

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Заболотный Г.И. Заболотный
Должность: Директор филиала
Дата подписания: 08.05.2026 10:30:58
Уникальный программный ключ:
476db7d4accb36ef8130172be235477473d63457266ce26b7e9e40f733b8b08

МИНОБРАЗОВАНИЯ РОССИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Самарский государственный технический университет»
(ФГБОУ ВО «СамГТУ»)

УТВЕРЖДАЮ:

Директор филиала ФГБОУ ВО
"СамГТУ" в г. Новокуйбышевске

_____ / Г.И. Заболотный

" ____ " _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.05.02 «Языки и методы программирования»

Код и направление подготовки (специальность)	09.03.01 Информатика и вычислительная техника
Направленность (профиль)	Информатика и вычислительная техника в нефтехимическом производстве
Квалификация	Бакалавр
Форма обучения	Очная
Год начала подготовки	2026
Институт / факультет	Кафедры филиала ФГБОУ ВО "СамГТУ" в г. Новокуйбышевске
Выпускающая кафедра	кафедра "Информатика и системы управления" (НФ-ИиСУ)
Кафедра-разработчик	кафедра "Информатика и системы управления" (НФ-ИиСУ)
Объем дисциплины, ч. / з.е.	324 / 9
Форма контроля (промежуточная аттестация)	Экзамен

Б1.О.05.02 «Языки и методы программирования»

Рабочая программа дисциплины разработана в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) **09.03.01 Информатика и вычислительная техника**, утвержденного приказом Министерства образования и науки РФ от № 929 от 19.09.2017 и соответствующего учебного плана.

Разработчик РПД:

Старший преподаватель

(должность, степень, ученое звание)

Т.П Рубцова

(ФИО)

Заведующий кафедрой

А.В. Волкодаева, кандидат
экономических наук, доцент

(ФИО, степень, ученое звание)

СОГЛАСОВАНО:

Председатель методического совета
факультета / института (или учебно-
методической комиссии)

Е.Т Демидова, кандидат
юридических наук, доцент

(ФИО, степень, ученое звание)

Руководитель образовательной
программы

А.В. Волкодаева, кандидат
экономических наук, доцент

(ФИО, степень, ученое звание)

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы	4
2. Место дисциплины (модуля) в структуре образовательной программы	4
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся	5
4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий	5
4.1 Содержание лекционных занятий	6
4.2 Содержание лабораторных занятий	9
4.3 Содержание практических занятий	10
4.4. Содержание самостоятельной работы	14
5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)	14
6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения	15
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем	16
8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)	16
9. Методические материалы	17
10. Фонд оценочных средств по дисциплине (модулю)	18

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
Общепрофессиональные компетенции			
	ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Владеть навыками составления алгоритмов и программ, пригодных для практического применения
			Знать методы составления алгоритмов и программ, пригодных для практического применения
			Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения
		ОПК-8.2 . Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса
			Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса
			Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса

2. Место дисциплины (модуля) в структуре образовательной программы

Место дисциплины (модуля) в структуре образовательной программы: **обязательная часть**

Код компетенции	Предшествующие дисциплины	Параллельно осваиваемые дисциплины	Последующие дисциплины

ОПК-8		Выполнение и защита выпускной квалификационной работы; Технологии программирования
-------	--	---

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Вид учебной работы	Всего часов / часов в электронной форме	1 семестр часов / часов в электронной форме	2 семестр часов / часов в электронной форме
Аудиторная контактная работа (всего), в том числе:	128	48	80
Лекции	32	16	16
Практические занятия	80	32	48
Лабораторные работы	16	0	16
Самостоятельная работа (всего), в том числе:	124	60	64
подготовка к зачету	20	20	0
подготовка к лекциям	20	10	10
подготовка к практическим занятиям	54	30	24
подготовка к лабораторным работам	10	0	10
подготовка к экзамену	20	0	20
Контроль	72	36	36
Итого: час	324	144	180
Итого: з.е.	9	4	5

4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий

№ раздела	Наименование раздела дисциплины	Виды учебной нагрузки и их трудоемкость, часы				
		ЛЗ	ЛР	ПЗ	СРС	Всего часов
2	Развитие языков программирования. Парадигмы программирования	16	0	32	60	108
4	Основные элементы языков программирования	16	16	48	64	144
	Контроль	0	0	0	0	72
	Итого	32	16	80	124	324

4.1 Содержание лекционных занятий

№ занятия	Наименование раздела	Тема лекции	Содержание лекции (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
1 семестр				
1	Развитие языков программирования. Парадигмы программирования	История и эволюция методов и языков программирования. Синтаксис, семантика, прагматика.	Эволюция методов и языков. Особенности формальных языков. Грамматики, классификация языков (контекстно-зависимые языки, контекстно-свободные языки, регулярные языки). Синтаксис и основные классы понятий, семантика и прагматика языков программирования.	2
2	Развитие языков программирования. Парадигмы программирования	История и эволюция методов и языков программирования. Синтаксис, семантика, прагматика.	Процедурная и структурная парадигмы. Программирование сверху-вниз. Программирование снизу-вверх. Низкоуровневые и высокоуровневые языки. Дихотомия императивного и декларативного подходов. Переход от императивного к декларативному программированию. Объектно-ориентированная парадигма. Функциональная парадигма. Основы парадигмальной декомпозиции. Средства поддержки автоматизации интеграционных решений и тестирования.	2
3	Развитие языков программирования. Парадигмы программирования	Основные элементы языков программирования	Введение в программирование на СИ. Структура программы. Подпрограммы, их назначение и классификация. Области доступности глобальных и локальных переменных. Объявление, инициализация.	2
17	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Ввод-вывод данных	2
18	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Условный оператор	2

19	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Типы данных	2
20	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Циклы for и while	2
21	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Строковый тип данных	2
Итого за семестр:				16
2 семестр				
9	Основные элементы языков программирования	Стандартные структуры данных	Принципы объектного подхода: абстрагирование, ограничение доступа, модульность, иерархия, типизация, параллелизм, устойчивость. Способы распределения и заимствования шаблонов и поведения. Инкапсуляция и спецификация правил доступности элементов класса. Конструкторы и деструкторы. Наследование. Иерархия классов. Одиночное и множественное наследование. Способы реализации множественного наследования, их достоинства и недостатки. Таблицы виртуальных функций. Абстрактные базовые классы. Конечные производные классы.	2
10	Основные элементы языков программирования	Основные положения объектного подхода к разработке программ.	Принципы объектного подхода: абстрагирование, ограничение доступа, модульность, иерархия, типизация, параллелизм, устойчивость. Способы распределения и заимствования шаблонов и поведения. Инкапсуляция и спецификация правил доступности элементов класса. Конструкторы и деструкторы. Наследование. Иерархия классов. Одиночное и множественное наследование. Способы реализации множественного наследования, их достоинства и недостатки. Таблицы виртуальных функций. Абстрактные базовые классы. Конечные производные классы.	2

11	Основные элементы языков программирования	Основные положения объектного подхода к разработке программ.	Принципы объектного подхода: абстрагирование, ограничение доступа, модульность, иерархия, типизация, параллелизм, устойчивость. Способы распределения и заимствования шаблонов и поведения. Инкапсуляция и спецификация правил доступности элементов класса. Конструкторы и деструкторы. Наследование. Иерархия классов. Одиночное и множественное наследование. Способы реализации множественного наследования, их достоинства и недостатки. Таблицы виртуальных функций. Абстрактные базовые классы. Конечные производные классы.	2
12	Основные элементы языков программирования	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования	Полиморфизм. Ранее и позднее связывание. Управление последовательностью действий в объектно -ориентированной программе. Объект и процесс. Инициализация и взаимодействие объектов и процессов. Сообщения. Реализация механизмов посылки сообщений. Примеры функционирования объектно-ориентированной программы. Проектирование библиотек классов. Виды классов: конкретный тип, абстрактный тип, узловой класс, интерфейсный класс. Динамическая идентификация типа. Управление видимостью и областью действия имен. Управление памятью. Библиотеки контейнерных классов. Номенклатура контейнеров и примеры их использования. Иерархия классов исключений.	2
13	Основные элементы языков программирования	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования	Полиморфизм. Ранее и позднее связывание. Управление последовательностью действий в объектно -ориентированной программе. Объект и процесс. Инициализация и взаимодействие объектов и процессов. Сообщения. Реализация механизмов посылки сообщений. Примеры функционирования объектно-ориентированной программы. Проектирование библиотек классов. Виды классов: конкретный тип, абстрактный тип, узловой класс, интерфейсный класс. Динамическая идентификация типа. Управление видимостью и областью действия имен. Управление памятью. Библиотеки контейнерных классов. Номенклатура контейнеров и примеры их использования. Иерархия классов исключений.	2
14	Основные элементы языков программирования	Обзор методов проектирования и разработки программного обеспечения.	Основные подходы к разработке программ: процедурное, логическое, функциональное и объектно-ориентированное программирование. Процедурное и декларативное представление данных.	2
15	Основные элементы языков программирования	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования	Типы данных в языках программирования. Связь между данными и операциями. Абстрактные типы данных. Понятие интерфейса.	2

