

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Заболотный, Глеб Иванович  
Должность: Директор филиала  
Дата подписания: 04.05.2026 09:36:53  
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. Заболотни

" \_\_\_\_ " \_\_\_\_\_ 20\_\_ г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Б1.О.03.01 «Алгоритмизация и основы программирования»**

<b>Код и направление подготовки (специальность)</b>	13.03.02 Электроэнергетика и электротехника
<b>Направленность (профиль)</b>	Электроэнергетика
<b>Квалификация</b>	Бакалавр
<b>Форма обучения</b>	Очная
<b>Год начала подготовки</b>	2026
<b>Институт / факультет</b>	Кафедры филиала ФГБОУ ВО "СамГТУ" в г. Новокуйбышевске
<b>Выпускающая кафедра</b>	кафедра "Электроэнергетика, электротехника и автоматизация технологических процессов" (НФ- ЭЭиАТП)
<b>Кафедра-разработчик</b>	кафедра "Информатика и системы управления" (НФ-ИиСУ)
<b>Объем дисциплины, ч. / з.е.</b>	144 / 4
<b>Форма контроля (промежуточная аттестация)</b>	Зачет

### **Б1.О.03.01 «Алгоритмизация и основы программирования»**

Рабочая программа дисциплины разработана в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) **13.03.02 Электроэнергетика и электротехника**, утвержденного приказом Министерства образования и науки РФ от № 144 от 28.02.2018 и соответствующего учебного плана.

Разработчик РПД:

А.Н Лада

\_\_\_\_\_  
(должность, степень, ученое звание)

\_\_\_\_\_  
(ФИО)

Заведующий кафедрой

А.В. Волкодаева, кандидат  
экономических наук, доцент

\_\_\_\_\_  
(ФИО, степень, ученое звание)

**СОГЛАСОВАНО:**

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

\_\_\_\_\_  
(ФИО, степень, ученое звание)

Руководитель образовательной  
программы

\_\_\_\_\_  
(ФИО, степень, ученое звание)

Заведующий выпускающей кафедрой

\_\_\_\_\_  
(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 .....	5
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 .....	12
4.4. Содержание самостоятельной работы .....	16
5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю) .....	18
6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения .....	19
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем .....	20
8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю) .....	20
9. Методические материалы .....	21
10. Фонд оценочных средств по дисциплине (модулю) .....	22

**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			
Информационная культура	ОПК-2 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-2.1 Алгоритмизирует решение задач и реализует алгоритмы с использованием программных средств.	Владеть навыками разработки алгоритмом и компьютерных программ
			Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
			Уметь разрабатывать алгоритмы для конкретных задач
		ОПК-2.2 Принимает участие в процессе разработки, отладки и тестирования компьютерных программ, пригодных для практического применения.	Владеть навыками разработки, отладки и тестирования компьютерных программ, пригодных для практического применения
		Знать языки программирования, основы объектно-ориентированного программирования на примере алгоритмического языка	
		Уметь разрабатывать алгоритмы для конкретных задач, работать в среде программирования, реализовывать построенные алгоритмы в виде программ на конкретном языке программирования, выполнять проверку, отладку кода программы.	

## 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-2			Государственная итоговая аттестация: подготовка к процедуре защиты и защита выпускной квалификационной работы; Учебная практика: профилирующая практика

## 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	2 семестр часов / часов в электронной форме
<b>Аудиторная контактная работа (всего),</b> в том числе:	48	48
Лабораторные работы	16	16
Лекции	16	16
Практические занятия	16	16
<b>Самостоятельная работа (всего),</b> в том числе:	96	96
подготовка к практическим занятиям	96	96
<b>Итого: час</b>	144	144
<b>Итого: з.е.</b>	4	4

## 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
1	Основы алгоритмизации и программирования	8	8	8	48	72
2	Основные конструкции языков программирования	8	8	8	48	72
	<b>Итого</b>	16	16	16	96	144

