

Документ подписан простой электронной подписью.
Информация о владельце:

ФИО: Кандрашина Елена Александровна

Должность: И.о. ректора ФГАОУ ВО «Самарский государственный экономический университет»

Дата подписания: 29.10.2025 14:29:09

Уникальный программный ключ:

2db64eb9605ce27edd3b8e8fdd32c70e0674ddd2

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Самарский государственный экономический университет»

Институт Институт национальной и мировой экономики

Кафедра Прикладной информатики

УТВЕРЖДЕНО

Ученым советом Университета
(протокол № 10 от 22 мая 2025 г.)

РАБОЧАЯ ПРОГРАММА

Наименование дисциплины	Б1.О.27 Современные технологии и языки программирования
Основная профессиональная образовательная программа	01.03.05 Статистика программа Информационные системы на финансовых рынках
Квалификация (степень) выпускника	Бакалавр

Целью изучения дисциплины является формирование результатов обучения, обеспечивающих достижение планируемых результатов освоения образовательной программы.

1. Место дисциплины в структуре ОП

Дисциплина Современные технологии и языки программирования входит в обязательную часть блока Б1. Дисциплины (модули)

Предшествующие дисциплины по связям компетенций: Технологии цифровой экономики, Основы алгоритмизации и программирования, Основы информационной безопасности

Последующие дисциплины по связям компетенций: Пакеты прикладных статистических программ, Кибербезопасность

2. Планируемые результаты обучения по дисциплине, обеспечивающие достижение планируемых результатов обучения по программе

Изучение дисциплины Современные технологии и языки программирования в образовательной программе направлено на формирование у обучающихся следующих компетенций:

Общепрофессиональные компетенции (ОПК):

ОПК-4 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

Планируемые результаты обучения по программе	Планируемые результаты обучения по дисциплине		
ОПК-4	ОПК-4.1: Знать:	ОПК-4.2: Уметь:	ОПК-4.3: Владеть (иметь навыки):
	основные методы, способы и средства получения, хранения, переработки информации с помощью современных информационных технологий; принципы работы с информацией в глобальных компьютерных сетях	осуществлять поиск, накопление и обработку информации, в т.ч. с использованием компьютера и глобальных информационных сетей; применять на практике ключевые методы сбора и обработки информации из различных источников, в том числе сети Интернет; работать с компьютером; эффективно управлять информацией с помощью информационных и сквозных технологий	основными методами, способами и средствами получения, хранения, переработки информации; навыками использования современных технических средств и информационных технологий для решения профессиональных задач; цифровыми инструментами коммуникации, инновационными методами обработки больших данных

ОПКЭ-6 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

Планируемые результаты обучения по программе	Планируемые результаты обучения по дисциплине		
ОПКЭ-6	ОПКЭ-6.1: Знать:	ОПКЭ-6.2: Уметь:	ОПКЭ-6.3: Владеть (иметь навыки):
	основные методы, способы и средства получения, хранения, переработки информации с помощью современных информационных технологий; принципы работы с информацией в глобальных компьютерных сетях	осуществлять поиск, накопление и обработку информации, в т.ч. с использованием компьютера и глобальных информационных сетей; применять на практике ключевые методы сбора и обработки информации из различных источников, в том числе сети Интернет; работать с компьютером; эффективно управлять информацией с помощью информационных и сквозных технологий	основными методами, способами и средствами получения, хранения, переработки информации; навыками использования современных технических средств и информационных технологий для решения профессиональных задач; цифровыми инструментами коммуникации, инновационными методами обработки больших данных

3. Объем и виды учебной работы

Учебным планом предусматриваются следующие виды учебной работы по дисциплине:

Очная форма обучения

Виды учебной работы	Всего час/ з.е.
	Сем 5
Контактная работа, в том числе:	54.15/1.5
Занятия лекционного типа	18/0.5
Занятия семинарского типа	36/1
Индивидуальная контактная работа (ИКР)	0.15/0
Самостоятельная работа:	35.85/1
Промежуточная аттестация	18/0.5
Вид промежуточной аттестации:	
Зачет	Зач
Общая трудоемкость (объем части образовательной программы): Часы	108
Зачетные единицы	3

4. Содержание дисциплины

4.1. Разделы, темы дисциплины и виды занятий:

Тематический план дисциплины Современные технологии и языки программирования представлен в таблице.

Разделы, темы дисциплины и виды занятий Очная форма обучения

№ п/п	Наименование темы (раздела) дисциплины	Контактная работа				Самостоятельная работа	Планируемые результаты обучения в соотношении с результатами обучения по образовательной программе
		Лекции	Занятия семинарского типа	ИКР	ГКР		
			Практич. занятия				
1.	Введение в программирование на языке C++. Основные элементы языка программирования. Базовые типы данных.	8	18	0.075		15	ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПКЭ-6.1, ОПКЭ-6.2, ОПКЭ-6.3
2.	Структурное программирование. Динамические структуры данных. Основы объектно-ориентированного программирования. Стандартная библиотека шаблонов.	10	18	0.075		20.85	ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПКЭ-6.1, ОПКЭ-6.2, ОПКЭ-6.3
	Контроль	18					
	Итого	18	36	0.15		35.85	

4.2 Содержание разделов и тем

4.2.1 Контактная работа

Тематика занятий лекционного типа

№п/п	Наименование темы (раздела) дисциплины	Вид занятия лекционного типа*	Тематика занятия лекционного типа
1.	Введение в программирование на языке C++. Основные элементы языка программирования. Базовые типы данных.	лекция	Введение в C++. Философия C++. Компиляция и линковка. Первая программа Hello World. Инструменты разработчика.
		лекция	Базовые типы данных и операции. Понятие переменной и идентификатора. Фундаментальные (встроенные) типы данных. Объявление и инициализация переменных. Базовые операции
		лекция	Основные элементы языка: управляющие конструкции. Структура программы. Условные конструкции. Циклы. Пространства имен.