16	Основные элементы языков программирования	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования	Типы данных в языках программирования. Связь между данными и операциями. Абстрактные типы данных. Понятие интерфейса.	2
Итого за семестр:				16
Итого:				32

4.2 Содержание лабораторных занятий

№ занятия	Наименование раздела	Тема лабораторного занятия	Содержание лабораторного занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
2 семестр				
1	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Решение объектных задач.	2
2	Основные элементы языков программирования	Отношения между объектами и/или классами	Решение задач на разработку классов.	2
3	Основные элементы языков программирования	Отношения между объектами и/или классами	Решение задач на разработку классов.	2
4	Основные элементы языков программирования	Основные подходы к разработке программ	Программирование статических изображений.	2
5	Основные элементы языков программирования	Основные подходы к разработке программ	Программирование движущихся изображений.	2
6	Основные элементы языков программирования	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования.	Решение задач на разработку интерфейсов.	2
7	Основные элементы языков программирования	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования.	Решение задач на разработку интерфейсов.	2
8	Основные элементы языков программирования	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования.	Решение задач на разработку интерфейсов.	2
Итого за семестр:				16
Итого:				16

4.3 Содержание практических занятий

№ занятия	Наименование раздела	Тема практического занятия	Содержание практического занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
1 семестр				
41	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Ввод-вывод данных	2
42	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Условный оператор	2
43	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Типы данных	2
44	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Циклы for и while	2
45	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Строковый тип данных	2

46	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Списки	2
47	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Функции	2
48	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Множества	2
49	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Словари	2
50	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Модули random и string	2
51	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Модули decimal, fraction и complex	2

52	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Функции с переменным количеством аргументов	2
53	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Встроенные функции map(), filter(), reduce()	2
54	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Работа с файлами	2
55	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Функции высшего порядка	2
56	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Парадигмы программирования	2
Итого за семестр:				32
2 семестр				
17	Основные элементы языков программирования	Стандартные структуры данных	Функции в языке программирования C++. Рекурсия. Встраиваемые функции. Перегрузка функций, использование аргументов по умолчанию.	2
18	Основные элементы языков программирования	Стандартные структуры данных	Указатели в языке программирования C++. Объявление и разыменовывание указателей.	2
19	Основные элементы языков программирования	Стандартные структуры данных	Указатели в языке программирования C++. Нулевые указатели и указатели на тип Void.	2

20	Основные элементы языков программирования	Стандартные структуры данных	Ссылки в языке программирования C++. Резервирование памяти в куче.	2
21	Основные элементы языков программирования	Модульное программирование	Модульное программирование	2
22	Основные элементы языков программирования	Модульное программирование	Модульное программирование	2
23	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Основы ООП.	2
24	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Основы ООП.	2
25	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Наследование. Одиночное и множественное наследование. Способы реализации множественного наследования.	2
26	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Наследование. Одиночное и множественное наследование. Способы реализации множественного наследования.	2
27	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Наследование. Одиночное и множественное наследование. Способы реализации множественного наследования.	2
28	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Проектирование библиотек классов.	2
29	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Управление видимостью и областью действия имен. Управление памятью.	2
30	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Библиотеки контейнерных классов. Номенклатура контейнеров и примеры их использования. Иерархия классов исключений.	2
31	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Обработка исключений.	2
32	Основные элементы языков программирования	Основы объектно-ориентированного программирования	Примеры функционирования объектно-ориентированной программ.	2
33	Основные элементы языков программирования	Методы программирования	Основные подходы к разработке программ.	2
34	Основные элементы языков программирования	Методы программирования	Связь между данными и операциями.	2
35	Основные элементы языков программирования	Методы программирования	Связь между данными и операциями.	2
36	Основные элементы языков программирования	Паттерны проектирования	Паттерны проектирования	2
37	Основные элементы языков программирования	Паттерны проектирования	Паттерны проектирования	2
38	Основные элементы языков программирования	Паттерны проектирования	Паттерны проектирования	2

39	Основные элементы языков программирования	Паттерны проектирования	Паттерны проектирования	2
40	Основные элементы языков программирования	Паттерны проектирования	Паттерны проектирования	2
Итого за семестр:				48
Итого:				80

4.4. Содержание самостоятельной работы

Наименование раздела	Вид самостоятельной работы	Содержание самостоятельной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов
1 семестр			
Развитие языков программирования. Парадигмы программирования	подготовка к зачету	Развитие языков программирования. Парадигмы программирования.	20
Развитие языков программирования. Парадигмы программирования	подготовка к лекциям	Развитие языков программирования. Парадигмы программирования.	10
Развитие языков программирования. Парадигмы программирования	подготовка к практическим занятиям	Развитие языков программирования. Парадигмы программирования.	30
Итого за семестр:			60
2 семестр			
Основные элементы языков программирования	подготовка к практическим занятиям	Основы объектно- ориентированного программирования.	24
Основные элементы языков программирования	подготовка к лекциям	Основы объектно- ориентированного программирования.	10
Основные элементы языков программирования	подготовка к лабораторным работам	Основы объектно- ориентированного программирования.	10
Основные элементы языков программирования	подготовка к экзаменам	Основы объектно- ориентированного программирования.	20
Итого за семестр:			64
Итого:			124

5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)

№ п/п	Библиографическое описание	Ресурс НТБ СамГТУ (ЭБС СамГТУ, IPRbooks и т.д.)
Основная литература		
1	WEB-программирование распределенных информационных систем: учебное пособие / Воронцов Ю.А., Козинец А.В., Московский технический университет связи и информатики: 2017.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 92420	Электронный ресурс
2	Алгоритмизация и программирование: учебное пособие / Тюльпинова Н.В., Вузовское образование: 2019.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 80539	Электронный ресурс
3	Программирование: основы языка C++: учебное пособие / , Санкт-Петербургский государственный университет промышленных технологий и дизайна, сост. Белая Т.И.: 2020.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 102464	Электронный ресурс
Дополнительная литература		
4	Web-программирование. Клиентские технологии: SVG: учебно-методическое пособие / Никулова Г.А., Липецкий государственный педагогический университет имени П.П. Семенова-Тян-Шанского: 2017.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 100895	Электронный ресурс
5	Башкинова, Е.В. Основы информационных технологий и численных методов. Часть 1 : лабораторный практикум / Е. В. Башкинова, А. А. Заусаев, М. А. Романюк; Самар.гос.техн.ун-т, Прикладная математика и информатика.- Самара, 2020.- 66 с.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu elib 4162	Электронный ресурс
6	Башкинова, Е.В. Основы информационных технологий и численных методов. Часть 2 : лабораторный практикум / Е. В. Башкинова, А. А. Заусаев, М. А. Романюк; Самарский государственный технический университет, Прикладная математика и информатика.- Самара, 2020.- 62 с.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu elib 4163	Электронный ресурс

Доступ обучающихся к ЭР НТБ СамГТУ (elib.samgtu.ru) осуществляется посредством электронной информационной образовательной среды университета и сайта НТБ СамГТУ по логину и паролю.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения

При проведении лекционных занятий используется мультимедийное оборудование.

Организовано взаимодействие обучающегося и преподавателя с использованием электронной ин-формационной образовательной среды университета.