## 4.1 Содержание лекционных занятий

№ занятия	Наименование раздела	Тема лекции	Содержание лекции (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
<b>2 семестр</b>				
1	Основы алгоритмизации и программирования	1. Введение в программирование	<p>Поколения языков программирования: машинные коды (1-е поколение), ассемблер (2-е поколение), высокоуровневые языки (Fortran, C, Pascal - 3-е поколение), специализированные и визуальные среды (4-е поколение). Понятие парадигмы программирования. Основные парадигмы: императивное программирование, декларативное программирование, функциональное программирование, логическое программирование, объектно-ориентированное программирование. Этапы жизненного цикла программы: постановка задачи (техническое задание), алгоритмизация (разработка алгоритма), написание исходного кода, компиляция и/или интерпретация, компоновка (линковка), отладка и тестирование, сопровождение и модификация. Различия между компиляторами и интерпретаторами: компиляция: исходный код, машинный код (зависимость от платформы), интерпретация: построчный анализ и исполнение, гибридные решения. Понятие интегрированной среды разработки и её компоненты.</p>	2
2	Основы алгоритмизации и программирования	1. Введение в программирование	<p>Поколения языков программирования: машинные коды (1-е поколение), ассемблер (2-е поколение), высокоуровневые языки (Fortran, C, Pascal - 3-е поколение), специализированные и визуальные среды (4-е поколение). Понятие парадигмы программирования. Основные парадигмы: императивное программирование, декларативное программирование, функциональное программирование, логическое программирование, объектно-ориентированное программирование. Этапы жизненного цикла программы: постановка задачи (техническое задание), алгоритмизация (разработка алгоритма), написание исходного кода, компиляция и/или интерпретация, компоновка (линковка), отладка и тестирование, сопровождение и модификация. Различия между компиляторами и интерпретаторами: компиляция: исходный код, машинный код (зависимость от платформы), интерпретация: построчный анализ и исполнение, гибридные решения. Понятие интегрированной среды разработки и её компоненты.</p>	2

3	Основы алгоритмизации и программирования	2. Основы алгоритмизации в программировании	<p>Понятие алгоритма и его свойства: дискретность (разбивка на шаги), детерминированность (однозначность), результативность (завершаемость), массовость (применимость к классу задач). Блок-схемы (ГОСТ): терминатор (овал) - начало и конец, процесс (прямоугольник) - действие, решение (ромб) - условие, ветвление, ввод/вывод (параллелограмм), предопределённый процесс - вызов подпрограммы, стрелки - поток управления. Псевдокод: смесь естественного языка и управляющих конструкций, отсутствие строгого синтаксиса, удобство для передачи идеи алгоритма без привязки к языку. Сравнение способов описания. Оценка сложности алгоритмов (сравнение эффективности независимо от скорости компьютера и языка). Временная сложность (количество элементарных операций). Пространственная сложность (объём используемой памяти). Асимптотический анализ: рост времени работы при увеличении входных данных <math>N</math>. Основные классы сложности (<math>O</math>-большое).</p>	2
4	Основы алгоритмизации и программирования	2. Основы алгоритмизации в программировании	<p>Понятие алгоритма и его свойства: дискретность (разбивка на шаги), детерминированность (однозначность), результативность (завершаемость), массовость (применимость к классу задач). Блок-схемы (ГОСТ): терминатор (овал) - начало и конец, процесс (прямоугольник) - действие, решение (ромб) - условие, ветвление, ввод/вывод (параллелограмм), предопределённый процесс - вызов подпрограммы, стрелки - поток управления. Псевдокод: смесь естественного языка и управляющих конструкций, отсутствие строгого синтаксиса, удобство для передачи идеи алгоритма без привязки к языку. Сравнение способов описания. Оценка сложности алгоритмов (сравнение эффективности независимо от скорости компьютера и языка). Временная сложность (количество элементарных операций). Пространственная сложность (объём используемой памяти). Асимптотический анализ: рост времени работы при увеличении входных данных <math>N</math>. Основные классы сложности (<math>O</math>-большое).</p>	2