		лекция	Основные элементы языка: составные типы данных и функции. Массивы. Указатели. Строки. Функции.
2.	Структурное программирование. Динамические структуры данных. Основы объектно-ориентированного программирования. Стандартная библиотека шаблонов.	лекция	Структурное программирование. Следование, ветвление, повторение. Принцип модульности. Функции как инструмент модульности. Области видимости и время жизни переменных.
		лекция	Управление памятью. Стек и Куча. Операторы new и delete. Опасности ручного управления памятью. «Умные указатели». Создание простейших динамических структур данных.
		лекция	Динамические структуры данных. Односвязный список. Двусвязный список. Абстрактные типы данных: стек, очередь. Инвариант класса.
		лекция	Основы ООП. От структур к классам. Инкапсуляция. Конструкторы и деструкторы. Введение в перегрузку функций и операторов.
		лекция	Стандартная библиотека шаблонов. Философия STL. Контейнеры. Итераторы. Алгоритмы.

*лекции и иные учебные занятия, предусматривающие преимущественную передачу учебной информации педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях, обучающимся

Тематика занятий семинарского типа

№п/п	Наименование темы (раздела) дисциплины	Вид занятия семинарского типа**	Тематика занятия семинарского типа
1.	Введение в программирование на языке C++. Основные элементы языка программирования. Базовые типы данных.	практическое занятие	Первое знакомство со средой разработки. Создание проекта Hello World.
		практическое занятие	Структура простой программы и базовый ввод-вывод. Создание приложения, подсчитывающего возраст пользователя.
		практическое занятие	Объявление переменных и арифметические операции. Создание простейшего приложения калькулятор.
		практическое занятие	Работа с условным оператором if-else. Создание приложения для сравнения чисел и нахождения минимума-максимума.
		практическое занятие	Работа с оператором выбора switch-case. Создание более сложного приложения калькулятор с обработчиком ошибок.
		практическое занятие	Работа с циклами while и do-while. Создание приложений, работающих с циклами.
		практическое занятие	Целочисленные типы данных и модификаторы. Исследование

			типов int, short, long, long long.
		практическое занятие	Числа с плавающей точкой и логический тип. Подсчет площади круга и длины окружности. Сравнение чисел.
		практическое занятие	Символьный тип. Неявное преобразование типов. Работа с ASCII кодами. Арифметические операции с символами.
2.	Структурное программирование. Динамические структуры данных. Основы объектно-ориентированного программирования. Стандартная библиотека шаблонов.	практическое занятие	Разработка многофайловых проектов. Создание собственной библиотеки функций.
		практическое занятие	Работа с массивами. Поиск, сортировка, обработка.
		практическое занятие	Работа с динамической памятью. Односвязный список.
		практическое занятие	Реализация стека и очереди на основе списков.
		практическое занятие	Создание классов. Инкапсуляция и методы доступа
		практическое занятие	Работа с классами. Наследование и полиморфизм.
		практическое занятие	Контейнеры STL: vector, list, map
		практическое занятие	Итераторы и алгоритмы STL.
		практическое занятие	Практическое применение STL для решения комплексной задачи.

** семинары, практические занятия, практикумы, лабораторные работы, коллоквиумы и иные аналогичные занятия

Иная контактная работа

При проведении учебных занятий СГЭУ обеспечивает развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств (включая при необходимости проведение интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализ ситуаций и имитационных моделей, преподавание дисциплин (модулей) в форме курсов, составленных на основе результатов научных исследований, проводимых организацией, в том числе с учетом региональных особенностей профессиональной деятельности выпускников и потребностей работодателей).

Формы и методы проведения иной контактной работы приведены в Методических указаниях по основной профессиональной образовательной программе.