№ п/п	Наименование	Производитель	Способ распространения
1	Microsoft Office	Microsoft (Зарубежный)	Лицензионное
2	Dev C++ (версия 4.9.9.2, свободное ПО)	Компания Bloodshed Software (Зарубежный)	Свободно распространяемое

3	Образовательная платформа «Юрайт»	ООО «ЭЛЕКТРОННОЕ ИЗДАТЕЛЬСТВО ЮРАЙТ» (Отечественный)	Лицензионное
4	МойОфис Образование	ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ» (Отечественный)	Лицензионное
5	Visual Studio Code (VS Code)	Microsoft (Зарубежный)	Свободно распространяемое

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем

№ п/п	Наименование	Краткое описание	Режим доступа
1	eLIBRARY.ru	http://www.eLIBRARY.ru/	Российские базы данных ограниченного доступа
2	КонсультантПлюс (правовые документы) - доступ с ПК в Медиацентре (ауд. 42)	http://www.consultant.ru/	Российские базы данных ограниченного доступа
3	Электронно-библиотечная система IPRbooks	http://www.iprbookshop.ru/	Российские базы данных ограниченного доступа
4	Электронная библиотека изданий СамГТУ	http://irbis.samgtu.local/cgi-bin/irbis64r_01/cgiirbis_64.exe	Российские базы данных ограниченного доступа

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лекционные занятия

Аудитория для лекционных, семинарских и практических занятий, текущего контроля и промежуточной аттестации (с мультимедийным оборудованием) укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Практические занятия

Аудитория для практических и семинарских занятий, текущего контроля и промежуточной аттестации, укомплектованные специализированной мебелью и техническими средствами обучения (проектор, экран, компьютер/ноутбук), с выходом в сеть Интернет и доступом в электронную информационно-образовательную среду СамГТУ. Аудитория оборудована специализированной мебелью: столы и стулья для обучающихся; стол и стул для преподавателя, доска.

- компьютерные классы (ауд. 101, 102, 201, 401, 404).

Лабораторные занятия

Аудитория для практических и семинарских занятий, текущего контроля и промежуточной аттестации, укомплектованные специализированной мебелью и техническими средствами обучения (проектор, экран, компьютер/ноутбук), с выходом в сеть Интернет и доступом в электронную

информационно-образовательную среду СамГТУ. Аудитория оборудована специализированной мебелью: столы и стулья для обучающихся; стол и стул для преподавателя, доска.

- компьютерные классы (ауд. 101, 102, 201, 401).

Самостоятельная работа

Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде СамГТУ:

- кабинет для текущего контроля и промежуточной аттестации, курсового проектирования, групповых и индивидуальных консультаций ауд. 212;

- кабинет для самостоятельной работы, аудитория 304;

- компьютерные классы (ауд. 101, 102, 111, 201, 401, 404).

9. Методические материалы

Методические рекомендации при работе на лекции

До лекции студент должен просмотреть учебно-методическую и научную литературу по теме лекции с тем, чтобы иметь представление о проблемах, которые будут разбираться в лекции.

Перед началом лекции обучающимся сообщается тема лекции, план, вопросы, подлежащие рассмотрению, доводятся основные литературные источники. Весь учебный материал, сообщаемый преподавателем, должен не просто прослушиваться. Он должен быть активно воспринят, т.е. услышан, осмыслен, понят, зафиксирован на бумаге и закреплен в памяти. Приступая к слушанию нового учебного материала, полезно мысленно установить его связь с ранее изученным. Следя за техникой чтения лекции (акцент на существенном, повышение тона, изменение ритма, пауза и т.п.), необходимо вслед за преподавателем уметь выделять основные категории, законы и определять их содержание, проблемы, предполагать их возможные решения, доказательства и выводы. Осуществляя такую работу, можно значительно облегчить себе понимание учебного материала, его конспектирование и дальнейшее изучение.

Конспектирование лекции позволяет обработать, систематизировать и лучше сохранить полученную информацию с тем, чтобы в будущем можно было восстановить в памяти основные, содержательные моменты. Типичная ошибка, совершаемая обучающимся, дословное конспектирование речи преподавателя. Как правило, при записи «слово в слово» не остается времени на обдумывание, анализ и синтез информации. Отбирая нужную информацию, главные мысли, проблемы, решения и выводы, необходимо сокращать текст, строить его таким образом, чтобы потом можно было легко в нем разобраться. Желательно оставить в рабочих конспектах поля, на которых можно будет делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С окончанием лекции работа над конспектом не может считаться завершенной. Нужно еще восстановить отдельные места, проверить, все ли понятно, уточнить что-то на консультации и т.п. с тем, чтобы конспект мог быть использован в процессе подготовки к практическим занятиям, зачету, экзамену. Конспект лекции – незаменимый учебный документ, необходимый для самостоятельной работы.

Методические рекомендации при подготовке и работе на практическом занятии

Практические занятия по дисциплине проводятся в целях выработки практических умений и приобретения навыков в решении профессиональных задач.

Рекомендуется следующая схема подготовки к практическому занятию:

1. ознакомление с планом практического занятия, который отражает содержание предложенной темы;
2. проработка конспекта лекции;
3. чтение рекомендованной литературы;
4. подготовка ответов на вопросы плана практического занятия;

5. выполнение тестовых заданий, задач и др.

Подготовка обучающегося к практическому занятию производится по вопросам, разработанным для каждой темы практических занятий и (или) лекций. В процессе подготовки к практическим занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы.

Работа студентов во время практического занятия осуществляется на основе заданий, которые выдаются обучающимся в начале или во время занятия. На практических занятиях приветствуется активное участие в обсуждении конкретных ситуаций, способность на основе полученных знаний находить наиболее эффективные решения поставленных проблем, уметь находить полезный дополнительный материал по тематике занятий. Обучающимся необходимо обращать внимание на основные понятия, алгоритмы, определять практическую значимость рассматриваемых вопросов. На практических занятиях обучающиеся должны уметь выполнить расчет по заданным параметрам или выработать определенные решения по обозначенной проблеме. Задания могут быть групповые и индивидуальные. В зависимости от сложности предлагаемых заданий, целей занятия, общей подготовки обучающихся преподаватель может подсказать обучающимся алгоритм решения или первое действие, или указать общее направление рассуждений. Полученные результаты обсуждаются с позиций их адекватности или эффективности в рассмотренной ситуации.

Методические рекомендации по выполнению самостоятельной работы

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению с учетом потребностей и возможностей обучающегося.

Самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Самостоятельная работа реализуется:

- непосредственно в процессе аудиторных занятий;
- на лекциях, практических занятиях;
- в контакте с преподавателем вне рамок расписания;
- на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.;
- в библиотеке, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Эффективным средством осуществления обучающимся самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем.

10. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств представлен в приложении № 1.

**Фонд оценочных средств
по дисциплине
Б1.О.05.02 «Языки и методы программирования»**

Код и направление подготовки (специальность)	09.03.01 Информатика и вычислительная техника
Направленность (профиль)	Информатика и вычислительная техника в нефтехимическом производстве
Квалификация	Бакалавр
Форма обучения	Очная
Год начала подготовки	2026
Институт / факультет	Кафедры филиала ФГБОУ ВО "СамГТУ" в г. Новокуйбышевске
Выпускающая кафедра	кафедра "Информатика и системы управления" (НФ-ИиСУ)
Кафедра-разработчик	кафедра "Информатика и системы управления" (НФ-ИиСУ)
Объем дисциплины, ч. / з.е.	324 / 9
Форма контроля (промежуточная аттестация)	Экзамен

**Перечень планируемых результатов обучения по дисциплине (модулю),
соотнесенных с планируемыми результатами освоения образовательной
программы**

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
Общепрофессиональные компетенции			
	ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Владеть навыками составления алгоритмов и программ, пригодных для практического применения
			Знать методы составления алгоритмов и программ, пригодных для практического применения
			Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения
		ОПК-8.2 . Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса
			Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса
			Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса

Матрица соответствия оценочных средств запланированным результатам обучения

Код индикатора достижения компетенции	Результаты обучения	Оценочные средства	Текущий контроль успеваемости	Промежуточная аттестация
---------------------------------------	---------------------	--------------------	-------------------------------	--------------------------

Развитие языков программирования. Парадигмы программирования				
ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Знать методы составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Знать методы составления алгоритмов и программ, пригодных для практического применения	тест	Да	Нет
	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	практические задачи	Да	Нет
	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
ОПК-8.2 . Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	тест	Да	Нет
	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
Основные элементы языков программирования				
ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да

