметь: Анализировать требования задачи и обоснованно выбирать</p>	<p>№№ 1-17</p>

¹ Наименования разделов и тем должен соответствовать рабочей программе дисциплины.

решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ		<p>язык программирования и парадигму для её решения. Сравнить синтаксические и семантические особенности языков. Оценивать применимость различных языковых конструкций для обеспечения эффективности, надёжности и безопасности программного обеспечения.</p> <p>Владеть: Терминологическим аппаратом в области языков программирования. Навыками критического анализа существующих языковых решений. Методологией выбора инструментария программирования в зависимости от контекста решаемой профессиональной задачи.</p>	
	ОПК-7.1.2 знает язык программирования высокого уровня (объектно-ориентированное программирование)	<p>Знать: Синтаксис и семантику языка C++, включая базовые типы данных, операторы, управляющие конструкции, функции, указатели, ссылки, массивы, строки. Основные принципы объектно-ориентированного программирования: инкапсуляцию, наследование, полиморфизм. Механизмы классов, конструкторов и деструкторов, виртуальных функций, абстрактных классов, множественного наследования. Правило трёх/пяти, семантику перемещения. Шаблоны функций и классов, обработку исключений.</p> <p>Уметь: Проектировать и реализовывать на C++ иерархии классов с использованием принципов ООП. Применять наследование для повторного использования кода и полиморфизм для создания гибких и расширяемых систем. Корректно управлять ресурсами с помощью конструкторов, деструкторов и умных указателей. Использовать шаблоны для создания обобщённых компонентов.</p> <p>Владеть: Навыками написания корректного и эффективного объектно-ориентированного кода на C++. Приёмами перегрузки операторов и создания дружественных функций. Методами обработки исключительных ситуаций для обеспечения надёжности программ. Инструментарием стандартной библиотеки C++ в сочетании с пользовательскими классами.</p>	№№ 1-17
	ОПК-7.2.2 умеет разрабатывать и реализовывать на языке высокого уровня алгоритмы	<p>Знать: Типовые алгоритмы обработки данных и их реализацию на C++. Возможности стандартной библиотеки шаблонов для работы с контейнерами, итераторами и алгоритмами. Основные подходы к разработке многопоточных приложений. Принципы сетевого</p>	№№ 1-17

	<p>решения типовых профессиональных задач</p>	<p>взаимодействия. Методы обеспечения безопасности кода.</p> <p>Уметь: Разрабатывать программы на C++, реализующие алгоритмы решения задач профессиональной направленности. Эффективно использовать STL для упрощения кода и повышения его надёжности. Применять лямбда-выражения, алгоритмы STL и итераторы для обработки коллекций данных. Создавать многопоточные приложения с корректной синхронизацией доступа к общим ресурсам. Реализовывать клиент-серверные приложения с использованием сетевых возможностей.</p> <p>Владеть: Навыками разработки законченных консольных приложений на C++ для решения конкретных задач. Методами отладки и тестирования создаваемых программ. Инструментами статического и динамического анализа кода для выявления потенциальных уязвимостей и ошибок. Приёмами документирования кода и соблюдения стандартов оформления.</p>	
--	---	--	--

2.1.2. Этапы формирования компетенций

Сформированность компетенций по дисциплине «Программирование на языке C++» определяется на следующих этапах:

1. **Этап текущих аттестаций** (Для проведения текущих аттестаций могут быть использованы оценочные средства, указанные в разделе 2)

2. **Этап промежуточных аттестаций** (Для проведения промежуточной аттестации могут быть использованы другие оценочные средства)

Таблица 2

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Этапы формирования компетенции					Этап промежуточной аттестации
		Этап текущих аттестаций				18-20 неделя	
		1-5 неделя	6-10 неделя	11-15 неделя	1-17 неделя		
		Текущая аттестация №1	Текущая аттестация №2	Текущая аттестация №3	СРС		КР/КП
1	2	3	4	5	6	7	
ОПК-2 Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности	ОПК-2.1.3 знает состав, назначение аппаратных средств и программного обеспечения персонального компьютера	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3			Вопросы для проведения зачета
ОПК-7 Способен создавать программы на языках общего назначения,	ОПК-7.1.1 знает общие принципы построения, области и особенности применения языков	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3			Вопросы для проведения зачета

применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	программирования высокого уровня						
	ОПК-7.1.2 знает язык программирования высокого уровня (объектно-ориентированное программирование)	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3			Вопросы для проведения зачета
	ОПК-7.2.2 умеет разрабатывать и реализовывать на языке высокого уровня алгоритмы решения типовых профессиональных задач	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3			Вопросы для проведения зачета

СРС – самостоятельная работа студентов;

КР – курсовая работа;

КП – курсовой проект.

2.2. Показатели уровней сформированности компетенций на этапах их формирования, описание шкал оценивания

2.2.1. Показатели уровней сформированности компетенций на этапах их формирования

Результатом освоения дисциплины «Программирование на языке C++» является установление одного из уровней сформированности компетенций: высокий, повышенный, базовый, низкий.

Таблица 3

Уровень	Универсальные компетенции	Общепрофессиональные/ профессиональные компетенции
Высокий (оценка «отлично», «зачтено»)	Сформированы четкие системные знания и представления по дисциплине. Ответы на вопросы оценочных средств полные и верные. Даны развернутые ответы на дополнительные вопросы. Обучающимся продемонстрирован высокий уровень освоения компетенции	Обучающимся усвоена взаимосвязь основных понятий дисциплины, в том числе для решения профессиональных задач. Ответы на вопросы оценочных средств самостоятельны, исчерпывающие, содержание вопроса/задания оценочного средства раскрыто полно, профессионально, грамотно. Даны ответы на дополнительные вопросы. Обучающимся продемонстрирован высокий уровень освоения компетенции
Повышенный (оценка «хорошо», «зачтено»)	Знания и представления по дисциплине сформированы на повышенном уровне. В ответах на вопросы/задания оценочных средств изложено понимание вопроса, дано достаточно подробное описание ответа, приведены и раскрыты в тезисной форме основные понятия. Ответ отражает полное знание материала, а также наличие, с незначительными пробелами, умений и навыков по изучаемой дисциплине. Допустимы единичные негрубые ошибки. Обучающимся продемонстрирован повышенный уровень освоения компетенции	Сформированы в целом системные знания и представления по дисциплине. Ответы на вопросы оценочных средств полные, грамотные. Продемонстрирован повышенный уровень владения практическими умениями и навыками. Допустимы единичные негрубые ошибки по ходу ответа, в применении умений и навыков
Базовый (оценка «удовлетворительно», «зачтено»)	Ответ отражает теоретические знания основного материала дисциплины в объеме, необходимом для дальнейшего освоения ОПОП. Обучающийся допускает неточности в ответе, но обладает необходимыми знаниями для их	Обучающийся владеет знаниями основного материал на базовом уровне. Ответы на вопросы оценочных средств неполные, допущены существенные ошибки. Продемонстрирован базовый уровень владения

Уровень	Универсальные компетенции	Общепрофессиональные/ профессиональные компетенции
	устранения. Обучающимся продемонстрирован базовый уровень освоения компетенции	практическими умениями и навыками, соответствующий минимально необходимому уровню для решения профессиональных задач
Низкий (оценка «неудовлетворительно», «не зачтено»)	Демонстрирует полное отсутствие теоретических знаний материала дисциплины, отсутствие практических умений и навыков	

Показатели уровней сформированности компетенций могут быть изменены, дополнены и адаптированы к конкретной рабочей программе дисциплины.

2.2.2. Описание шкал оценивания

В ФГБОУ ВО «ДГТУ» внедрена модульно-рейтинговая система оценки учебной деятельности студентов. В соответствии с этой системой применяются пятибалльная, двадцатибалльная и стобальная шкалы знаний, умений, навыков.