4.2.2 Самостоятельная работа

№п/п	Наименование темы (раздела) дисциплины	Вид самостоятельной работы ***
1.	Введение в программирование на языке C++. Основные элементы языка программирования. Базовые типы данных.	- подготовка доклада - подготовка электронной презентации - тестирование
2.	Структурное программирование. Динамические структуры данных. Основы объектно-ориентированного программирования. Стандартная библиотека шаблонов.	- подготовка доклада - подготовка электронной презентации - тестирование

*** самостоятельная работа в семестре, написание курсовых работ, докладов, выполнение контрольных работ

5. Материально-техническое и учебно-методическое обеспечение дисциплины

5.1 Литература:

Основная литература

1. Гниденко, И. Г. Технологии и методы программирования : учебник для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 248 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560978>

Дополнительная литература

1. Зыков, С. В. Объектно-ориентированное программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 151 с. — (Высшее образование). — ISBN 978-5-534-16941-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561434>
2. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 285 с. — (Высшее образование). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560815>
3. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 150 с. — (Высшее образование). — ISBN 978-5-534-16942-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561744>

Литература для самостоятельного изучения

1. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 432 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-07604-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/436514> (дата обращения: 01.09.2025).
2. Соколова, В. В. Вычислительная техника и информационные технологии. Разработка мобильных приложений : учебное пособие для прикладного бакалавриата / В. В. Соколова. — Москва : Издательство Юрайт, 2019. — 175 с. — (Университеты России). — ISBN 978-5-9916-6525-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/433981> (дата обращения: 01.09.2025).

5.2. Перечень лицензионного программного обеспечения

1. Astra Linux Special Edition «Смоленск», «Орел»; РедОС ; ОС "Альт Рабочая станция" 10; ОС "Альт Образование" 10
2. МойОфис Стандартный 2, МойОфис Образование, Р7-Офис Профессиональный, МойОфис Стандартный 3, МойОфис Профессиональный 3

5.3 Современные профессиональные базы данных, к которым обеспечивается доступ обучающихся

1. Профессиональная база данных «Информационные системы Министерства экономического развития Российской Федерации в сети Интернет» (Портал «Официальная Россия» - <http://www.gov.ru/>)
2. Государственная система правовой информации «Официальный интернет-портал правовой информации» (<http://pravo.gov.ru/>)
3. Профессиональная база данных «Финансово-экономические показатели Российской Федерации» (Официальный сайт Министерства финансов РФ - <https://www.minfin.ru/ru/>)
4. Профессиональная база данных «Официальная статистика» (Официальный сайт

5.4. Информационно-справочные системы, к которым обеспечивается доступ обучающихся

1. Справочно-правовая система «Консультант Плюс»
2. Справочно-правовая система «ГАРАНТ-Максимум»

5.5. Специальные помещения

Учебные аудитории для проведения занятий лекционного типа	Комплекты ученической мебели Мультимедийный проектор Доска Экран
Учебные аудитории для проведения практических занятий (занятий семинарского типа)	Комплекты ученической мебели Мультимедийный проектор Доска Экран Компьютеры с выходом в сеть «Интернет» и ЭИОС СГЭУ
Учебные аудитории для групповых и индивидуальных консультаций	Комплекты ученической мебели Мультимедийный проектор Доска Экран Компьютеры с выходом в сеть «Интернет» и ЭИОС СГЭУ
Учебные аудитории для текущего контроля и промежуточной аттестации	Комплекты ученической мебели Мультимедийный проектор Доска Экран Компьютеры с выходом в сеть «Интернет» и ЭИОС СГЭУ
Помещения для самостоятельной работы	Комплекты ученической мебели Мультимедийный проектор Доска Экран Компьютеры с выходом в сеть «Интернет» и ЭИОС СГЭУ
Помещения для хранения и профилактического обслуживания оборудования	Комплекты специализированной мебели для хранения оборудования

6. Фонд оценочных средств по дисциплине Современные технологии и языки программирования:

6.1. Контрольные мероприятия по дисциплине

Вид контроля	Форма контроля	Отметить нужное знаком « + »
Текущий контроль	Оценка докладов	+
	Устный/письменный опрос	—
	Тестирование	+
	Практические задачи	+
Промежуточный контроль	Зачет	+

Порядок проведения мероприятий текущего и промежуточного контроля определяется Методическими указаниями по основной профессиональной образовательной программе высшего образования; Положением о балльно-рейтинговой системе оценки успеваемости обучающихся по основным образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры в федеральном государственном автономном образовательном учреждении высшего образования «Самарский государственный экономический университет».

6.2. Планируемые результаты обучения по дисциплине, обеспечивающие достижение планируемых результатов обучения по программе

Общепрофессиональные компетенции (ОПК):

ОПК-4 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

Планируемые результаты обучения по программе	Планируемые результаты обучения по дисциплине		
	ОПК-4.1: Знать:	ОПК-4.2: Уметь:	ОПК-4.3: Владеть (иметь навыки):
	основные методы, способы и средства получения, хранения, переработки информации с помощью современных информационных технологий; принципы работы с информацией в глобальных компьютерных сетях	осуществлять поиск, накопление и обработку информации, в т.ч. с использованием компьютера и глобальных информационных сетей; применять на практике ключевые методы сбора и обработки информации из различных источников, в том числе сети Интернет; работать с компьютером; эффективно управлять информацией с помощью информационных и сквозных технологий	основными методами, способами и средствами получения, хранения, переработки информации; навыками использования современных технических средств и информационных технологий для решения профессиональных задач; цифровыми инструментами коммуникации, инновационными методами обработки больших данных
Пороговый	Знание теоретических основ и базовых понятий	Действие по строгому алгоритму, осуществление простого поиска информации, работа с файлами	Навыки простого поиска информации по ключевым словам. Использование офисных пакетов. Использование цифровых инструментов коммуникаций (мессенджеры).
Стандартный (в дополнение к пороговому)	Знание методов и средств получения и переработки информации	Осознанный выбор источников, продвинутый поиск, базовый анализ данных	Использование продвинутых поисковых операторов, работа с электронными библиотеками. Навыки работы в специальном ПО, релевантном специальности.