	Знать методы составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
		тест	Да	Нет
	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
ОПК-8.2 . Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	практические задачи	Да	Нет
	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	тест	Да	Нет
		оценочные средства промежуточного контроля	Нет	Да
	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да

Типовые задания для промежуточной аттестации по дисциплине
Б1.О.05.02 Языки и методы программирования
(шифр и наименование дисциплины)

для направления подготовки 09.03.01 Информатика и вычислительная техника в нефтехимическом производстве

(шифр и наименование направления подготовки, специальности)

2026 ГОД ПРИЕМА

(год приема на образовательную программу)

Контролируемая (ые) компетенция(и):

ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения

(шифр и наименование компетенции(й))

Спецификация тестовых заданий

Содержание дисциплины (разделы / темы)	Число заданий									Всего
	закрытые			открытые				комбинированные		
	однозначный выбор варианта ответа	многозначный выбор варианта ответа	задание на сопоставление	задание на установление правильной последовательности	задания на дополнение	задания с развернутым ответом	практико-ориентированные задания	Задания с выбором одного ответа и обоснованием выбора ответа	Задания с выбором нескольких ответов и обоснованием выбора ответов	
Раздел 1. Развитие языков программирования. Парадигмы программирования	4	4	4	4	4	4	0	0	0	24
Тема 1. История и эволюция методов и языков программирования. Синтаксис, семантика, прагматика.	2	2	2	2	2	2	0	0	0	12
Тема 2. Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	2	2	2	2	2	2	0	0	0	12
Раздел 2. Основные элементы языков программирования	12	12	12	12	12	12	0	0	0	72
Тема 3. Основные элементы языков программирования	2	2	2	2	2	2	0	0	0	12
Тема 4. Стандартные структуры данных	2	2	2	2	2	2	0	0	0	12
Тема 5. Основные положения объектного подхода к разработке программ.	2	2	2	2	2	2	0	0	0	12
Тема 6. Обзор методов проектирования и разработки программного обеспечения.	2	2	2	2	2	2	0	0	0	12

Тема 7. Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования	2	2	2	2	2	2	0	0	0	12
Тема 8. Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования	2	2	2	2	2	2	0	0	0	12
Всего	16	16	16	16	16	16	0	0	0	96

Количество заданий в комплекте оценочных материалов

Код компетенции	Наименование компетенции	Количество заданий
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	96

Сценарии выполнения диагностических заданий

Тип задания	Последовательность действий при выполнении задания
Задание закрытого типа с однозначным выбором варианта ответа	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания. 2. Выбрать единственный вариант ответа из предложенных.
Задание закрытого типа с многозначным выбором вариантов ответа	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания. 2. Выбрать несколько вариантов ответа из предложенных.
Задание закрытого типа на установление соответствия	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 - вопросы, утверждения, факты, понятия и т.д.; список 2 - утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать буквы вариантов ответа (например, АБВГ)
Задание закрытого типа на установление последовательности	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты ответа. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы вариантов ответа в нужной последовательности без пробелов и знаков препинания (например, БВА)
Задание открытого типа на дополнение	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается недостающее дополнение. 2. Определить какой информации не хватает. 3. Внесение пропущенного слова. 4. Записать в ответ только дополнение.
Задание открытого типа с развернутым ответом	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять суть вопроса. 2. Продумать логику и полноту ответа. 3. Записать ответ, используя четкие компактные формулировки. 4. В случае расчетной задачи записать решение и ответ.
Задание комбинированного типа: практико-ориентированные задания	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания. 2. Выполните указанные в задания действия
Задание комбинированного типа с выбором одного ответа и обоснованием выбора ответа	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один ответ, наиболее верный. 4. Записать только букву выбранного варианта ответа. 5. Записать аргументы, обосновывающие выбор ответа
Задание комбинированного типа с выбором нескольких ответов и обоснованием выборов ответов	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать несколько верных вариантов ответов. 4. Записать последовательно буквы выбранных вариантов без пробелов и знаков препинания (например, АБВ). 5. Записать аргументы, обосновывающие выбор каждого из ответов

Система оценивания заданий

Указания по оцениванию	Результат оценивания (баллы, полученные за выполнение задания / характеристика правильности ответа)
------------------------	---

Задание закрытого типа с однозначным выбором варианта ответа считается верным, если правильно определен вариант ответа	За правильный вариант ответа начисляется 1 балл
Задание закрытого типа с многозначным выбором вариантов ответа считается верным, если правильно определены все варианты ответа	За правильный вариант ответа начисляется 1 балл
Задание закрытого типа на установление соответствия считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)	Количество баллов определяется числом пар для сопоставления. За каждое правильно установленное соответствие начисляется 1 балл.
Задание закрытого типа на установление последовательности считается верным, если правильно указана вся последовательность цифр	Максимальный балл определяется количеством элементов в последовательности. В случае ошибки в одном месте - снижение на один балл. За каждое правильно указанное место элемента в последовательности начисляется 1 балл.
Задание открытого типа на дополнение, где предоставляется предложение или фрагмент текста, в котором пропущено одно или несколько слов или фраз. Задача состоит в том, чтобы заполнить пропуски, восстановив тем самым исходный смысл предложения.	2 балла засчитывается, если студент вписал правильный ответ в соответствии с ключом. 1 балл может быть засчитан за близкий к правильному ответ, если он демонстрирует частичное понимание.
Задание открытого типа с развернутым ответом считается верным, если ответ совпадает с эталонным по содержанию и полноте	Максимальный балл - 4. Студент может получить 4 балла за полный и правильный ответ, логично изложенный и с корректной терминологией, или меньше за неполные или неточно сформулированные ответы. Полнота (1 балл), Правильность (1 балл), Логичность (1 балл), Терминология (1 балл).
Задание комбинированного типа с выбором одного ответа и обоснованием выбора ответа считается верным, если правильно указана цифра и приведены корректные аргументы, используемые при выборе ответа	За правильный выбор ответа начисляется 1 балл. За качественное обоснование - еще 2-3 балла. Критерии оценивания обоснования должны быть четко определены (например, логичность, полнота, использование фактов). Неправильный выбор ответа - 0 баллов, даже если обоснование частично верное.
Задание комбинированного типа с выбором нескольких вариантов ответа и обоснованием выбора ответа считается верным, если правильно указана цифра и приведены корректные аргументы, используемые при выборе ответа	За правильный выбор ответа начисляется 1 балл. За качественное обоснование - еще 2-3 балла. Критерии оценивания обоснования должны быть четко определены (например, логичность, полнота, использование фактов). Неправильный выбор ответа - 0 баллов, даже если обоснование частично верное.

Тестовые задания с ключами ответов

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Темы
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения						
1	Кто написал первый алгоритм для вычислительной машины (Аналитической машины), что считается началом программирования?	Ада Лавлейс	Закрытое – однозначный выбор	1	1	1
2	Что описывает синтаксис языка программирования?	Правила, задающие структуру и форму допустимых конструкций	Закрытое – однозначный выбор	1	1	1
3	Выберите все языки высокого уровня, появившиеся в конце 1950-х – начале 1960-х годов: FORTRAN, ALGOL, COBOL, Haskell.	FORTRAN; ALGOL; COBOL	Закрытое – множественный выбор	1	1	1
4	Какие аспекты относятся к прагматике языка	Читаемость кода;	Закрытое –	1	1	1