Шкалы оценивания			Критерии оценивания
пятибалльная	двадцатибалльная	стобальная	
«Отлично» - 5 баллов	«Отлично» - 18-20 баллов	«Отлично» - 85 – 100 баллов	Показывает высокий уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> - продемонстрирует глубокое и прочное усвоение материала; - исчерпывающе, четко, последовательно, грамотно и логически стройно излагает теоретический материал; - правильно формирует определения; - демонстрирует умения самостоятельной работы с нормативно-правовой литературой; - умеет делать выводы по излагаемому материалу.
«Хорошо» - 4 баллов	«Хорошо» - 15 - 17 баллов	«Хорошо» - 70 - 84 баллов	Показывает достаточный уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> - демонстрирует достаточно полное знание материала, основных теоретических положений; - достаточно последовательно, грамотно логически стройно излагает материал; - демонстрирует умения ориентироваться в нормальной литературе; - умеет делать достаточно обоснованные выводы по излагаемому материалу.
«Удовлетворительно» - 3 баллов	«Удовлетворительно» - 12 - 14 баллов	«Удовлетворительно» - 56 – 69 баллов	Показывает пороговый уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> - демонстрирует общее знание изучаемого материала; - испытывает серьезные затруднения при ответах на дополнительные вопросы; - знает основную рекомендуемую литературу; - умеет строить ответ в соответствии со структурой излагаемого материала.
«Неудовлетворительно» - 2 баллов	«Неудовлетворительно» - 1-11 баллов	«Неудовлетворительно» - 1-55 баллов	Ставится в случае: <ul style="list-style-type: none"> - незнания значительной части программного материала; - не владения понятийным аппаратом дисциплины; - допущения существенных ошибок при изложении учебного материала; - неумение строить ответ в соответствии со структурой излагаемого вопроса; - неумение делать выводы по излагаемому материалу.

3. Типовые контрольные задания, иные материалы и методические рекомендации, необходимые для оценки сформированности компетенций в процессе освоения ОПОП
3 семестр

3.1. Задания и вопросы для входного контроля

1. Что такое алгоритм? Перечислите основные свойства алгоритма.
2. Назовите основные типы данных, доступные в языках программирования высокого уровня. Приведите примеры.
3. В чём отличие компиляции от интерпретации? Приведите примеры компилируемых и интерпретируемых языков.
4. Опишите назначение и основные элементы структур данных: стек, очередь, список.
5. Что такое функция? Для чего используются параметры и возвращаемое значение?
6. Объясните понятие рекурсии. Приведите пример рекурсивной функции.
7. Какие конструкции языка позволяют реализовать ветвление в программе? Приведите пример условного оператора.
8. Что такое массив? Как осуществляется доступ к элементам массива?
9. Дайте определение объектно-ориентированного программирования. Перечислите основные принципы ООП.
10. Какие структуры данных из стандартной библиотеки Python вы знаете? Для чего используется словарь (dict)?

3.2. Оценочные средства и критерии сформированности компетенций

3.2.1. Аттестационная контрольная работа №1

1. Расскажите об истории создания языка C++. Какие основные особенности отличают C++ от других языков?
2. Перечислите этапы трансляции программы на C++ (препроцессирование, компиляция, ассемблирование, линковка). Какие задачи решаются на каждом этапе?
3. Назовите базовые типы данных в C++. В чём отличие типов int, float, double, char?
4. Что такое переменная? Как объявить и инициализировать переменную в C++?
5. Опишите арифметические, логические операторы и операторы сравнения в C++. Приведите примеры.
6. Какие управляющие конструкции используются для ветвления? Напишите фрагмент кода с использованием if-else и switch.
7. Циклы в C++: for, while, do-while. В каких случаях предпочтительнее использовать каждый из них?
8. Что такое функция? Опишите синтаксис объявления и определения функции. Что такое прототип функции?
9. Какие способы передачи параметров в функцию существуют в C++? Объясните передачу по значению, по ссылке и по указателю.
10. Что такое перегрузка функций? Приведите пример двух перегруженных функций с одинаковым именем.

3.2.2. Аттестационная контрольная работа №2

1. Что такое указатель? Как объявить указатель, получить адрес переменной и значение по адресу (операторы & и *)?
2. В чём отличие ссылки от указателя? Приведите примеры использования ссылок.
3. Как объявить одномерный и двумерный массив в C++? Как осуществляется доступ к элементам массива?
4. Какие существуют способы работы со строками в C++? Опишите строки в стиле C (массивы char) и класс std::string.