5	Основные конструкции языков программирования	3. Конструкции языков программирования и ведение в структурированное программирование	Понятие переменной и константы. Простые типы данных. Литералы. Статическая типизация. Динамическая типизация. Явное и неявное приведение типов. Базовые управляющие конструкции: ветвления и циклы. Понятие подпрограммы. Различия между функцией и процедурой. Формальные и фактические параметры. Способы передачи параметров. Области видимости переменных. Стек вызовов и возврат из подпрограммы. Структуры данных: массивы, строки, записи.	2
6	Основные конструкции языков программирования	3. Конструкции языков программирования и ведение в структурированное программирование	Понятие переменной и константы. Простые типы данных. Литералы. Статическая типизация. Динамическая типизация. Явное и неявное приведение типов. Базовые управляющие конструкции: ветвления и циклы. Понятие подпрограммы. Различия между функцией и процедурой. Формальные и фактические параметры. Способы передачи параметров. Области видимости переменных. Стек вызовов и возврат из подпрограммы. Структуры данных: массивы, строки, записи.	2
7	Основные конструкции языков программирования	4. Введение в объектно-ориентированное программирование	Понятие класса и объекта: класс - чертёж, описание (тип данных), объект - экземпляр класса (конкретный представитель). Поля (свойства) и методы класса. Конструкторы и деструкторы: конструктор - инициализация объекта при создании, деструктор - освобождение ресурсов (при необходимости). Инкапсуляция: сокрытие внутренней реализации, модификаторы доступа. Геттеры и сеттеры (методы доступа к закрытым полям). Наследование. Базовый (родительский) и производный (дочерний) класс. Расширение функциональности: добавление новых полей/методов. Переопределение методов родителя. Иерархия классов. Полиморфизм. Один интерфейс - разная реализация. Перегрузка методов (overload) - одинаковое имя, разные параметры. Переопределение методов (override) - изменение поведения в наследнике. Абстрактные классы и интерфейсы (базовое введение).	2

8	Основные конструкции языков программирования	4. Введение в объектно-ориентированное программирование	<p>Понятие класса и объекта: класс - чертёж, описание (тип данных), объект - экземпляр класса (конкретный представитель). Поля (свойства) и методы класса. Конструкторы и деструкторы: конструктор - инициализация объекта при создании, деструктор - освобождение ресурсов (при необходимости). Инкапсуляция: сокрытие внутренней реализации, модификаторы доступа. Геттеры и сеттеры (методы доступа к закрытым полям). Наследование. Базовый (родительский) и производный (дочерний) класс. Расширение функциональности: добавление новых полей/методов. Переопределение методов родителя. Иерархия классов. Полиморфизм. Один интерфейс - разная реализация. Перегрузка методов (overload) - одинаковое имя, разные параметры. Переопределение методов (override) - изменение поведения в наследнике. Абстрактные классы и интерфейсы (базовое введение).</p>	2
<b>Итого за семестр:</b>				<b>16</b>
<b>Итого:</b>				<b>16</b>

## 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
<b>2 семестр</b>				
1	Основы алгоритмизации и программирования	1. Введение в программирование	<p>Поколения языков программирования: машинные коды (1-е поколение), ассемблер (2-е поколение), высокоуровневые языки (Fortran, C, Pascal - 3-е поколение), специализированные и визуальные среды (4-е поколение). Понятие парадигмы программирования. Основные парадигмы: императивное программирование, декларативное программирование, функциональное программирование, логическое программирование, объектно-ориентированное программирование. Этапы жизненного цикла программы: постановка задачи (техническое задание), алгоритмизация (разработка алгоритма), написание исходного кода, компиляция и/или интерпретация, компоновка (линковка), отладка и тестирование, сопровождение и модификация. Различия между компиляторами и интерпретаторами: компиляция: исходный код, машинный код (зависимость от платформы), интерпретация: построчный анализ и исполнение, гибридные решения. Понятие интегрированной среды разработки и её компоненты.</p>	2