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполне-ния задания, мин	Уровен ь слож-ности (балл)	№ Тем ы
	программирования? Укажите все верные варианты: читаемость кода; практическая полезность конструкций; грамматика языка; значение конструкций.	Практическая полезность конструкций	множественный выбор			
5	Соотнесите язык программирования с его характеристикой: 1) FORTRAN 2) ALGOL 3) COBOL 4) Ассемблер А) Научные вычисления В) Структурированный язык с блочной структурой С) Язык для бизнес-задач D) Низкоуровневый язык с мнемониками.	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	1
6	Соотнесите термин с определением: 1) Семантика 2) Синтаксис 3) Прагматика 4) Метаязык А) Значение конструкций и их исполнение компилятором В) Правила записи допустимых выражений С) Практический аспект использования конструкций D) Формальное средство описания синтаксиса, например BNF.	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	1
7	Упорядочьте ключевые этапы эволюции языков программирования: А) Алгоритм Ады Лавлейс для Аналитической машины В) Создание языка FORTRAN (1957) С) Разработка ALGOL 60 с блочной структурой D) Появление объектно-ориентированного языка C++.	$A \rightarrow B \rightarrow C \rightarrow D$	Открытое – установление правильной последовательности	2	2	1
8	Расположите этапы использования спецификации языка при создании программы: А) Написание кода в соответствии с синтаксисом В) Анализ и перевод кода согласно семантике компилятором или интерпретатором С) Выполнение программы в подходящей среде D) Оценка поведения программы на практике (прагматика).	$A \rightarrow B \rightarrow C \rightarrow D$	Открытое – установление правильной последовательности	2	2	1
9	Заполните пропуски: «В теории языков программирования _____ определяет правила построения допустимых программ, а _____ описывает их смысл».	синтаксис; семантика	Открытое – дополнение	3	2	1
10	Заполните пропуск: «Первый широко используемый язык высокого уровня для научных	FORTRAN	Открытое – дополнен	3	2	1

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполне-ния задания, мин	Уровен ь слож-ности (балл)	№ Тем ы
	вычислений, разработанный в 1957 году, называется _____».		ие			
11	Объясните различия между синтаксисом, семантикой и прагматикой языка программирования. Приведите примеры, как каждый аспект влияет на разработку программ.	Синтаксис задаёт форму и структуру программных конструкций; семантика определяет их смысл и поведение при исполнении; прагматика описывает, как конструкции применяются на практике для удобства, читаемости и поддерживаемости кода.	Открытое – развернутый ответ	5	3	1
12	Опишите два значимых этапа в эволюции языков программирования и объясните, как они повлияли на практику разработки ПО.	Например, внедрение блочной структуры в ALGOL-60 улучшило структурирован ие программ, а появление объектно-ориентированных языков (C++, Java) позволило использовать наследование и инкапсуляцию для моделирования сложных систем и повторного использования кода.	Открытое – развернутый ответ	5	3	1
13	Как называется парадигма программирования, основанная на последовательных командах, изменяющих состояние программы с использованием переменных и циклов?	Процедурная (императивная) парадигма	Закрытое – однозначный выбор	1	1	2
14	В объектно-ориентированном программировании первостепенное значение уделяется:?	Данным и их инкапсуляции	Закрытое – однозначный выбор	1	1	2
15	Выберите преимущества объектно-ориентированного программирования: безопасность данных, наследование и повторное использование кода, гибкость и абстракция, глобальные переменные.	Безопасность данных; Наследование и повторное использование кода; Гибкость и абстракция	Закрытое – множественный выбор	1	1	2
16	Какие характеристики относятся к	Чистые	Закрытое	1	1	2

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Темы
	функциональному программированию? Укажите все верные варианты: чистые функции; рекурсия вместо циклов; изменяемое общее состояние; референциальная прозрачность; неизменяемость данных.	функции; Рекурсия вместо циклов; Референциальная прозрачность; Неизменяемость данных	– множественный выбор			
17	Соотнесите парадигму и её описание: 1) Событийно-ориентированная 2) Компонентно-ориентированная 3) Реактивная 4) Процедурная А) Выполнение определяется событиями пользователя или системы В) Конструирование из модулей с чёткими интерфейсами С) Описание потоков данных и автоматическое распространение изменений D) Пошаговое изменение состояния программы.	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	2
18	Соотнесите термин с описанием: 1) Объект 2) Класс 3) Метод 4) Свойство А) Конкретный экземпляр с состоянием и поведением В) Шаблон, определяющий структуру и поведение объектов С) Функция, принадлежащая классу D) Переменная, принадлежащая объекту.	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	2
19	Упорядочьте шаги разработки объектно-ориентированного приложения: А) Выделить классы и их отношения В) Определить атрибуты и методы С) Реализовать классы в коде D) Создать объекты и использовать их.	A → B → C → D	Открытое – установление правильной последовательности	2	2	2
20	Расположите действия при создании событийно-ориентированного приложения: А) Определить возможные события В) Назначить обработчики событиям С) Запустить главный цикл ожидания событий D) Выполнять обработчики при наступлении событий.	A → B → C → D	Открытое – установление правильной последовательности	2	2	2
21	Заполните пропуск: «Функции, которые всегда возвращают один и тот же результат для одних и тех же аргументов и не имеют побочных эффектов, называются _____».	чистыми функциями	Открытое – дополнение	3	2	2
22	Заполните пропуск: «Парадигма, в которой выполнение программы	событийно-ориентированным	Открытое –	3	2	2

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Тем ы
	определяется событиями пользователя или системы, называется _____ программированием».		дополнен ие			
23	Раскройте основные принципы объектно-ориентированного программирования и объясните, как инкапсуляция и наследование способствуют повторному использованию кода и безопасности данных.	Инкапсуляция объединяет данные и методы в классе и скрывает внутреннюю реализацию через интерфейс; наследование позволяет создавать новые классы на основе существующих, наследуя свойства и методы, что способствует повторному использованию и упрощает расширение функционально сти.	Открытое – развернут ый ответ	5	3	2
24	Сравните процедурную и функциональную парадигмы: обсудите различия в управлении состоянием, подходе к функции и типичных областях применения.	Процедурное программирова ние изменяет состояние через переменные и использует последователь ные шаги, тогда как функционально е базируется на чистых функциях, избегает изменения состояния и применяет рекурсию; процедурный подход часто используется для системного и прикладного программирова ния, а функциональн ый — для задач, требующих простоты проверки, параллелизма и безопасности.	Открытое – развернут ый ответ	5	3	2

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Тем ы
25	Что такое переменная в программировании?	Именованный контейнер для хранения данных	Закрытое – однозначный выбор	1	1	3
26	Какой из перечисленных типов является примитивным?	Целое число (integer)	Закрытое – однозначный выбор	1	1	3
27	Выберите все арифметические операторы: +, -, *, /, %, >	+; -; *; /; %	Закрытое – множественный выбор	1	1	3
28	Какие конструкции относятся к управляющим структурам? Выберите все правильные варианты: if-else, for, while, определение функции.	if-else; for; while	Закрытое – множественный выбор	1	1	3
29	Соотнесите тип оператора с примером: 1) Арифметический 2) Оператор сравнения 3) Логический 4) Присваивание A) * B) > C) && D) =	1–A; 2–B; 3–C; 4–D	Закрытое – сопоставление	2	2	3
30	Соотнесите тип данных с описанием: 1) Boolean 2) String 3) Float 4) Integer A) Представляет значения true/false B) Последовательность символов C) Число с плавающей точкой D) Целое число	1–A; 2–B; 3–C; 4–D	Закрытое – сопоставление	2	2	3
31	Упорядочьте действия при использовании переменной: A) Объявить переменную B) Присвоить значение C) Использовать переменную в выражении D) Изменить значение переменной	$A \rightarrow B \rightarrow C \rightarrow D$	Открытое – установление правильной последовательности	2	2	3
32	Расположите этапы вычисления выражения в Python: A) Вычисление отдельных значений и литералов B) Выполнение арифметических операций C) Объединение результатов в итоговое значение D) Вывод результата или его использование	$A \rightarrow B \rightarrow C \rightarrow D$	Открытое – установление правильной последовательности	2	2	3
33	Заполните пропуск: «Две основные категории типов данных – это примитивные типы и _____ типы, к которым относятся строки, массивы и объекты».	ссылочные	Открытое – дополнение	3	2	3
34	Заполните пропуск: «В Python оператор для проверки равенства	==	Открытое –	3	2	3