5. Для чего используется динамическое выделение памяти? Опишите операторы new и delete, а также new[] и delete[].
6. Что такое утечка памяти? Как избежать утечек при работе с динамической памятью?
7. Что такое структура (struct)? Как объявить структуру и обратиться к её полям?
8. Чем структура отличается от объединения (union)? Приведите пример использования объединения.
9. Что такое класс? В чём отличие класса от структуры в C++ по умолчанию?
10. Опишите конструкторы и деструкторы. Какие виды конструкторов существуют (по умолчанию, копирования, параметризованный)? Сформулируйте «правило трёх».

3.2.3. Аттестационная контрольная работа №3

1. Что такое наследование? Какие виды наследования поддерживаются в C++ (public, protected, private)?
2. Объясните понятие множественного наследования. Какие проблемы могут возникнуть при его использовании?
3. Что такое полиморфизм? Как достигается полиморфизм во время выполнения с помощью виртуальных функций?
4. Что такое виртуальный деструктор? Для чего он нужен?
5. Что такое абстрактный класс? Как объявить чисто виртуальную функцию?
6. Перегрузка операторов: какие операторы можно перегружать? Приведите пример перегрузки оператора + для пользовательского класса.
7. Что такое дружественная функция (friend)? Для каких целей она используется?
8. Шаблоны функций: синтаксис и пример использования. В чём преимущество шаблонов перед перегрузкой?
9. Шаблоны классов: как объявить и использовать шаблонный класс? Приведите пример.
10. Обработка исключений: конструкция try, catch, throw. Какие стандартные исключения существуют в C++?

3.2.4. Список вопросов к зачету

1. История создания языка C++. Особенности и области применения.
2. Этапы трансляции программы на C++: препроцессирование, компиляция, ассемблирование, линковка.
3. Базовые типы данных в C++. Модификаторы signed, unsigned, short, long.
4. Переменные: объявление, инициализация, область видимости.
5. Арифметические, логические операторы и операторы сравнения. Приоритет операций.
6. Условные операторы: if, else if, else, вложенные условия. Тернарный оператор.
7. Оператор выбора switch. Особенности использования break.
8. Циклы: for, while, do-while. Операторы break и continue.
9. Функции: определение, объявление (прототип), вызов. Возвращаемое значение.
10. Параметры функций: передача по значению, по ссылке, по указателю.
11. Перегрузка функций. Правила разрешения перегрузки.
12. Указатели: объявление, инициализация, арифметика указателей.
13. Ссылки: отличие от указателей, использование в качестве параметров и возвращаемых значений.
14. Массивы: одномерные и многомерные. Инициализация и доступ к элементам.
15. Строки в стиле C (массивы символов). Функции для работы со строками (strlen, strcpy и др.).
16. Класс std::string. Основные методы и операции.
17. Динамическое выделение памяти: операторы new, delete, new[], delete[].
18. Утечки памяти и способы их предотвращения.
19. Структуры (struct). Объявление, доступ к полям, вложенные структуры.
20. Объединения (union). Особенности использования.

21. Основы ООП: классы и объекты. Инкапсуляция, модификаторы доступа.
22. Конструкторы: виды (по умолчанию, копирования, параметризованный). Список инициализации.
23. Деструкторы. Правило трёх (Rule of Three).
24. Наследование: базовый и производный классы. Виды наследования (public, protected, private).
25. Множественное наследование. Проблема ромбовидного наследования и виртуальные базовые классы.
26. Виртуальные функции и полиморфизм. Таблица виртуальных функций (vtable).
27. Виртуальные деструкторы. Абстрактные классы и чисто виртуальные функции.
28. Перегрузка операторов. Ограничения и примеры.
29. Шаблоны функций и классов. Специализация шаблонов.
30. Обработка исключений: синтаксис try-catch-throw, стандартные классы исключений.