Повышенный (в дополнение к пороговому, стандартному)	Глубокое понимание методов работы с информацией, автоматизация переработки, анализ и синтез информации	Разработка стратегии поиска, синтез из разнородных источников. Веб-скрапинг, работа с API, написание макросов	Навыки веб-скрапинга. Автоматизация обработки данных с помощью макросов или скриптов. Использование современных технических средств и инновационных методов обработки больших данных.
--	--	---	---

ОПКЭ-6 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

Планируемые результаты обучения по программе	Планируемые результаты обучения по дисциплине		
	ОПКЭ-6.1: Знать:	ОПКЭ-6.2: Уметь:	ОПКЭ-6.3: Владеть (иметь навыки):
	основные методы, способы и средства получения, хранения, переработки информации с помощью современных информационных технологий; принципы работы с информацией в глобальных компьютерных сетях	осуществлять поиск, накопление и обработку информации, в т.ч. с использованием компьютера и глобальных информационных сетей; применять на практике ключевые методы сбора и обработки информации из различных источников, в том числе сети Интернет; работать с компьютером; эффективно управлять информацией с помощью информационных и сквозных технологий	основными методами, способами и средствами получения, хранения, переработки информации; навыками использования современных технических средств и информационных технологий для решения профессиональных задач; цифровыми инструментами коммуникации, инновационными методами обработки больших данных
Пороговый	Основные методы и средства получения, хранения и переработки информации. Принципы работы в глобальных сетях (поисковые системы, браузеры). Базовые функции офисных пакетов	Осуществлять поиск информации в интернете по заданным критериям. Накапливать и структурировать найденную информацию. Работать с основными офисными приложениями	Навыками работы с ПК и основными прикладными программами. Навыками эффективного поиска и первичного отбора информации в сети Интернет
Стандартный (в дополнение к пороговому)	Принципы работы с базами данных, CRM- и ERP-системами. Основы облачных технологий (Google Workspace, MS Office 365) и их применение в бизнесе. Методы критической оценки достоверности	Эффективно управлять информацией с помощью современных технологий (облачные диски, совместные документы). Применять специализированные отраслевые информационные	Навыками использования профессиональных информационных систем и онлайн-сервисов (например, порталов госстатистики, финансовых платформ like Bloomberg/Reuters). Навыками цифровой

	интернет-источников	системы и сервисы для сбора и анализа экономических данных	коммуникации и коллаборации в профессиональной среде
Повышенный (в дополнение к пороговому и стандартному)	Концепции и инструменты сквозных цифровых технологий (Big Data, AI, IoT, блокчейн) и их применение в экономике и финансах. Передовые методы анализа больших данных	Применять инновационные методы и инструменты (например, Python для веб-скрейпинга, анализа больших массивов данных) для решения комплексных профессиональных задач. Интегрировать информацию из разнородных цифровых источников	Навыками работы с большими данными: сбор, очистка, анализ и визуализация. Цифровыми инструментами для автоматизации рутинных задач и повышения эффективности профессиональной деятельности

6.3. Паспорт оценочных материалов

№ п/п	Наименование темы (раздела) дисциплины	Контролируемые планируемые результаты обучения в соотношении с результатами обучения по программе	Вид контроля/используемые оценочные средства	
			Текущий	Промежуточный
1.	Введение в программирование на языке C++. Основные элементы языка программирования. Базовые типы данных.	ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПКЭ-6.1, ОПКЭ-6.2, ОПКЭ-6.3	Оценка докладов Практические занятия Тестирование	Зачет
2.	Структурное программирование. Динамические структуры данных. Основы объектно-ориентированного программирования. Стандартная библиотека шаблонов.	ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПКЭ-6.1, ОПКЭ-6.2, ОПКЭ-6.3	Оценка докладов Практические занятия Тестирование	Зачет

6.4. Оценочные материалы для текущего контроля

Примерная тематика докладов

Раздел дисциплины	Темы
Введение в программирование на языке C++. Основные элементы языка программирования. Базовые типы данных.	1. История создания и эволюция языка C++: от «C с классами» до современного стандарта. 2. Сравнительный анализ языков C и C++: что общего и в чем ключевые различия. 3. Обзор современных сред разработки (IDE) для C++: Visual Studio, CLion, Qt Creator, VS Code. 4. Магия указателей в C++: зачем они нужны и как не запутаться. 5. Этапы трансляции программы: от исходного кода до исполняемого файла. 6. Best Practices: правила написания чистого и читаемого кода на

