

Документ подписан простой электронной подписью.
Информация о владельце:

ФИО: Кандрашина Елена Александровна

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

Дата подписания: 10.07.2025 10:20:29

Уникальный программный ключ:

2db64eb9605ce27edd3b8e8fdd32c70e0674ddd2

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

Институт Институт экономики предприятий

Кафедра Программной инженерии (ПГУТИ)

УТВЕРЖДЕНО

Ученым советом Университета

(протокол № 10 от 22 мая 2025 г.)

РАБОЧАЯ ПРОГРАММА

Наименование дисциплины Б1.О.09 Программирование

Основная профессиональная образовательная программа 09.03.03 Прикладная информатика программа
Интеллектуальные цифровые системы и сервисы
в управлении

Квалификация (степень) выпускника Бакалавр

Самара 2025

Содержание (рабочая программа)

Стр.

- 1 Место дисциплины в структуре ОП
- 2 Планируемые результаты обучения по дисциплине, обеспечивающие достижение планируемых результатов обучения по программе
- 3 Объем и виды учебной работы
- 4 Содержание дисциплины
- 5 Материально-техническое и учебно-методическое обеспечение дисциплины
- 6 Фонд оценочных средств по дисциплине

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

1. Место дисциплины в структуре ОП

Дисциплина Программирование входит в обязательную часть блока Б1. Дисциплины (модули)

Предшествующие дисциплины по связям компетенций: Алгоритмизация и программирование, Высшая математика

Последующие дисциплины по связям компетенций: Проектирование баз данных, Теория вероятности и математическая статистика, Дискретная математика, Имитационное моделирование

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

Изучение дисциплины Программирование в образовательной программе направлено на формирование у обучающихся следующих компетенций:

Общепрофессиональные компетенции (ОПК):

ОПК-7 - Способен разрабатывать алгоритмы и программы, пригодные для практического применения

Планируемые результаты обучения по программе	Планируемые результаты обучения по дисциплине		
ОПК-7	ОПК-7.1: Знать:	ОПК-7.2: Уметь:	ОПК-7.3: Владеть (иметь навыки):
	основы алгоритмизации, языки программирования и работы с базами данных	применять языки программирования и работы с базами данных при решении прикладных задач	навыками использования технологии работы с базами данных, алгоритмизации и программирования для создания программных продуктов в прикладных областях деятельности

ОПК-1 - Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности

Планируемые результаты обучения по программе	Планируемые результаты обучения по дисциплине		
ОПК-1	ОПК-1.1: Знать:	ОПК-1.2: Уметь:	ОПК-1.3: Владеть (иметь навыки):
	основы математики, вычислительной техники и программирования	решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования	навыками теоретического и экспериментального исследования объектов профессиональной деятельности

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 2	Сем 3
Контактная работа, в том числе:	60.15/1.67	62.3/1.73
Занятия лекционного типа	32/0.89	32/0.89
Лабораторные работы (лабораторный практикум)	28/0.78	28/0.78
Индивидуальная контактная работа (ИКР)	0.15/0	0.3/0.01
Групповая контактная работа (ГКР)	/0	2/0.06
Самостоятельная работа:	65.85/1.83	47.7/1.32
Промежуточная аттестация	18/0.5	34/0.94
Вид промежуточной аттестации:		
Экзамен, Зачет	Зач	Экз
Общая трудоемкость (объем части образовательной программы): Часы	144	144
Зачетные единицы	4	4

4. Содержание дисциплины

4.1. Разделы, темы дисциплины и виды занятий:

Тематический план дисциплины Программирование представлен в таблице.

Разделы, темы дисциплины и виды занятий Очная форма обучения

№ п/п	Наименование темы (раздела) дисциплины	Контактная работа				Самосто	Планируемые результаты обучения в соотношении с результатами обучения по образовательной программе
		Лекц	Занятия семинарского типа	ИКР	ГКР		
1.	Понятия алгоритма и языка программирования. Разновидности языков программирования. Жизненный цикл программы.	4				2	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3
2.	Данные и память. Абстракция данных.	4	4			4	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3
3.	Язык программирования высокого уровня C / C++ и среда разработки Visual Studio.	4	4			4	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3
4.	Типы данных. Ключевые слова. Литералы в программировании. Целочисленные литералы. Строковые литералы.	4	4			6	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3
5.	Операции и операторы языка	4	4			6	ОПК-7.1, ОПК-7.2,