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполне-ния задания, мин	Уровен ь слож-ности (балл)	№ Тем ы
	двух значений обозначается как ».		дополнен ие			
35	Объясните назначение управляющих структур в программировании и опишите различия между последовательностью, выбором и итерацией, приведя примеры.	Управляющие структуры задают порядок выполнения команд: последовательность выполняет инструкции одну за другой; выбор (if/else) выбирает ветвь в зависимости от условия; итерация (for/while) повторяет блок кода, пока условие истинно. Например, цикл for перебирает элементы списка, а конструкция if-else позволяет выполнить разный код в зависимости от сравнения.	Открытое – развернутый ответ	5	3	3
36	Опишите, что такое преобразование типов в программировании, и приведите примеры явного и неявного преобразования.	Преобразование типов — это изменение представления данных из одного типа в другой; явное преобразование выполняется программистом явно (например, int("20") + 5), а неявное выполняется автоматически языком (например, в выражении 5 + 2.0 результат будет float).	Открытое – развернутый ответ	5	3	3
37	Как называется структура данных, которая хранит элементы в порядке «первый пришёл — первый вышел»?	Очередь	Закрытое – однозначный выбор	1	1	4
38	Для чего предназначена хеш-таблица?	Для хранения пар ключ–значение с использованием хеш-функции	Закрытое – однозначный выбор	1	1	4
39	Выберите линейные структуры	массив;	Закрытое	1	1	4

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Тем ы
	данных: массив, связанный список, стек, очередь, дерево, граф.	связанный список; стек; очередь	– множественный выбор			
40	Какие операции характерны для стека? Выберите все правильные варианты: push, pop, извлечение любого элемента, peek (просмотр вершины).	push; pop; peek	Закрытое – множественный выбор	1	1	4
41	Соотнесите структуру данных с областью применения: 1) Связанный список 2) Массив 3) Дерево 4) Граф А) Удобное добавление и удаление элементов (история браузера) В) Хранение однородных элементов в смежных ячейках памяти С) Представление иерархий (файловая система) D) Представление сетевых отношений (социальные сети)	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	4
42	Соотнесите структуру данных с её характеристикой: 1) Стек 2) Очередь 3) Хеш-таблица 4) Дерево А) Последним пришёл – первым вышел В) Первым пришёл – первым вышел С) Использует хеш-функцию для отображения ключей D) Узлы имеют отношения родитель–ребёнок	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	4
43	Упорядочьте действия для обработки задач с использованием очереди: А) Создать пустую очередь В) Добавлять задачи по мере поступления С) Извлекать задачу для обработки D) Продолжать, пока очередь не опустеет	$A \rightarrow B \rightarrow C \rightarrow D$	Открытое – установление правильной последовательности	2	2	4
44	Расположите этапы вставки элемента в бинарное дерево поиска: А) Сравнить новое значение с текущим узлом В) Перейти влево, если значение меньше, или вправо, если больше С) Повторять, пока не найдётся пустое место D) Вставить новый узел на найденную позицию	$A \rightarrow B \rightarrow C \rightarrow D$	Открытое – установление правильной последовательности	2	2	4
45	Заполните пропуск: «В _____ структуре данных последний добавленный элемент удаляется первым».	стеке	Открытое – дополнение	3	2	4
46	Заполните пропуск: «_____ — это структура данных, которая	Массив	Открытое –	3	2	4

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Темы
	хранит элементы одинакового типа в смежных ячейках памяти, обеспечивая быстрый доступ по индексу».		дополнение			
47	Опишите различия между линейными и нелинейными структурами данных и приведите пример ситуации, где предпочтительно использование нелинейной структуры.	Линейные структуры (массивы, списки, стеки, очереди) организуют элементы последовательно; нелинейные (деревья, графы) организуют элементы иерархически или в виде сети. Например, для представления файловой системы удобнее использовать дерево, чтобы быстро перемещаться по уровням.	Открытое – развернутый ответ	5	3	4
48	Объясните принцип работы хеш-таблицы и её преимущества. Приведите примеры применения.	Хеш-таблица использует хеш-функцию для преобразования ключа в индекс массива, что позволяет за постоянное время добавлять и находить элементы; она удобна для реализации словарей, кешей, ассоциативных массивов.	Открытое – развернутый ответ	5	3	4
49	Какой принцип ООП объединяет данные и методы в единый модуль и ограничивает прямой доступ к внутреннему состоянию?	Инкапсуляция	Закрытое – однозначный выбор	1	1	5
50	Какое преимущество даёт наследование в ООП?	Повторное использование кода	Закрытое – однозначный выбор	1	1	5
51	Выберите преимущества инкапсуляции: улучшение безопасности данных; скрытие реализации; облегчение сопровождения; использование только глобальных переменных.	улучшенная безопасность данных; скрытие реализации; облегчение	Закрытое – множественный выбор	1	1	5

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Темы
		сопровождения				
52	Какие утверждения верны для полиморфизма? Выберите все: объект может принимать разные формы; реализуется через перегрузку и переопределение методов; препятствует приведению типов; позволяет функциям обрабатывать данные разных типов.	Объект может принимать разные формы; Реализуется через перегрузку и переопределение методов; Позволяет функциям обрабатывать данные разных типов	Закрытое – множественный выбор	1	1	5
53	Соотнесите принцип ООП с примером: 1) Абстракция 2) Полиморфизм 3) Наследование 4) Инкапсуляция А) Определение интерфейса для платежной системы без раскрытия деталей реализации В) Функция, принимающая объекты разных классов и вызывающая общий метод С) Класс Car наследует свойства от класса Vehicle D) Приватные поля и публичные методы для доступа к ним	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	5
54	Соотнесите понятия с определением: 1) Класс 2) Объект 3) Метод 4) Атрибут А) Шаблон для создания объектов В) Конкретный экземпляр класса С) Функция, принадлежащая классу D) Переменная внутри объекта	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	5
55	Упорядочьте шаги при проектировании ООП-системы: А) Выделить сущности и представить их в виде классов В) Определить отношения наследования и ассоциации С) Реализовать классы с атрибутами и методами D) Создать экземпляры классов и использовать их.	$A \rightarrow B \rightarrow C \rightarrow D$	Открытое – установление правильной последовательности	2	2	5
56	Расположите шаги переопределения метода в классе-наследнике: А) Создать базовый класс с методом В) Создать производный класс и унаследовать базовый С) Определить метод в потомке с тем же именем D) Вызвать метод через ссылку на базовый класс, используя динамическую диспетчеризацию.	$A \rightarrow B \rightarrow C \rightarrow D$	Открытое – установление правильной последовательности	2	2	5
57	Заполните пропуск: «Процесс создания нового класса на основе	наследованием	Открытое –	3	2	5

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Тем ы
	существующего класса с использованием его свойств и методов называется _____».		дополнен ие			
58	Заполните пропуск: «Механизм, позволяющий объекту скрывать своё внутреннее состояние и предоставлять доступ к данным через методы, называется _____».	инкапсуляцией	Открытое – дополнен ие	3	2	5
59	Поясните, что такое полиморфизм в объектно-ориентированном программировании. Опишите различия между полиморфизмом времени компиляции (перегрузка) и времени исполнения (переопределение).	Полиморфизм позволяет объектам разных классов выступать в едином интерфейсе; перегрузка функций и операторов реализует полиморфизм во время компиляции, позволяя одной функции иметь несколько реализаций с разными параметрами, а переопределе ние методов в наследниках обеспечивает динамический полиморфизм во время выполнения.	Открытое – развернут ый ответ	5	3	5
60	Обсудите преимущества и потенциальные недостатки использования наследования в больших программных системах и приведите примеры, когда предпочтительно применять композицию.	Наследование упрощает повторное использование кода и расширение функционально сти, но при глубокой иерархии может приводить к жёсткой связанности и сложному сопровождени ю. В случаях, когда требуется гибкая комбинация функционально стей, предпочтитель нее композиция, при которой	Открытое – развернут ый ответ	5	3	5

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Тем ы
		объект содержит ссылки на другие объекты и делегирует им обязанности.				
61	Какой метод разработки делит проект на небольшие инкременты и фокусируется на адаптивности и обратной связи с заказчиком?	Agile-модель	Закрытое – однозначный выбор	1	1	6
62	В каком методе жизненный цикл программного обеспечения проходит строго последовательно без возврата к предыдущим этапам?	Водопадная модель	Закрытое – однозначный выбор	1	1	6
63	Выберите модели разработки, содержащие элементы итеративности: Agile, Инкрементальная, Спиральная, Водопадная.	Agile; Инкрементальная; Спиральная	Закрытое – множественный выбор	1	1	6
64	Какие утверждения характерны для RAD-модели? Выберите все: повторное использование компонентов; подходит для небольших проектов; запрещает обратную связь пользователя; делает упор на быструю разработку прототипов.	повторное использование компонентов; подходит для небольших проектов; делает упор на быструю разработку прототипов	Закрытое – множественный выбор	1	1	6
65	Соотнесите модель разработки с характеристикой: 1) V-модель 2) Инкрементальная модель 3) Спиральная модель 4) Прототипная модель А) Тестирование проводится параллельно с каждым этапом разработки В) Система собирается по частям, каждая часть добавляет функциональность С) Имеет циклы, совмещающие разработку с анализом рисков D) Создание черновой версии продукта для уточнения требований	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	6
66	Соотнесите модель с типом проекта, для которого она наиболее подходит: 1) Agile 2) Водопад 3) Спираль 4) RAD А) Проекты с быстро меняющимися требованиями В) Проекты с чётко определёнными требованиями С) Большие и рискованные проекты D) Небольшие проекты, требующие быстрого прототипирования	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	6
67	Упорядочьте этапы водопадной	A → B → C →	Открытое	2	2	6