	C++. 7. Глубокое погружение в типы с плавающей точкой: точность, ошибки и особенности представления в памяти. 8. Тип bool: реализация, размер в памяти и использование в условиях. 9. Кодировки символов: от ASCII к Unicode и UTF-8. Как C++ работает с разными языками. 10. Квалификаторы const и volatile: для чего они нужны и как помогают компилятору оптимизировать код.
Структурное программирование. Динамические структуры данных. Основы объектно-ориентированного программирования. Стандартная библиотека шаблонов.	11. Рекурсия и Итерация: теория, практика, подводные камни в C++. 12. Жизненный цикл программы в памяти: стек, куча (heap), статические данные. Как это работает в C++. 13. Умные указатели (unique_ptr, shared_ptr, weak_ptr) в современном C++. 14. Асимптотический анализ на практике: сравнительная эффективность операций с массивами, списками, деревьями и хэш-таблицами. 15. Правило пяти (Rule of Five) в C++11 и далее: необходимость, реализация и современные альтернативы. 16. Абстрактные классы и интерфейсы в C++: от чисто виртуальных функций к полной абстракции. 17. Множественное наследование: мощный инструмент или антипаттерн. 18. Внутреннее устройство std::vector: как работает самый популярный контейнер и в чем секрет его эффективности. 19. Лямбда-выражения и функциональные объекты в современном C++: новая степень свободы для алгоритмов STL. 20. Итераторы STL: универсальный доступ к данным. Категории итераторов и их ограничения.

Задания для тестирования/ практические задачи по дисциплине для оценки сформированности компетенций

Контролируемые компетенции: ОПК-4 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности; ОПКЭ-6 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

№ п/п	Задание	Ключ к заданию (Эталонный ответ)
1.	При решении задач профессиональной деятельности программисту будет полезным ключевое идеологическое отличие C++ от C. Что это за отличие? 1. Наличие цикла for 2. Поддержка объектно-ориентированного программирования 3. Использование оператора printf для вывода 4. Возможность объявлять переменные	2
2.	Для решения задач профессиональной деятельности вам нужно корректно объявить переменную в C++. Как это сделать? 1. int 1value; 2. double double; 3. char _symbol; 4. string my variable;	3
3.	Для решения задач профессиональной деятельности вам нужно хранить логический флаг (истина/ложь) в C++. Какой тип данных наиболее подходит для этого? 1. int 2. short 3. bool 4. char	3

4.	При решении задач профессиональной деятельности вам нужно использовать цикл, который гарантированно выполнит свое тело как минимум один раз. Какой? 1. for 2. while 3. do-while 4. foreach	3
5.	В процессе решения задач профессиональной деятельности какая операция для односвязного списка обычно имеет временную сложность $O(1)$? 1. Поиск элемента по значению 2. Вставка элемента в конец списка (без хранения tail) 3. Вставка элемента в начало списка 4. Доступ к элементу по индексу	3
6.	Одной из современных информационных технологий является объектно-ориентированное программирование. Какой принцип позволяет скрыть внутреннюю реализацию объекта от пользователя и предоставить только интерфейс? 1. Наследование 2. Инкапсуляция 3. Полиморфизм 4. Абстракция	2
7.	В процессе решения задач профессиональной деятельности какой контейнер STL предоставляет возможность быстрого произвольного доступа к элементам по индексу (за время $O(1)$)? 1. std::list 2. std::vector 3. std::map 4. std::set	2
8.	В процессе решения задач профессиональной деятельности используется парадигма программирования, которую добавляет C++ к процедурному стилю языка C. Какая именно?	Объектно-ориентированное программирование (ООП)
9.	Одним из ключевых понятий современных информационных технологий является виртуальный деструктор. Каким ключевым словом он объявляется?	Virtual
10.	Какой из контейнеров STL при решении задач профессиональной деятельности реализуется как динамический массив и обеспечивает быстрое произвольное чтение и запись по индексу, но медленные вставку и удаление в середине?	std::vector
11.	Какой базовый тип данных в C++ следует использовать для хранения одного символа при решении задач профессиональной деятельности?	char
12.	Какая структура данных является одним из ключевых понятий современных информационных технологий и лежит в основе контейнера std::map?	Сбалансированное бинарное дерево
13.	Для решения задач профессиональной деятельности вам нужно установить соответствие между типом данных и его описанием. 1. bool; 2. int; 3. double; 4. char; 5. void. А. Целочисленный тип, размер которого зависит от архитектуры процессора; Б. Символьный тип, предназначенный для хранения одного символа. В. Тип для хранения логических значений "истина" (true) или "ложь" (false). Г. Тип с плавающей запятой двойной точности. Д. Специальный тип, означающий отсутствие типа или значения.	1-В 2-А 3-Г 4-Б 5-Д
14.	Для решения задач профессиональной деятельности вам нужно установить соответствие между принципом ООП и его описанием. 1. Инкапсуляция; 2. Наследование; 3. Полиморфизм; 4. Абстракция. А. Возможность одному и тому же интерфейсу	1-Б 2-В 3-А 4-Г