	программирования С / С++.						ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	
6.	Программирование алгоритмов циклической структуры. Итерационные циклы.	6	6			6	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	
7.	Функции, определяемые программистом, в языке С / С++.	6	6			6	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	
8.	Массивы: одномерные, двумерные (матрицы), многомерные. Алгоритмы обработки числовых массивов.	8	8			19	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	
9.	Указатели. Операции над указателями. Указатели на массивы. Массивы указателей.	8	8			19	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	
10.	Символы и строки в С / С++.	8	6			20	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	
11.	Статические и динамические структуры данных.	8	6			21,5	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	
	Контроль	52						
	Итого	64	56			113,5 5		

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.	Понятия алгоритма и языка программирования. Разновидности языков программирования. Жизненный цикл программы.	лекция	Предмет и задачи курса. Понятие алгоритма. Свойства алгоритма: дискретность, детерминированность, конечность, массовость. Формы записи алгоритмов. История развития языков программирования. Классификация языков программирования. Языки программирования: Ассемблер, Фортран, Бэйсик, Паскаль, С, С++, Python, Java, их назначение и особенности.
		лекция	Программы на языках программирования высокого уровня. Жизненный цикл программы: абстракция, математическая модель, алгоритм, запись алгоритма на языке

			программирования, трансляция в машинный код, компоновка программы, тестирование и отладка программы, эксплуатация и сопровождение программ.
2.	Данные и память. Абстракция данных.	лекция	Абстракция данных. Данные и память. Данные в языках программирования. Структуризация данных. Определение типа данных. Статический и динамический контроль типов и классификация языков в соответствии с ним.
		лекция	Слабая, сильная и строгая типизации в статически типизируемых языках. Конструкции структурного программирования и теорема о структурировании. Абстракция управления.
3.	Язык программирования высокого уровня C / C++ и среда разработки Visual Studio.	лекция	Среда разработки Visual Studio. Консольные приложения. Структура программы.
		лекция	Оформление исходного кода программы на языке программирования высокого уровня C / C++. Ключевые слова. Программирование алгоритмов разветвляющейся структуры.
4.	Типы данных. Ключевые слова. Литералы в программировании. Целочисленные литералы. Строковые литералы.	лекция	Определение типа данных в C / C++. Стандартные типы и действия с ними. Литералы в программировании. Целочисленные литералы. Строковые литералы. Иерархия типов данных в C / C++. Преобразования типов.
		лекция	Статический и динамический контроль типов и классификация языков в соответствии с ним. Слабая, сильная и строгая типизации в статически типизируемых языках. Типы данных, определяемые пользователем.
5.	Операции и операторы языка программирования C / C++.	лекция	Простые данные и вычисления. Выражения, правила их записи и выполнения. Классификация выражений в зависимости от типа результата. Операции и операторы языка программирования C / C++: арифметические, битовые, логические операции.
		лекция	Приоритеты выполнения операций. Директивы препроцессора, ввод и вывод в C / C++. Библиотечные функции в C / C++.
6.	Программирование алгоритмов циклической структуры. Итерационные	лекция	Программирование алгоритмов циклической структуры, организация специального выхода из цикла.

	циклы.	лекция	Понятие итерационного цикла.
		лекция	Генератор случайных чисел.
7.	Функции, определяемые программистом, в языке C / C++.	лекция	Функции, определяемые программистом. Описание, схема вызова функции. Формальные и фактические параметры, их взаимодействие.
		лекция	Параметры сложных типов. Механизмы отведения памяти под параметры. Стек. Возвращаемое значение. Механизмы передачи информации из функции и из основной программы. Область видимости имен. Локальные и глобальные идентификаторы.
		лекция	Вложенные функции. Функции с параметрами по умолчанию. Перегруженные функции. Использование статических переменных в функции. Понятие рекурсии. Реализация рекурсивных алгоритмов.
8.	Массивы: одномерные, двумерные (матрицы), многомерные. Алгоритмы обработки числовых массивов.	лекция	Одномерные массивы. Способы объявления и инициализации элементов массива.
		лекция	Соблюдение границ массива. Алгоритмы обработки числовых массивов с использованием индексов.
		лекция	Двумерный массив. Инициализация и обработка двумерного массива (матрицы).
		лекция	Многомерные массивы. Инициализация многомерных массивов.
9.	Указатели. Операции над указателями. Указатели на массивы. Массивы указателей.	лекция	Понятие указателя. Операции получения адреса объекта и разыменования указателя.
		лекция	Операции над указателями. Алгоритмы обработки числовых массивов с использованием указателей.
		лекция	Массивы указателей.
		лекция	Ступенчатые массивы. Указатели на массивы.
10.	Символы и строки в C / C++.	лекция	Символы. Строки. Способы объявления и инициализации.
		лекция	Библиотечные функции для работы со строками и символами в C / C++: копирование строк, конкатенация строк, сравнение строк, обработка символов, преобразование строк.
		лекция	Функции стандартной библиотеки ввода/вывода <stdio>. Способы передачи символьных строк в функцию.