3.3. Вопросы по остаточным знаниям

1. Дайте определение алгоритма. Перечислите основные свойства алгоритма.
2. Что такое переменная? Какие базовые типы данных вам известны?
3. В чём различие между компилируемыми и интерпретируемыми языками программирования? Приведите примеры.
4. Опишите структуру условного оператора (if-else) на любом известном вам языке программирования.
5. Что такое цикл? Назовите виды циклов и приведите пример задачи, решаемой с их помощью.
6. Дайте определение функции. Для чего используются параметры и возвращаемое значение?
7. Что такое рекурсия? Приведите пример рекурсивной функции.
8. Какие структуры данных (контейнеры) вы знаете? Опишите назначение списка, словаря, множества.
9. Назовите основные принципы объектно-ориентированного программирования (ООП).
10. Что такое сложность алгоритма? Чем характеризуются временная и ёмкостная сложности?

4 семестр

3.4. Задания и вопросы для входного контроля

1. Перечислите основные принципы объектно-ориентированного программирования и кратко охарактеризуйте каждый.
2. В чём отличие передачи параметров по ссылке от передачи по указателю?
3. Что такое виртуальная функция? Для чего используется виртуальный деструктор?
4. Опишите конструктор копирования и оператор присваивания. В каких ситуациях они вызываются?
5. Что такое шаблон класса? Приведите пример шаблонного класса для хранения пары значений.
6. Как обрабатываются исключения в C++? Напишите фрагмент кода с try-catch.
7. Какие контейнеры стандартной библиотеки шаблонов (STL) вы знаете? Для чего используется std::vector?
8. Что такое итератор? Какие виды итераторов существуют?
9. Объясните разницу между new/delete и malloc/free.
10. Что такое перегрузка операторов? Можно ли перегрузить оператор ::?

3.5. Оценочные средства и критерии сформированности компетенций

3.5.1. Аттестационная контрольная работа №1

1. Какие стандартные исключения определены в C++? Как создать собственный класс исключения?
2. Реализуйте функцию, которая делит два числа и выбрасывает исключение при делении на ноль.
3. Какие классы и функции используются для файлового ввода-вывода в C++? (ifstream, ofstream, fstream).
4. Напишите программу, которая записывает в текстовый файл строку, а затем считывает её обратно.
5. Что такое умные указатели? Перечислите виды (unique_ptr, shared_ptr, weak_ptr). Для чего нужен каждый?
6. В чём преимущество использования unique_ptr перед обычным указателем?
7. Что такое лямбда-выражение? Опишите синтаксис и приведите пример использования.
8. Как лямбда-выражение может захватывать переменные из внешней области видимости (по значению, по ссылке)?
9. Что такое move-семантика? В чём отличие перемещения от копирования?
10. Что такое std::move? Приведите пример, когда перемещение эффективнее копирования.

3.5.2. Аттестационная контрольная работа №2

1. Сравните std::vector и std::list. В каких случаях предпочтительнее использовать каждый из них?
2. Как получить итератор на начало и конец контейнера? Для чего используются begin() и end()?
3. Что такое ассоциативные контейнеры? Опишите std::map и std::set. В чём их отличие?
4. Напишите фрагмент кода, который с помощью std::map подсчитывает частоту слов в строке.
5. Какие алгоритмы STL для сортировки вы знаете? Как отсортировать вектор по убыванию?
6. Как найти элемент в векторе с помощью алгоритма std::find? Что возвращается, если элемент не найден?
7. Что такое многопоточность? Как создать поток с помощью std::thread?
8. Какие проблемы синхронизации возникают при работе нескольких потоков с общими данными?
9. Что такое мьютекс (std::mutex)? Как с его помощью защитить критическую секцию?
10. Что такое thread-safe контейнер? Какие стандартные контейнеры являются потокобезопасными?

3.5.3. Аттестационная контрольная работа №3

1. Что такое асинхронное программирование? Как в C++ реализована асинхронность с помощью std::async?
2. Чем отличается std::async от создания явного потока?
3. Что такое шаблонное метапрограммирование? Приведите пример вычисления факториала на этапе компиляции с использованием шаблонов.
4. Для чего нужны пользовательские аллокаторы? Когда может потребоваться написать собственный аллокатор?
5. Какие системные вызовы используются для работы с файлами и процессами в кроссплатформенном коде? Как обеспечить переносимость?
6. Что такое профилирование программы? Какие инструменты можно использовать для профилирования C++ кода (например, Valgrind, gprof)?
7. Какие типичные уязвимости могут возникать в C++ программах? (переполнение буфера, использование после освобождения и т.д.)