	представлять разные реализации. Б. Соккрытие внутреннего устройства класса и объединение данных и методов в единую сущность. В. Механизм создания нового класса на основе существующего, с возможностью reuse и расширения функциональности. Г. Моделирование только существенных характеристик объекта, игнорируя нерелевантные детали.	
15.	В процессе решения задач профессиональной деятельности программист пишет программу для расчёта физических формул, где критична точность вычислений и производительность. Язык С привлекателен скоростью, но требуется высокая степень модульности и повторного использования кода. Какой парадигмой программирования следует воспользоваться в рамках C++ для этого проекта?	Использовать объектно-ориентированное программирование (ООП). Оно позволяет инкапсулировать сложные формулы в классы, делая код модульным, хорошо организованным и пригодным для повторного использования, сохраняя при этом производительность C/C++.
16.	В процессе решения задач профессиональной деятельности требуется хранить результат проверки условия (например, включен ли свет). Какой наиболее семантически подходящий и экономичный тип данных следует выбрать?	Тип bool. Он семантически корректен (хранит значения true/false) и обычно занимает 1 байт, что экономично.
17.	Для решения задачи профессиональной деятельности программа должна прочитать из файла список данных, но размер файла неизвестен на этапе компиляции. Почему использование статического массива <code>int array[1000];</code> является плохой практикой в данном случае? Что следует использовать вместо этого?	Это плохо, потому что если данных больше 1000, произойдёт переполнение буфера. Если меньше — память будет использована нерационально. Следует использовать динамическую структуру, такую как <code>std::vector</code> , которая автоматически масштабируется.
18.	В классе <code>BankAccount</code> есть поле <code>balance</code> . Почему прямое изменение этого поля извне класса в процессе решения задачи профессиональной деятельности, например, <code>myAccount.balance = -1000;</code> , является опасным? Как это предотвратить?	Это опасно, так как нарушает инварианты класса (баланс не может быть отрицательным, если это не овердрафт). Предотвращение: объявить поле <code>private</code> и предоставить для его изменения публичный метод <code>setBalance()</code> , который включает проверку на валидность нового значения.
19.	Есть иерархия классов: базовый <code>Shape</code> и производные <code>Circle</code> , <code>Square</code> . У базового класса есть виртуальный метод <code>draw()</code> . Объясните, почему в процессе решения задачи профессиональной деятельности следующий код не будет работать корректно, и как его исправить: <code>Shape* shape = new Circle();</code> <code>// ... какой-то код</code> <code>delete shape;</code>	Если деструктор базового класса <code>Shape</code> не является виртуальным, то при удалении объекта через указатель на базовый класс будет вызван только деструктор <code>Shape</code> , а деструктор <code>Circle</code> — нет. Это приведёт к утечке ресурсов, которые мог выделять <code>Circle</code> . Исправление: объявить деструктор базового класса виртуальным: <code>virtual ~Shape() {}</code> .
20.	Имеется вектор <code>std::vector<int> numbers = {5, 2, 8, 1, 9};</code> . Какую однострочную команду STL нужно написать в процессе решения задачи профессиональной деятельности для его сортировки по возрастанию?	<code>std::sort(numbers.begin(), numbers.end());</code>
21.	В процессе решения задачи профессиональной деятельности после цикла, в котором создаются динамические объекты через <code>new</code> , программа начинает потреблять всё больше оперативной памяти, хотя работает с одним и тем же объёмом данных. В чём наиболее вероятная причина и как её устранить?	Причина: утечка памяти (memory leak). Объекты, созданные через <code>new</code> , не были удалены с помощью <code>delete</code> . Устранение: обеспечить парность всех вызовов <code>new</code> и <code>delete</code> . Лучше использовать умные указатели (<code>std::unique_ptr</code> , <code>std::shared_ptr</code>), которые автоматически управляют временем жизни объектов.

6.5. Примерные вопросы к зачету

Контролируемые компетенции: ОПК-4 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности; ОПКЭ-6 - Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

№ п/п	Задание	Ключ к заданию / эталонный ответ
1.	В решении задач профессиональной деятельности на современном этапе развития информационных технологий используется язык C++. В чем заключаются основные исторические и идеологические различия между языками C и C++?	C++ был создан Бьерном Страуструпом как расширение языка C, но он включает в себя идеологически новые парадигмы, главная из которых — объектно-ориентированное программирование (ООП).