		лекция	Некоторые алгоритмы обработки строк
11.	Статические и динамические структуры данных.	лекция	Статические структуры данных. Описание. Примеры использования.
		лекция	Динамические структуры данных: списки, очереди, деревья. Общие свойства динамических структур данных.
		лекция	Списки: односвязные, двусвязные. Очереди: циклическая очередь.
		лекция	Деревья. Примеры реализации динамических структур данных.

*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
2.	Данные и память. Абстракция данных.	лабораторные работы	Целый тип данных. Разработка простых программ.
		лабораторные работы	Программирование алгоритмов разветвляющейся структуры.
		лабораторные работы	Вычисление суммы ряда с заданной точностью.
5.	Операции и операторы языка программирования C / C++.	лабораторные работы	Математические функции языка C.
		лабораторные работы	Простые данные и вычисления.
6.	Программирование алгоритмов циклической структуры. Итерационные циклы.	лабораторные работы	Программирование алгоритмов циклической структуры. Итерационные циклы.
		лабораторные работы	Вычисление суммы ряда с заданной точностью.
7.	Функции, определяемые программистом, в языке C / C++.	лабораторные работы	Функции, определяемые программистом.
		лабораторные работы	Функции, определяемые программистом
		лабораторные работы	Функции, определяемые программистом
		лабораторные работы	Перегруженные функции.
		лабораторные работы	Перегруженные функции.
		лабораторные работы	Рекурсия.
		лабораторные работы	Реализация рекурсивных алгоритмов.
		лабораторные работы	Итерационные циклы.
8.	Массивы: одномерные, двумерные (матрицы), многомерные. Алгоритмы обработки числовых массивов	лабораторные работы	Одномерные массивы.
		лабораторные работы	Способы объявления и инициализации элементов массива.

		лабораторные работы	Генератор случайных чисел.
		лабораторные работы	Алгоритмы обработки одномерных массивов.
		лабораторные работы	Двумерные массивы.
		лабораторные работы	Инициализация и обработка двумерного массива (матрицы).
9.	Указатели. Операции над указателями. Указатели на массивы. Массивы указателей.	лабораторные работы	Указатели.
		лабораторные работы	Операции над указателями.
		лабораторные работы	Алгоритмы обработки числовых массивов с использованием указателей.
		лабораторные работы	Массивы указателей.
		лабораторные работы	Указатели на массивы.
10.	Символы и строки в C / C++.		Символы. Строки.
		лабораторные работы	Библиотечные функции для работы со строками и символами.
		лабораторные работы	Алгоритмы обработки строк.
11.	Статические и динамические структуры данных.	лабораторные работы	Алгоритмы обработки строк.

** 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.	Языки программирования: Ассемблер, Фортран, Бэйсик, Паскаль, С, С++, Python, Java, их назначение и особенности	- подготовка доклада - подготовка электронной презентации
2.	Статический и динамический контроль типов данных и классификация языков в соответствии с ним.	- подготовка доклада - подготовка электронной презентации
3.	Теорема о структурировании.	- подготовка доклада - подготовка электронной презентации
4.	Слабая, сильная и строгая типизации в статически типизируемых языках.	- подготовка доклада - подготовка электронной презентации
5.	Ступенчатые массивы. Указатели на массивы	- подготовка доклада - подготовка электронной презентации
6.	Динамические структуры данных: очереди, деревья.	- подготовка доклада - подготовка электронной презентации

--	--	--

*** 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. 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>
2. Кудрявцева, И. А. Программирование: теория типов : учебник для вузов / И. А. Кудрявцева, М. В. Швецкий. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 652 с. — (Высшее образование). — ISBN 978-5-534-11088-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/565920>

Дополнительная литература

1. 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>
2. Кудрявцева, И. А. Программирование: комбинаторная логика : учебник для вузов / И. А. Кудрявцева, М. В. Швецкий. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 524 с. — (Высшее образование). — ISBN 978-5-534-10620-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/565666>

Литература для самостоятельного изучения

1.