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Тем ы
	модели: А) Анализ требований В) Проектирование С) Реализация D) Тестирование Е) Внедрение и сопровождение	D → E	– установле- ние правильно й последова- тельности			
68	Расположите шаги итерации в Agile-процессе разработки: А) Планирование итерации В) Разработка функций С) Тестирование и интеграция D) Демонстрация и получение обратной связи от заказчика Е) Планирование следующей итерации	A → B → C → D → E	Открытое – установле- ние правильно й последова- тельности	2	2	6
69	Заполните пропуск: «Модель жизненного цикла, в которой тестирование выполняется параллельно с разработкой и фазы расположены в форме буквы V, называется _____».	V-моделью	Открытое – дополнен- ие	3	2	6
70	Заполните пропуск: «В спиральной модели каждая петля (итерация) называется _____ и включает планирование, анализ рисков, реализацию и оценку».	фазой	Открытое – дополнен- ие	3	2	6
71	Объясните, почему компании переходят на Agile-методологии вместо традиционных водопадных моделей. Какие преимущества Agile предоставляет в плане адаптивности и удовлетворения клиента?	Agile позволяет разбивать проект на небольшие итерации, регулярно получать обратную связь, быстро реагировать на изменения требований и выпускать функциональ- ные части продукта, что повышает удовлетворён- ность заказчика и снижает риски.	Открытое – развернут ый ответ	5	3	6
72	Опишите основные различия между итеративной и инкрементальной моделями разработки и приведите примеры, когда каждая модель наиболее уместна.	Итеративная модель предполагает создание версии продукта, которая дорабатывает- ся через повторяющиеся циклы, улучшая существующую функциональ- ность; инкременталь- ная модель	Открытое – развернут ый ответ	5	3	6

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполне-ния задания, мин	Уровен ь слож-ности (балл)	№ Тем ы
		разбивает систему на модули, каждый из которых добавляется к уже работающей системе. Итеративный подход подходит, когда требуется постепенно уточнять требования, а инкрементальн ый — когда можно выделить независимые функциональн ые блоки.				
73	Какой цикл обычно используется для обхода массива, когда известен диапазон индексов?	Цикл for	Закрытое – однозначн ый выбор	1	1	7
74	Какую структуру данных используют для реализации алгоритма поиска в глубину (DFS) на графе?	Стек	Закрытое – однозначн ый выбор	1	1	7
75	Выберите управляющие структуры, используемые для условного ветвления: if-else, switch-case, циклы, объявление переменной.	if-else; switch-case	Закрытое – множеств енный выбор	1	1	7
76	Какие структуры данных лучше всего подходят для реализации поиска в ширину (BFS) в графе? Выберите все подходящие варианты: очередь, массив, стек, хеш-таблица.	очередь	Закрытое – множеств енный выбор	1	1	7
77	Соотнесите алгоритм с оптимальной структурой данных для его реализации: 1) Поиск в ширину 2) Поиск в глубину 3) Алгоритм Дейкстры 4) Поиск в хеш-таблице А) Очередь В) Стек или рекурсия С) Приоритетная очередь (минимальная куча) D) Таблица с хеш-функцией	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставл ение	2	2	7
78	Соотнесите тип управляющей структуры с её определением: 1) Последовательность 2) Выбор 3) Итерация 4) Рекурсия А) Выполнение операторов один за другим В) Выбор альтернативы на основе	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставл ение	2	2	7

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Тем ы
	условия С) Повторение блока кода до тех пор, пока условие истинно D) Вызов функции самой себя для решения подзадачи					
79	Расположите шаги обхода связанного списка циклом: A) Инициализировать указатель на начало списка B) Проверить, не равен ли указатель null C) Обработать текущий элемент D) Перейти к следующему элементу E) Повторять, пока указатель не станет null	$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$	Открытое – установле- ние правильно й последова- тельности	2	2	7
80	Упорядочьте действия при выполнении условного оператора if-else: A) Проверить условие B) Если условие истинно, выполнить «then» часть C) Иначе выполнить «else» часть D) Продолжить выполнение программы	$A \rightarrow B \rightarrow C \rightarrow D$	Открытое – установле- ние правильно й последова- тельности	2	2	7
81	Заполните пропуск: «Алгоритм, который обходит граф по уровням, используя очередь, называется _____».	поиском в ширину (BFS)	Открытое – дополнен- ие	3	2	7
82	Заполните пропуск: «Три базовые управляющие структуры, определяющие поток программы, — это последовательность, выбор и _____».	итерация	Открытое – дополнен- ие	3	2	7
83	Объясните, как выбор структуры данных влияет на эффективность алгоритма. Приведите пример сравнения массива и связанного списка при вставке элементов.	Выбор структуры данных определяет сложность операций: вставка в массив требует сдвига элементов, что имеет линейную сложность, тогда как в связанный список можно вставить элемент за постоянное время, изменив ссылки; поэтому для частых вставок предпочтительнее связанный список.	Открытое – развернут ый ответ	5	3	7
84	Опишите роль управляющих структур при реализации алгоритмов и объясните, как комбинация управляющих структур	Управляющие структуры (последовательность, выбор,	Открытое – развернут ый ответ	5	3	7

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполне-ния задания, мин	Уровен ь слож-ности (балл)	№ Тем ы
	и структур данных способствует написанию понятного и эффективного кода.	итерация, рекурсия) определяют порядок выполнения команд. В сочетании с правильными структурами данных они позволяют реализовать алгоритмы эффективно и понятно: например, цикл for с массивом для обхода, очередь с циклом while для BFS, рекурсия для обхода дерева.				
85	Какую структуру данных обычно используют для хранения результатов подзадач в динамическом программировании?	Массив (таблица)	Закрытое – однозначный выбор	1	1	8
86	Какая структура данных лежит в основе приоритетной очереди, применяемой в алгоритме Дейкстры?	Бинарная куча	Закрытое – однозначный выбор	1	1	8
87	Выберите преимущества рекурсии при обходе деревьев: лаконичность кода; автоматическое использование стека вызовов; риск переполнения стека; экономия памяти.	лаконичность кода; автоматическое использование стека вызовов; риск переполнения стека	Закрытое – множественный выбор	1	1	8
88	Какие управляющие структуры могут использоваться для обхода графа? Выберите все подходящие варианты: циклы; рекурсия; условные операторы; объявление класса.	циклы; рекурсия	Закрытое – множественный выбор	1	1	8
89	Соотнесите алгоритм с управляющей структурой, наиболее подходящей для его реализации: 1) Линейный поиск 2) Бинарный поиск 3) Сортировка слиянием 4) Динамическое программирование А) Итерация (цикл) В) Выбор и итерация/рекурсия С) Рекурсивное разделение и объединение D) Вложенные циклы и запоминание результатов	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставление	2	2	8
90	Соотнесите структуру данных с управляющей конструкцией, чаще всего используемой для её обхода:	1–А; 2–В; 3–С; 4–D	Закрытое – сопоставл	2	2	8