8. Как можно защититься от переполнения буфера при работе с C-строками?
9. Что такое сетевое программирование? Какие классы или библиотеки используются в C++ для создания клиент-серверных приложений (например, сокет, Boost.Asio)?
10. Напишите простейший пример TCP-клиента, который подключается к серверу и отправляет сообщение (псевдокод или реальный код с использованием сокетов).

3.6. Список вопросов к зачету

1. Обработка исключений: синтаксис, типы исключений, создание собственных исключений.
2. Файловый ввод-вывод: работа с текстовыми и бинарными файлами, режимы открытия.
3. Умные указатели: `unique_ptr`, `shared_ptr`, `weak_ptr`. Назначение и примеры использования.
4. Пользовательские `deleter` для умных указателей.
5. Лямбда-выражения: синтаксис, захват переменных, обобщённые лямбды (C++14).
6. Move-семантика: `rvalue`-ссылки, семантика перемещения, `std::move`.
7. Контейнеры STL: последовательные контейнеры (`vector`, `deque`, `list`, `forward_list`).
8. Ассоциативные контейнеры (`set`, `map`, `multiset`, `multimap`) и неупорядоченные контейнеры (`unordered_set`, `unordered_map`).
9. Адаптеры контейнеров (`stack`, `queue`, `priority_queue`).
10. Итераторы: категории, операции, инвалидация итераторов.
11. Алгоритмы STL: немодифицирующие и модифицирующие алгоритмы, алгоритмы сортировки, поиска, сведения.
12. Функциональные объекты (функторы) и `std::function`.
13. Многопоточность: создание потоков (`std::thread`), `join` и `detach`.
14. Синхронизация: мьютексы (`std::mutex`, `std::recursive_mutex`), блокировки (`lock_guard`, `unique_lock`).
15. Условные переменные (`std::condition_variable`).
16. Атомарные операции (`std::atomic`).
17. Асинхронное программирование: `std::async`, `std::future`, `std::promise`.
18. Параллельные алгоритмы STL (C++17).
19. Шаблонное метапрограммирование: основы, рекурсивные шаблоны, `static_assert`.
20. Типажи (traits) и `std::enable_if`, концепты (C++20).
21. Пользовательские аллокаторы: назначение, интерфейс, пример.
22. Профилирование и оптимизация кода: инструменты (Valgrind, perf), ключи компиляции.
23. Измерение времени выполнения: `std::chrono`.
24. Кроссплатформенная разработка: препроцессорные макросы, условная компиляция.
25. Системные вызовы и их обёртки в C++ (работа с файловой системой, процессы).
26. Сетевое программирование: сокет (Berkeley sockets), работа с TCP/UDP.
27. Библиотека Boost.Asio: основы асинхронного ввода-вывода.
28. Безопасность C++ приложений: переполнение буфера, утечки памяти, использование после освобождения, двойное освобождение.
29. Инструменты статического и динамического анализа кода (Clang Static Analyzer, AddressSanitizer, Valgrind Memcheck).
30. Стандарты C++: основные нововведения C++11/14/17/20, влияние на безопасность и производительность.

3.7. Вопросы по остаточным знаниям

1. Перечислите основные этапы трансляции программы на C++ (препроцессирование, компиляция, ассемблирование, линковка).
2. Чем передача параметров по ссылке отличается от передачи по указателю?
3. Что такое динамическое выделение памяти? Опишите операторы `new` и `delete`.

4. Объясните понятие класса и объекта. В чём отличие класса от структуры в C++?
5. Для чего используются конструкторы и деструкторы? Назовите виды конструкторов.
6. Что такое наследование? Какие типы наследования поддерживаются в C++?
7. Опишите механизм виртуальных функций. Для чего нужен виртуальный деструктор?
8. Что такое абстрактный класс и чисто виртуальная функция?
9. Приведите пример перегрузки оператора для пользовательского класса.
10. Какие контейнеры из стандартной библиотеки шаблонов (STL) вы знаете? Для чего используется `std::vector`?