5.2. Перечень лицензионного программного обеспечения

1. Microsoft Visual Studio

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. Профессиональная база данных «Финансово-экономические показатели Российской Федерации» (Официальный сайт Министерства финансов РФ - <https://www.minfin.ru/ru/>)
3. Профессиональная база данных «Официальная статистика» (Официальный сайт Федеральной службы государственной статистики - <http://www.gks.ru/>)

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» , 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

5.6 Лаборатории и лабораторное оборудование

Аудитории для лабораторных занятий	Количество посадочных мест по количеству обучающихся. Компьютеры с выходом в сеть «Интернет»
------------------------------------	---

6. Фонд оценочных средств по дисциплине Программирование:

6.1. Контрольные мероприятия по дисциплине

Вид контроля	Форма контроля	Отметить нужное знаком « + »
Текущий контроль	Оценка докладов	+
	Устный/письменный опрос	+
	Тестирование	
	Практические задачи	
	Оценка контрольных работ (для заочной формы обучения)	

	Зачет	+
	Экзамен	+

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-7 - Способен разрабатывать алгоритмы и программы, пригодные для практического применения

Планируемые результаты обучения по программе	Планируемые результаты обучения по дисциплине		
	ОПК-7.1: Знать: основы алгоритмизации, языки программирования и работы с базами данных	ОПК-7.2: Уметь: применять языки программирования и работы с базами данных при решении прикладных задач	ОПК-7.3: Владеть (иметь навыки): навыками использования технологии работы с базами данных, алгоритмизации и программирования для создания программных продуктов в прикладных областях деятельности
Пороговый	Знать основные языки программирования и работы с базами данных.	Уметь применять языки программирования и работы с базами данных, обнаруживать ошибки в ходе тестирования программы.	Владеть навыками программирования.
Стандартный (в дополнение к пороговому)	Знать основные языки программирования и работы с базами данных, операционные системы и оболочки.	Уметь применять современные программные среды разработки информационных систем и технологий при решении прикладных задач.	Владеть методом оценки программного обеспечения, навыками в области разработки программного обеспечения и тестирования программного обеспечения.
Повышенный (в дополнение к пороговому, стандартному)	Знать научную литературу в области создания программного обеспечения, основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды.	Уметь проектировать и разрабатывать программное обеспечение на языках программирования высокого уровня для решения практических задач, а так же тестировать программное обеспечение	Владеть существенными навыками работы с инструментальными средами разработки программного обеспечения, а так же тестирования программного обеспечения

ОПК-1 - Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности

Планируемые результаты обучения по программе	Планируемые результаты обучения по дисциплине		
	ОПК-1.1: Знать:	ОПК-1.2: Уметь:	ОПК-1.3: Владеть (иметь навыки):
	основы математики, вычислительной техники и программирования	решать стандартные профессиональные задачи с применением естественнонаучных и общинженерных знаний, методов математического анализа и моделирования	навыками теоретического и экспериментального исследования объектов профессиональной деятельности
Пороговый	Знать основы математики, методы математического анализа.	Уметь применять методы математического анализа, создавать программные прототипы решения прикладных задач.	Владеть навыками программирования приложений и создания программных прототипов решения прикладных задач.
Стандартный (в дополнение к пороговому)	Знать основы математики, методы математического анализа, основные положения структурного и объектно-ориентированного программирования	Уметь применять методы математического анализа, создавать программные прототипы решения прикладных задач, применять на практике принципы объектно-ориентированного программирования: абстракцию, инкапсуляцию, наследование, полиморфизм.	Владеть навыками программирования приложений и создания программных прототипов решения прикладных задач, а также использования классов, объектов, виртуальных функций.
Повышенный (в дополнение к пороговому, стандартному)	Знать основы математики, физики, вычислительной техники и программирования, научную литературу в области создания программного обеспечения и программирования.	Уметь измерять время выполнения, протоколировать и анализировать ход выполнения приложения, узкие участки кода, возникновения событий, выделять память проектировать и разрабатывать программное обеспечение на языках программирования высокого уровня для решения практических задач.	Владеть навыками адаптации программ к различным категориям пользователей, решения концептуальных задач, навыками теоретического и экспериментального исследования объектов.