	Почему C++ не является просто «дополненным» Си?	С — это процедурный язык. Программа на С представляет собой набор функций, работающих с данными. Данные и функции, их обрабатывающие, логически разделены. C++ — это мультипарадигмальный язык, поддерживающий процедурный, объектно-ориентированный, обобщенный (шаблоны) и функциональный стили. Ключевое отличие — инкапсуляция: объединение данных и методов в единые сущности — классы. Это позволяет моделировать реальные объекты, скрывать детали реализации и строить более сложные и надежные системы. Таким образом, C++ — это не просто "С с классами", а язык с принципиально иным подходом к проектированию программ.
2.	Одним из ключевых понятий современных информационных технологий является токен. Что такое токен (лексическая единица) в C++? Перечислите основные виды токенов.	Токен — это минимальная неделимая единица языка, имеющая самостоятельный смысл. Компилятор разбивает исходный код именно на токены. Основные виды токенов: 1. Ключевые слова (int, if, class, return) — зарезервированные языком слова, которые нельзя использовать для имен переменных. 2. Идентификаторы — имена, придуманные программистом (имена переменных, функций, классов). 3. Литералы (константы) — непосредственно заданные значения: целые, вещественные, символьные, строковые, логические. 4. Операторы — символы для выполнения операций: арифметические (+, -, *), сравнения (==, <), присваивания (=), логические (&&,) и др. 5. Разделители и специальные символы — скобки (), [], {}, запятая ,, точка с запятой ;.
3.	В решении задач профессиональной деятельности программист использует базовые целочисленные типы данных в C++. Какие они бывают? Как выбрать подходящий тип?	Базовые целочисленные типы различаются по размеру (в байтах) и знаковости. 1. int: основной тип для целых чисел. Размер обычно 4 байта. Может быть signed int (по умолчанию, диапазон ~ -2.1 млрд до +2.1 млрд) и unsigned int (только неотрицательные, 0 до ~4.2 млрд). 2. short (short int): "короткое" целое, обычно 2 байта. Экономит память, но имеет меньший диапазон. 3. long (long int): "длинное" целое. Не менее 4 байт (часто 4 или 8). 4. long long: не менее 8 байт. Для очень больших чисел. 5. char: 1 байт. Может хранить символ (например, 'a') или малое целое число. Выбор типа зависит от: - Необходимого диапазона значений. Для счетчика, который не превысит 100, хватит short; для подсчета населения Земли нужен long long. - Экономии памяти. Если создается огромный массив чисел, лучше выбрать минимально достаточный тип. - Знаковости. Если переменная по смыслу не может быть отрицательной (возраст, количество), используйте unsigned. Для кроссплатформенности лучше использовать типы из <stdint>: int32_t, uint64_t и т.д.
4.	Одним из ключевых понятий современных информационных технологий является область видимости переменной. Что это такое? Какие виды областей видимости вы знаете?	Область видимости (scope) — это часть программы, в которой можно использовать идентификатор (имя переменной, функции и т.д.). 1. Локальная область видимости (блочная): переменная объявлена внутри блока кода { ... } (например, внутри функции, цикла, условного оператора). Видна только с момента объявления и до конца этого блока. 2. Глобальная область видимости: переменная объявлена вне всех функций. Видна из любого места

		файла, следующего после объявления. Использование глобальных переменных считается плохой практикой, так как усложняет понимание кода и приводит к ошибкам в коде. 3. Область видимости пространства имен: переменная/функция объявлена внутри пространства имен. Доступна через оператор разрешения области видимости :: (MyNamespace::myVariable). 4. Область видимости класса: члены класса (поля и методы) видны внутри методов этого класса.
5.	В решении задач профессиональной деятельности вы будете использовать принципы структурного программирования. Каковы они? Как они реализуются в C++?	Структурное программирование — это парадигма, направленная на улучшение читаемости, ясности и качества кода за счет использования только трех основных управляющих конструкций: 1. Следование: выполнение инструкций последовательно, одна за другой. 2. Ветвление: изменение порядка выполнения кода основанного на условиях (if, if-else, switch). 3. Повторение (Цикл): многократное выполнение блока кода (while, do-while, for). В C++ эти принципы реализуются с помощью соответствующих операторов. Запрещается использование "неструктурного" перехода goto, так как он создает т.н. «спагетти-код».
6.	В решении задач профессиональной деятельности вы будете использовать циклы while и do-while. В чем их разница?	Разница в моменте проверки условия. 1. while: сначала проверяет условие, и только если оно истинно, выполняет тело цикла. 2. do-while: сначала выполняет тело цикла, а затем проверяет условие. Тело гарантированно выполнится хотя бы один раз. do-while идеален для задач, где действие обязательно должно выполниться, например, запрос ввода от пользователя с последующей проверкой корректности.
7.	Одним из ключевых понятий современных информационных технологий является динамическая память? Дайте характеристику этого понятия. Для чего нужны операторы new и delete?	Динамическая память (куча, heap) — это область памяти, которая управляется программистом вручную во время выполнения программы. new: Оператор для выделения памяти в куче. Он находит блок памяти нужного размера, помечает его как занятый и возвращает указатель на этот блок. delete: Оператор для освобождения памяти, ранее выделенной new. Это критически важно для избежания утечек памяти. Зачем это нужно? Это позволяет создавать структуры данных, размер которых неизвестен на этапе компиляции (например, зависит от ввода пользователя), и существующие за пределами области видимости, где они были созданы.
8.	В решении задач профессиональной деятельности вы будете использовать односвязные списки. Каковы их структура, основные операции и их сложность?	Односвязный список — это динамическая структура данных, состоящая из узлов. Каждый узел содержит: 1. Данные (значение). 2. Указатель на следующий узел в списке. Главный указатель списка хранит адрес первого узла. Основные операции и их сложность: 1. Вставка в начало. Просто создать новый узел и направить его указатель на старый head, а затем сделать новый узел новым head. 2. Удаление из начала. Удалить узел head и сделать head равным head->next. 3. Вставка/удаление после известного узла. Мы меняем только указатели. 4. Вставка в конец. Придется пройти по всему списку до последнего узла, чтобы изменить его указатель. 5. Поиск элемента по значению/индексу. Требуется последовательный проход. 6. Доступ по индексу (получение i-го элемента). Нет произвольного доступа, как в массиве.
9.	Для решения задач профессиональной деятельности	Инкапсуляция: Механизм, который объединяет