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполнения задания, мин	Уровень сложности (балл)	№ Темы
	1) Массив 2) Связанный список 3) Бинарное дерево 4) Граф A) Цикл for B) Цикл while C) Рекурсия D) Цикл с очередью или стеком		ение			
91	Упорядочьте шаги выполнения двоичного поиска в отсортированном массиве: A) Задать начальный и конечный индексы B) Найти средний индекс C) Сравнить среднее значение с искомым D) Если искомое меньше, сдвинуть верхнюю границу; если больше, сдвинуть нижнюю E) Повторять, пока границы не пересекутся	$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$	Открытое – установле- ние правильно й последова тельности	2	2	8
92	Расположите этапы создания хеш-таблицы: A) Выбрать хеш-функцию B) Инициализировать массив (бакеты) C) Для каждого ключа вычислить хеш и индекс D) Вставить пару ключ–значение, обрабатывая коллизии	$A \rightarrow B \rightarrow C \rightarrow D$	Открытое – установле- ние правильно й последова тельности	2	2	8
93	Заполните пропуск: «_____ список — это структура данных, в которой каждый элемент указывает на следующий, что обеспечивает быструю вставку и удаление, но медленный доступ по индексу».	Связанный	Открытое – дополнен ие	3	2	8
94	Заполните пропуск: «_____ программирование — это метод решения задач путём разбиения их на подзадачи с сохранением результатов, обычно реализуемый с помощью таблиц и рекурсии».	Динамическое	Открытое – дополнен ие	3	2	8
95	Опишите, как работает алгоритм поиска в ширину (BFS) на графе, и объясните, почему для его реализации необходима очередь.	BFS начинает с выбранной вершины, помещает её в очередь, затем последовательно извлекает вершины, посещает их соседей, добавляя неизведанные узлы в очередь; очередь обеспечивает обработку вершин в порядке их обнаружения (FIFO), что гарантирует обход графа по	Открытое – развернут ый ответ	5	3	8

№ Зада-ния	Содержание задания	Ответ на задание	Тип задания	Время выполне-ния задания, мин	Уровен ь слож-ности (балл)	№ Тем ы
96	Обсудите достоинства и недостатки использования итеративных циклов и рекурсии при обработке структур данных, таких как деревья и графы. Приведите примеры, когда лучше использовать каждый подход.	уровням. Рекурсия делает код короче и естественнее при работе с деревьями или графами, но может привести к переполнению стека; итеративные циклы более контролируемы по памяти и подходят, когда необходима большая глубина или оптимизация — например, BFS реализуется циклом с очередью, а DFS может быть реализован рекурсией или стеком.	Открытое – развернутый ответ	5	3	8

Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие процессы формирования компетенций

Характеристика процедуры текущего контроля успеваемости и промежуточной аттестации по дисциплине

Оценивание знаний, умений, навыков и опыта деятельности проводятся на основе сведений, приводимых в матрице соответствия оценочных средств запланированным результатам обучения.

Цель текущего контроля успеваемости и промежуточной аттестации по учебным дисциплинам в семестре – проверка приобретаемых обучающимися знаний, умений, навыков в контексте формирования установленных образовательной программой компетенций в течение семестра.

Шкала оценивания:

«Отлично» – выставляется, если сформированность заявленных образовательных результатов компетенций оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно»: студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций;

«Хорошо» – выставляется, если сформированность заявленных образовательных результатов компетенций оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки

«неудовлетворительно», допускается оценка «удовлетворительно»: обучающийся показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций;

«Удовлетворительно» – выставляется, если сформированность заявленных образовательных результатов компетенций оценивается критериями «удовлетворительно», «хорошо» и «отлично»: обучающийся показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение конкретной практической задачи из числа предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой;

«Неудовлетворительно» – выставляется, если при ответе обучающегося выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины.

Ответы и решения, обучающихся оцениваются по следующим общим критериям: распознавание проблем; определение значимой информации; анализ проблем; аргументированность; использование стратегий; творческий подход; выводы; общая грамотность.

Обучающиеся обязаны сдавать все задания в сроки, установленные преподавателем. Оценка

«Удовлетворительно» по дисциплине, может выставляться и при неполной сформированности компетенций в ходе освоения отдельной учебной дисциплины, если их формирование предполагается продолжить на более поздних этапах обучения, в ходе изучения других учебных дисциплин.

Текущий контроль осуществляется через систему оценки преподавателем всех видов работ обучающихся, предусмотренных рабочей программой дисциплины и учебным планом.

Критерии оценки теста.

Количество верных ответов:

80-100% -оценка «отлично»: обучающийся демонстрирует глубокое знание учебно-программного материала, умение свободно выполнять задания, усвоивший взаимосвязь основных понятий дисциплины; способный самостоятельно приобретать новые знания и умения; способный самостоятельно использовать углубленные знания;

71-85% -оценка «хорошо»: обучающийся демонстрирует полное знание учебно-программного материала, успешно выполняющий предусмотренные программой задания, показывающий систематический характер знаний по дисциплине и способный к их самостоятельному пополнению и обновлению в ходе дальнейшего обучения в вузе и в будущей профессиональной деятельности;

50-70% -оценка «удовлетворительно»: обучающийся обнаруживает знание основного учебного программного материала в объеме, необходимом для дальнейшего обучения, выполняющего задания, предусмотренные программой, допустившим неточности в ответе, но обладающим необходимыми знаниями для их устранения;

менее 50% -оценка «неудовлетворительно»: обучающийся демонстрирует пробелы в знаниях основного учебного программного материала, допускает принципиальные ошибки в выполнении предусмотренных программой заданий.

На этапе промежуточной аттестации используется система оценки успеваемости обучающихся, которая позволяет преподавателю оценить сформированность планируемых результатов обучения, а также уровень освоения материала обучающимися.

Форма оценки знаний: оценка - 5 «отлично»; 4 «хорошо»; 3 «удовлетворительно»; 2 «неудовлетворительно». возможно использовать балльно-рейтинговые оценки.

Основанием для определения оценки на зачете служит уровень освоения обучающимся материала и формирования компетенция, предусмотренных учебным планом.

Успеваемость на зачете определяется оценками: «зачтено»; «не зачтено».

Оценка	Критерии оценивания	Балльно-рейтинговая оценка
«Зачтено»	Обучающийся освоил компетенции дисциплины на 51-100 % и показал хорошие знания изученного учебного материала, логично и последовательно изложил и полностью раскрыл смысл предлагаемого вопроса; продемонстрировал умение применить теоретические знания для решения практической задачи; выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	51-100
«Не зачтено»	Обучающийся освоил компетенции дисциплины менее чем на 51% и при ответе на предлагаемый вопрос выявились существенные пробелы в знаниях учебного материала, неумение с помощью преподавателя получить правильное решение практической задачи; не в полном объеме выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	0- 50

Основанием для определения оценки на экзамене служит уровень освоения обучающимся учебного материала, умение решать практические задачи и формирования компетенция, предусмотренных учебным планом.

Успеваемость на экзамене определяется оценками: «отлично»; «хорошо»; «удовлетворительно»; «не удовлетворительно».

Оценка	Критерии оценивания	Балльно-рейтинговая оценка
«Отлично»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования на 86-100 %, показал глубокие знания учебного материала, логично и последовательно изложил содержание ответов на вопросы билета; продемонстрировал умение иллюстрировать теоретические положения конкретными примерами и свободно выполнять экзаменационные задания; усвоил основную и ознакомился с дополнительной литературой; выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	86-100
«Хорошо»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования на 61-85 %, показал глубокие знания учебного материала, логично и последовательно изложил содержание ответов на вопросы билета, но допустил несущественные неточности; продемонстрировал умение иллюстрировать теоретические положения конкретными примерами и выполнять экзаменационные задания; усвоил основную и ознакомился с дополнительной литературой; выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	61-85
«Удовлетворительно»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования на 51-60 %, показал знания учебного материала в объеме, необходимом для дальнейшего освоения учебных программ, но допустил погрешности в изложении ответов на вопросы билета и при выполнении экзаменационных заданий; ознакомился с основной литературой, рекомендованной программой; справился с контрольными заданиями, предусмотренными рабочей программой дисциплины	51-60
«Не удовлетворительно»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования менее чем на 51 %, обнаружил пробелы в знаниях учебного материала, допустил принципиальные ошибки в	0-50

	выполнении контрольных заданий, предусмотренных рабочей программой дисциплины	
--	---	--

Интегральная оценка

Критерии	Традиционная оценка	Балльно-рейтинговая оценка
5	5	86 - 100
4	4	61-85
3	3	51-60
2 и 1	2, Незачет	0-50
5, 4, 3	Зачет	51-100