6.3. Паспорт оценочных материалов

№ п/п	Наименование темы	Контролируемые	Вид контроля/используемые
-------	-------------------	----------------	---------------------------

	(раздела) дисциплины	планируемые результаты обучения в соотношении с результатами обучения по программе	оценочные средства	
			Текущий	Промежуточный
1.	Понятия алгоритма и языка программирования. Разновидности языков программирования. Жизненный цикл программы.	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	Устный/письменный опрос Исходные коды программ	Зачет
2.	Данные и память. Абстракция данных.	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	Устный/письменный опрос Исходные коды программ	Зачет
3.	Язык программирования высокого уровня C / C++ и среда разработки Visual Studio.	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	Устный/письменный опрос	Зачет
4.	Типы данных. Ключевые слова. Литералы в программировании. Целочисленные литералы. Строковые литералы.	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	Устный/письменный опрос Исходные коды программ	Зачет
5.	Операции и операторы языка программирования C / C++.	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	Устный/письменный опрос Исходные коды программ	Зачет
6.	Программирование алгоритмов циклической структуры. Итерационные циклы.	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	Устный/письменный опрос Исходные коды программ	Зачет
7.	Функции, определяемые программистом, в языке C / C++.	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	Устный/письменный опрос Исходные коды программ	Зачет
8.	Массивы: одномерные, двумерные (матрицы), многомерные. Алгоритмы обработки числовых массивов.	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	Устный/письменный опрос Исходные коды программ	Экзамен
9.	Указатели. Операции над указателями. Указатели на массивы. Массивы указателей.	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	Устный/письменный опрос Исходные коды программ	Экзамен
10.	Символы и строки в C /	ОПК-7.1, ОПК-7.2,	Устный/письменный	Экзамен

	C++.	ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	ный опрос Исходные коды программ	
11.	Статические и динамические структуры данных	ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3	Устный/письменный опрос Исходные коды программ	Экзамен

6.4.Оценочные материалы для текущего контроля

Примерная тематика докладов

Раздел дисциплины	Темы
Понятия алгоритма и языка программирования. Разновидности языков программирования. Жизненный цикл программы.	1. Классификация языков программирования. 2. Языки программирования: Ассемблер, Фортран, Бэйсик, Паскаль, С, С++, Python, Java, их назначение и особенности. Жизненный цикл программы: абстракция, математическая
Данные и память. Абстракция данных.	1. Структуризация данных. 2. Определение типа данных. Слабая, сильная и строгая типизации в статически типизируемых языках
Язык программирования высокого уровня С / С++ и среда разработки Visual Studio.	1. Структура программы на языке программирования С / С++. 2. Оформление исходного кода программы. 3. Ключевые слова. 4. Среда разработки Visual Studio.
Типы данных. Ключевые слова. Литералы в программировании. Целочисленные литералы. Строковые литералы.	1. Литералы в программировании. Целочисленные литералы. Строковые литералы. 2. Иерархия типов данных в С / С++. 3. Преобразования типов.
Операции и операторы языка программирования С / С++.	1. Классификация выражений в зависимости от типа результата. 2. Операции и операторы языка программирования С / С++: арифметические, битовые, логические операции. 3. Приоритеты выполнения операций.
Программирование алгоритмов циклической структуры. Итерационные циклы.	1. Циклы с предусловием и постусловием. 2. Организация специального выхода из цикла. 3. Понятие итерационного цикла.
Функции, определяемые программистом, в языке С / С++.	1. Формальные и фактические параметры функции, их взаимодействие. 2. Параметры сложных типов. Механизмы отведения памяти под параметры. 3. Стек. Возвращаемое значение. Механизмы передачи информации из функции и из основной программы. 4. Область видимости имен. Локальные и глобальные идентификаторы. 5. Вложенные функции.

Массивы: одномерные, двумерные (матрицы), многомерные. Алгоритмы обработки числовых массивов.	<ol style="list-style-type: none"> 1. Одномерные массивы. 2. Способы объявления и инициализации элементов массива. Соблюдение границ массива. 3. Алгоритмы обработки числовых массивов с использованием индексов. 4. Двумерный массив. Инициализация и обработка двумерного массива (матрицы).
Указатели. Операции над указателями. Указатели на массивы. Массивы указателей.	<ol style="list-style-type: none"> 1. Понятие указателя. Операции получения адреса объекта и разыменования указателя. 2. Операции над указателями. 3. Алгоритмы обработки числовых массивов с использованием указателей. 4. Массивы указателей.
Символы и строки в C / C++.	<ol style="list-style-type: none"> 1. Символы. Способы объявления и инициализации. 2. Строки. Способы объявления и инициализации. 3. Библиотечные функции для работы со строками и символами в C / C++: копирование строк, конкатенация строк, сравнение строк, обработка символов, преобразование строк.
Статические и динамические структуры данных	<ol style="list-style-type: none"> 1. Статические структуры данных. Описание. Примеры использования. 2. Динамические структуры данных: списки, очереди, деревья. 3. Общие свойства динамических структур данных. 4. Списки: односвязные, двусвязные