	программисту нужно придерживаться трех основных принципов ООП: инкапсуляции, наследования и полиморфизма. Дайте им определение.	данные (поля) и методы, которые с этими данными работают, в единый объект — класс. Он также позволяет скрывать внутреннюю структуру от внешнего кода с помощью модификаторов доступа. Цель — защита данных от некорректного использования и упрощение интерфейса. Наследование: Механизм, позволяющий создать новый класс (производный класс/подкласс) на основе существующего (базового класса/суперкласса). Производный класс наследует все поля и методы базового и может добавлять свои или изменять унаследованные (переопределять). Цель — повторное использование кода и создание иерархий объектов. Полиморфизм: Возможность объектов с одинаковым интерфейсом (одинаковыми именами методов) иметь разную реализацию. В C++ достигается через виртуальные функции. Указатель базового класса может указывать на объект производного класса, и вызов метода через такой указатель приведет к выполнению кода производного класса.
10.	При решении задач профессиональной деятельности вы будете использовать конструктор и деструктор класса? Дайте определение этих терминов. Каковы их особенности?	Конструктор — это специальный метод класса, который автоматически вызывается при создании объекта. Его задача — инициализировать поля объекта и выделить необходимые ресурсы. Особенности: • Имя совпадает с именем класса. • Не имеет возвращаемого типа. • Может быть перегружен (несколько конструкторов с разными параметрами). • Конструктор по умолчанию — конструктор без параметров. Деструктор — это специальный метод класса, который автоматически вызывается при уничтожении объекта (при выходе из области видимости или при вызове delete). Его задача — освободить ресурсы, которые использовал объект (закрывать файлы, освободить динамическую память и т.д.). Особенности: • Имя: ~ИмяКласса. • Не имеет параметров и возвращаемого типа. • Не может быть перегружен (только один деструктор).
11.	Одними из ключевых понятий современных информационных технологий являются виртуальные функции и абстрактные классы. Для чего они нужны?	Виртуальные функции нужны для реализации полиморфизма. Они позволяют определить интерфейс в базовом классе, а реализация — в производных. Когда виртуальная функция вызывается через указатель или ссылку на базовый класс, выполняется реализация того производного класса, на который сейчас указывает указатель. Абстрактный класс — это класс, который содержит хотя бы одну чисто виртуальную функцию. Нельзя создать объект абстрактного класса. Его цель — служить интерфейсом, то есть задавать набор методов (контракт), которые обязаны реализовать все его конкретные (не абстрактные) производные классы.
12.	При решении задач профессиональной деятельности вы будете использовать библиотеку STL. Из каких основных компонентов она состоит?	STL построена на трех фундаментальных компонентах: 1. Контейнеры: шаблонные классы для хранения коллекций данных. 2. Алгоритмы: шаблонные функции для выполнения основных операций над данными (поиск, сортировка, модификация). Они независимы от контейнеров и работают через итераторы. 3. Итераторы: объекты, которые предоставляют доступ к элементам контейнера в унифицированном ключе. Они абстрагируют конкретную структуру контейнера и позволяют алгоритмам работать с разными контейнерами одинаково. Итераторы действуют как указатели (есть оператор * для

		разыменования, ++ для перехода к следующему элементу).
13.	В чем преимущества использования std::vector при решении задач профессиональной деятельности перед обычным динамическим массивом (int* arr = new int[n])?	std::vector — это обертка над динамическим массивом, которая автоматизирует управление памятью и предоставляет множество удобств: 1. Автоматическое управление памятью: не нужно вручную вызывать delete[]. При выходе из области видимости vector автоматически освободит память (благодаря RAII). 2. Автоматическое изменение размера: можно добавлять элементы (push back()), и vector сам выделит новую память большего размера, если это необходимо (reallocation). 3. Знание своего размера: у vector есть метод size(), который всегда возвращает текущее количество элементов. Для обычного массива приходится хранить размер отдельно. 4. Безопасность доступа: метод at() проверяет границы и генерирует исключение std::out_of_range, в отличие доступа по индексу [] или указателю, который может привести to undefined behavior. 5. Интеграция с STL: Vector полностью совместим со всеми алгоритмами STL и итераторами.
14.	Одним из ключевых понятий современных информационных технологий является итератор. Что это такое? Какие виды итераторов вы знаете?	Итератор — это объект, предоставляющий доступ к элементам контейнера и позволяющий их перебирать. Это абстракция над указателем. Основные категории итераторов (по возрастанию возможностей): 1. Input Iterator. 2. Output Iterator. 3. Forward Iterator. 4. Bidirectional Iterator. 5. Random Access Iterator.
15.	Одними из ключевых понятий современных информационных технологий в C++ являются шаблоны (templates). Дайте определение этого понятия. Для чего они нужны?	Шаблоны — это механизм обобщенного программирования. Они позволяют писать код, не зависящий от конкретного типа данных. Для чего нужны: чтобы избежать дублирования кода для разных типов. Вместо написания нескольких практически одинаковых функций (например, max_int, max_float, max_double) пишется одна шаблонная функция.

6.6. Шкалы и критерии оценивания по формам текущего контроля и промежуточной аттестации

Шкала и критерии оценивания

Оценка	Критерии оценивания для мероприятий контроля с применением 2-х балльной системы
«зачтено»	ОПК-4, ОПКЭ-6
«не зачтено»	Результаты обучения не сформированы на пороговом уровне