Зачеты и экзамены могут быть проведены в письменной форме, а также в письменной форме с устным дополнением ответа. Зачеты служат формой проверки качества выполнения студентами лабораторных работ, усвоения семестрового учебного материала по дисциплине (модулю), практических и семинарских занятий (при отсутствии экзамена по дисциплине).

По итогам зачета, соответствии с модульно – рейтинговой системой университета, выставляются баллы с последующим переходом по шкале баллы – оценки за зачет, выставляемый как по наименованию «зачтено», «не зачтено», так и дифференцированно т.е. с выставлением отметки по схеме – «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно», определяемое решением Ученого совета университета и прописываемого в учебном плане.

Экзамен по дисциплине (модулю) служит для оценки работы студента в течении семестра (года, всего срока обучения и др.) и призван выявить уровень, качество и систематичность полученных им теоретических и практических знаний, приобретения навыков самостоятельной работы, развития творческого мышления, умения синтезировать полученные знания и применять их в решении практических задач. По итогам экзамена, в соответствии с модульно – рейтинговой системой университета выставляются баллы, с последующим переходом по шкале оценок на оценки: «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно», свидетельствующие о приобретенных компетенциях или их отсутствии.

Критерии оценки уровня сформированности компетенций по результатам проведения зачета:

- оценка «зачтено»: обучающийся демонстрирует всестороннее, систематическое и глубокое знание материала, свободно выполняет задания, предусмотренные программой дисциплины, усвоивший основную и дополнительную литературу. Обучающийся выполняет задания, предусмотренные программой дисциплины, на уровне не ниже базового;

- оценка «не зачтено»: обучающийся демонстрирует незнание материала, не выполняет задания, предусмотренные программой дисциплины. Обучающийся не выполняет задания, предусмотренные программой дисциплины, на уровне ниже базового. Дальнейшее освоение ОПОП не возможно без дополнительного изучения материала и подготовки к зачету.

Критерии оценки уровня сформированности компетенций по результатам проведения дифференцированного зачёта (зачета с оценкой) / экзамена:

- оценка «**отлично**»: обучающийся дал полный, развернутый ответ на поставленный вопрос, проявил совокупность осознанных знаний об объекте, доказательно раскрыл основные положения темы. В ответе прослеживается четкая структура, логическая последовательность, отражающая сущность раскрываемых понятий, явлений. Обучающийся подкрепляет теоретический ответ практическими примерами. Ответ сформулирован научным языком, обоснована авторская позиция обучающегося. Могут быть допущены недочеты в определении понятий, исправленные студентом самостоятельно в процессе ответа или с помощью «наводящих» вопросов преподавателя. Обучающимся продемонстрирован высокий уровень владения компетенцией(-ями);

- оценка «**хорошо**»: обучающимся дан полный, развернутый ответ на поставленный вопрос, проявлено умение выделять существенные и несущественные признаки, причинно-следственные связи. Ответ четко структурирован, логичен, но есть недочеты в формулировании понятий, решении задач. При ответах на дополнительные вопросы допущены незначительные ошибки. Обучающимся продемонстрирован повышенный уровень владения компетенцией(-ями);

- оценка «**удовлетворительно**»: обучающимся дан неполный ответ на вопрос, логика и последовательность изложения имеют существенные нарушения. Допущены грубые ошибки при определении сущности раскрываемых понятий, явлений, нарушена логика ответа, не сделаны выводы. Речевое оформление требует коррекции. Обучающийся испытывает затруднение при ответе на дополнительные вопросы. Обучающимся продемонстрирован базовый уровень владения компетенцией(-ями);

- оценки «**неудовлетворительно**»: обучающийся испытывает значительные трудности в ответе на вопрос, допускает существенные ошибки, не владеет терминологией, не знает основных понятий, не может ответить на «наводящие» вопросы преподавателя. Обучающимся продемонстрирован низкий уровень владения компетенцией(-ями).