Вопросы для устного/письменного опроса

Раздел дисциплины	Вопросы
<p>Понятия алгоритма и языка программирования.</p> <p>Разновидности языков программирования.</p> <p>Жизненный цикл программы.</p>	<ol style="list-style-type: none"> 3. Понятие алгоритма. Свойства алгоритма. 4. Формы записи алгоритмов. 5. История развития языков программирования. 6. Классификация языков программирования. 7. Языки программирования: Ассемблер, Фортран, Бэйсик, Паскаль, C, C++, Python, Java, их назначение и особенности. 8. Жизненный цикл программы: абстракция, математическая модель, алгоритм, запись алгоритма на языке программирования. 9. тестирование и отладка программ. 10. Эксплуатация и сопровождение программ.
<p>Данные и память. Абстракция данных.</p>	<ol style="list-style-type: none"> 3. Данные в языках программирования. 4. Структуризация данных. 5. Определение типа данных. 6. Слабая, сильная и строгая типизации в статически типизируемых языках. 7. Конструкции структурного программирования. 8. Абстракция управления.

<p>Язык программирования высокого уровня C / C++ и среда разработки Visual Studio.</p>	<ol style="list-style-type: none"> 5. Структура программы на языке программирования C / C++. 6. Оформление исходного кода программы. 7. Ключевые слова. 8. Среда разработки Visual Studio. 9. Программирование алгоритмов разветвляющейся структуры.
<p>Типы данных. Ключевые слова. Литералы в программировании. Целочисленные литералы. Строковые литералы.</p>	<ol style="list-style-type: none"> 4. Определение типа данных в C / C++. Стандартные типы и действия с ними. 5. Литералы в программировании. Целочисленные литералы. Строковые литералы. 6. Иерархия типов данных в C / C++. 7. Преобразования типов. 8. Слабая, сильная и строгая типизации в статически типизируемых языках. 9. Типы данных, определяемые пользователем.
<p>Операции и операторы языка программирования C / C++.</p>	<ol style="list-style-type: none"> 10. Выражения, правила их записи и выполнения на языке программирования C / C++. 11. Классификация выражений в зависимости от типа результата. 12. Операции и операторы языка программирования C / C++: арифметические, битовые, логические операции. 13. Приоритеты выполнения операций. 14. Директивы препроцессора, ввод и вывод в C / C++. 15. Библиотечные функции в C / C++.
<p>Программирование алгоритмов циклической структуры. Итерационные циклы.</p>	<ol style="list-style-type: none"> 4. Программирование алгоритмов циклической структуры. 5. Циклы с предусловием и постусловием. 6. Организация специального выхода из цикла. 7. Понятие итерационного цикла. 8. Генератор случайных чисел.
<p>Функции, определяемые программистом, в языке C / C++.</p>	<ol style="list-style-type: none"> 6. Функции, определяемые программистом. 7. Описание, схема вызова функции. 8. Формальные и фактические параметры функции, их взаимодействие. 9. Параметры сложных типов. Механизмы отведения памяти под параметры. 10. Стек. Возвращаемое значение. Механизмы передачи информации из функции и из основной программы. 11. Область видимости имен. Локальные и глобальные идентификаторы. 12. Вложенные функции. 13. Функции с параметрами по умолчанию. 14. Перегруженные функции. 15. Использование статических переменных в функции. 16. Понятие рекурсии.
<p>Массивы: одномерные, двумерные (матрицы), многомерные. Алгоритмы обработки числовых массивов.</p>	<ol style="list-style-type: none"> 5. Одномерные массивы. 6. Способы объявления и инициализации элементов массива. Соблюдение границ массива. 7. Алгоритмы обработки числовых массивов с использованием индексов. 8. Двумерный массив. Инициализация и обработка двумерного массива (матрицы). 9. Многомерные массивы. Инициализация многомерных массивов.
<p>Указатели. Операции над указателями. Указатели на</p>	<ol style="list-style-type: none"> 5. Понятие указателя. Операции получения адреса объекта и разыменования указателя.

массивы. указателей.	Массивы	<ol style="list-style-type: none"> 6. Операции над указателями. 7. Алгоритмы обработки числовых массивов с использованием указателей. 8. Массивы указателей. 9. Ступенчатые массивы. 10. Указатели на массивы.
Символы и строки в C / C++.		<ol style="list-style-type: none"> 4. Символы. Способы объявления и инициализации. 5. Строки. Способы объявления и инициализации. 6. Библиотечные функции для работы со строками и символами в C / C++: копирование строк, конкатенация строк, сравнение строк, обработка символов, преобразование строк. 7. Функции стандартной библиотеки ввода/вывода <stdio>. 8. Способы передачи символьных строк в функцию. 9. Некоторые алгоритмы обработки строк.
Статические и динамические структуры данных.		<ol style="list-style-type: none"> 5. Статические структуры данных. Описание. Примеры использования. 6. Динамические структуры данных: списки, очереди, деревья. 7. Общие свойства динамических структур данных. 8. Списки: односвязные, двусвязные. 9. Очереди: циклическая очередь. 10. Деревья. 11. Примеры реализации динамических структур данных.
Понятия алгоритма и языка программирования. Разновидности языков программирования. Жизненный цикл программы.		<ol style="list-style-type: none"> 11. Понятие алгоритма. Свойства алгоритма. 12. Формы записи алгоритмов. 13. История развития языков программирования. 14. Классификация языков программирования. 15. Языки программирования: Ассемблер, Фортран, Бэйсик, Паскаль, C, C++, Python, Java, их назначение и особенности. 16. Жизненный цикл программы: абстракция, математическая модель, алгоритм, запись алгоритма на языке программирования. 17. тестирование и отладка программ. 18. Эксплуатация и сопровождение программ.
Данные и память. Абстракция данных.		<ol style="list-style-type: none"> 9. Данные в языках программирования. 10. Структуризация данных. 11. Определение типа данных. 12. Слабая, сильная и строгая типизации в статически типизируемых языках. 13. Конструкции структурного программирования. 14. Абстракция управления.

6.5. Оценочные материалы для промежуточной аттестации

Фонд вопросов для проведения промежуточного контроля в форме зачета

Раздел дисциплины	Вопросы
Понятия алгоритма и языка программирования. Разновидности языков программирования. Жизненный цикл программы.	<ol style="list-style-type: none"> 1. Алфавит языка. Идентификаторы. Константы. Выражения. Понятие операции. 2. Понятие переменной 3. Объявление переменных.

<p>Данные и память. Абстракция данных.</p>	<ol style="list-style-type: none"> 1. Типы данных, определяемые пользователем. 2. Инициализация переменных. 3. Основные типы переменных 4. Синтаксис объявления имени переменной
<p>Язык программирования высокого уровня C / C++ и среда разработки Visual Studio.</p>	<ol style="list-style-type: none"> 1. Основные конструкции алгоритмического языка 2. Жизненный цикл программы 3. Особенности языка C++ 4. Структура программы в языке C++ 5. Понятие константы. Объявление.
<p>Типы данных. Ключевые слова. Литералы в программировании. Целочисленные литералы. Строковые литералы.</p>	<ol style="list-style-type: none"> 1. Понятие константы. Инициализация. 2. Простейшие арифметические операции 3. Стандартные арифметические процедуры и функции. 4. Операция %. Ее особенности. 5. Основные математические функции
<p>Операции и операторы языка программирования C / C++.</p>	<ol style="list-style-type: none"> 1. Принципы структурного программирования 2. Оператор множественного ветвления switch 3. Операторы цикла. Операторы break и continue.
<p>Программирование алгоритмов циклической структуры. Итерационные циклы.</p>	<ol style="list-style-type: none"> 1. Условный оператор if-else 2. Вложенный оператор if. 3. Классификация циклов 4. Операции инкремента и декремента. 5. Оператор while() 6. Оператор For. 7. Оператор do...while() 8. Принципы модульного программирования
<p>Функции, определяемые программистом, в языке C / C++.</p>	<ol style="list-style-type: none"> 1. Описание функции 2. Вызов функции. 3. Прототипы функций. 4. Передача параметров по имени. 5. Принципы объектно-ориентированного программирования. Инкапсуляция. 6. Принципы объектно-ориентированного программирования. Полиморфизм. 7. Принципы объектно-ориентированного программирования. Наследование. 8. Структуры. Объявление. 9. Структуры. Инициализация. 10. Прямой доступ к элементам структуры. 11. Косвенный доступ к элементам структуры.

Фонд вопросов для проведения промежуточного контроля в форме экзамена

Раздел дисциплины	Вопросы
<p>Массивы: одномерные, двумерные (матрицы), многомерные. Алгоритмы обработки числовых массивов.</p>	<ol style="list-style-type: none"> 1. История языков программирования от машинного кода до современных языков структурного программирования. 2. Алгоритмический язык C — язык компилируемого типа. Структура программы на языке C++. Стадии получения загрузочного модуля в C/C++. 3. Понятие о синтаксисе языка программирования. Основные типы данных языка C. 4. Описание переменных и констант в языке C/C++. 5. История языков программирования от машинного кода до современных языков структурного программирования. 6. Алгоритмический язык C — язык компилируемого типа. Структура программы на языке C++. Стадии получения загрузочного модуля в C/C++. 7. Понятие о синтаксисе языка программирования. Основные типы данных языка C. 8. Описание переменных и констант в языке C/C++. 9. Одномерные массивы: объявление, инициализация, обработка, использование массивов в C/C++. 10. Одномерные массивы: последовательный поиск элементов в массивах и его организация на языке C/C++.
<p>Указатели. Операции над указателями. Указатели на массивы. Массивы указателей.</p>	<ol style="list-style-type: none"> 1. Управляющие последовательности, манипуляторы, используемые при форматном выводе в языке C/C++. Представление не изображаемых символов. Привести примеры. 2. Арифметические операции языка C/C++. Синтаксис операторов. Операции увеличения ++ и уменьшения --. Привести примеры реализации. 3. Операции сравнения и логические операции в языке C/C++. Синтаксис операторов. Привести примеры реализации. Приоритет операций для языка C. 4. Стандартные математические функции языка программирования C/C++. Привести примеры использования. 5. Структурный подход к программированию и его реализация в языке C/C++. Рассмотреть структуры языка C/C++, позволяющие сделать единичный, двойной и множественный выбор. 6. Базовые конструкции структурного программирования в C/C++. Оператор switch. Привести синтаксис и пример использования оператора в программе. 7. Операторы повторения в C/C++. Привести блок схемы и примеры реализации операторов цикла. Дать рекомендации по выбору наилучшего оператора цикла для конкретного случая. 8. Циклы с предусловием и с постусловием в C/C++. Объяснить различия. Привести блок схемы и примеры реализации. 9. Цикл с выходом из середины. Привести блок схему и примеры реализации. 10. Что такое вложенные циклы? Проанализировать работу программы, содержащей вложенные циклы.
<p>Символы и строки в C / C++.</p>	<ol style="list-style-type: none"> 1. Итерационные алгоритмы. Привести примеры реализации итерационных алгоритмов в задачах вычисления суммы

	<p>бесконечного ряда с заданной точностью.</p> <p>2. Пользовательские функции в C/C++. Разновидности функций. Определение функций. Привести пример функции, возвращающей значение.</p> <p>3. Способы передачи параметров в вызываемую функцию на языке C/C++. Последовательность выполнения вызова функции. Привести примеры.</p> <p>4. Правила разработки функций. Формальные и фактические параметры функции. Привести примеры определения и вызова функций в C/C++.</p> <p>5. Функция с параметрами по умолчанию. Привести пример определения и вызова такой функции в C/C++.</p>
Статические и динамические структуры данных	<p>1. Область видимости и "время жизни" переменных. Использование статических переменных в функции. Привести примеры.</p> <p>2. Передача параметров в функцию по ссылке. Привести примеры определения и вызова таких функций в C/C++.</p> <p>3. Перегрузка функций. Пример реализации перегруженной функции.</p> <p>4. Одномерные массивы: объявление, инициализация,</p> <p>5. Определения тестирования ПО</p> <p>6. Методы тестирования ПО</p> <p>7. Определение ошибок в ходе тестирования ПО</p>

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-7, ОПК-1
«не зачтено»	Результаты обучения не сформированы на пороговом уровне

Оценка	Критерии оценивания для мероприятий контроля с применением 4-х балльной системы
«отлично»	Повышенный ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3
«хорошо»	Стандартный ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК-1.3
«удовлетворительно»	Пороговый ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-1.1, ОПК-1.2, ОПК- 1.3
«неудовлетворительно»	Результаты обучения не сформированы на пороговом уровне