2	Основы алгоритмизации и программирования	1. Введение в программирование	<p>Поколения языков программирования: машинные коды (1-е поколение), ассемблер (2-е поколение), высокоуровневые языки (Fortran, C, Pascal - 3-е поколение), специализированные и визуальные среды (4-е поколение). Понятие парадигмы программирования. Основные парадигмы: императивное программирование, декларативное программирование, функциональное программирование, логическое программирование, объектно-ориентированное программирование. Этапы жизненного цикла программы: постановка задачи (техническое задание), алгоритмизация (разработка алгоритма), написание исходного кода, компиляция и/или интерпретация, компоновка (линковка), отладка и тестирование, сопровождение и модификация. Различия между компиляторами и интерпретаторами: компиляция: исходный код, машинный код (зависимость от платформы), интерпретация: построчный анализ и исполнение, гибридные решения. Понятие интегрированной среды разработки и её компоненты.</p>	2
3	Основы алгоритмизации и программирования	2. Основы алгоритмизации в программировании	<p>Понятие алгоритма и его свойства: дискретность (разбивка на шаги), детерминированность (однозначность), результативность (завершаемость), массовость (применимость к классу задач). Блок-схемы (ГОСТ): терминатор (овал) - начало и конец, процесс (прямоугольник) - действие, решение (ромб) - условие, ветвление, ввод/вывод (параллелограмм), предопределённый процесс - вызов подпрограммы, стрелки - поток управления. Псевдокод: смесь естественного языка и управляющих конструкций, отсутствие строгого синтаксиса, удобство для передачи идеи алгоритма без привязки к языку. Сравнение способов описания. Оценка сложности алгоритмов (сравнение эффективности независимо от скорости компьютера и языка). Временная сложность (количество элементарных операций). Пространственная сложность (объём используемой памяти). Асимптотический анализ: рост времени работы при увеличении входных данных N. Основные классы сложности (O-большое).</p>	2

4	Основы алгоритмизации и программирования	2. Основы алгоритмизации в программировании	<p>Понятие алгоритма и его свойства: дискретность (разбивка на шаги), детерминированность (однозначность), результативность (завершаемость), массовость (применимость к классу задач). Блок-схемы (ГОСТ): терминатор (овал) - начало и конец, процесс (прямоугольник) - действие, решение (ромб) - условие, ветвление, ввод/вывод (параллелограмм), предопределённый процесс - вызов подпрограммы, стрелки - поток управления. Псевдокод: смесь естественного языка и управляющих конструкций, отсутствие строгого синтаксиса, удобство для передачи идеи алгоритма без привязки к языку. Сравнение способов описания. Оценка сложности алгоритмов (сравнение эффективности независимо от скорости компьютера и языка). Временная сложность (количество элементарных операций). Пространственная сложность (объём используемой памяти). Асимптотический анализ: рост времени работы при увеличении входных данных N. Основные классы сложности (O-большое).</p>	2
5	Основные конструкции языков программирования	3. Конструкции языков программирования и ведение в структурированное программирование	<p>Понятие переменной и константы. Простые типы данных. Литералы. Статическая типизация. Динамическая типизация. Явное и неявное приведение типов. Базовые управляющие конструкции: ветвления и циклы. Понятие подпрограммы. Различия между функцией и процедурой. Формальные и фактические параметры. Способы передачи параметров. Области видимости переменных. Стек вызовов и возврат из подпрограммы. Структуры данных: массивы, строки, записи.</p>	2
6	Основные конструкции языков программирования	3. Конструкции языков программирования и ведение в структурированное программирование	<p>Понятие переменной и константы. Простые типы данных. Литералы. Статическая типизация. Динамическая типизация. Явное и неявное приведение типов. Базовые управляющие конструкции: ветвления и циклы. Понятие подпрограммы. Различия между функцией и процедурой. Формальные и фактические параметры. Способы передачи параметров. Области видимости переменных. Стек вызовов и возврат из подпрограммы. Структуры данных: массивы, строки, записи.</p>	2

7	Основные конструкции языков программирования	4. Введение в объектно-ориентированное программирование	Понятие класса и объекта: класс - чертёж, описание (тип данных), объект - экземпляр класса (конкретный представитель). Поля (свойства) и методы класса. Конструкторы и деструкторы: конструктор - инициализация объекта при создании, деструктор - освобождение ресурсов (при необходимости). Инкапсуляция: сокрытие внутренней реализации, модификаторы доступа. Геттеры и сеттеры (методы доступа к закрытым полям). Наследование. Базовый (родительский) и производный (дочерний) класс. Расширение функциональности: добавление новых полей/методов. Переопределение методов родителя. Иерархия классов. Полиморфизм. Один интерфейс - разная реализация. Перегрузка методов (overload) - одинаковое имя, разные параметры. Переопределение методов (override) - изменение поведения в наследнике. Абстрактные классы и интерфейсы (базовое введение).	2
8	Основные конструкции языков программирования	4. Введение в объектно-ориентированное программирование	Понятие класса и объекта: класс - чертёж, описание (тип данных), объект - экземпляр класса (конкретный представитель). Поля (свойства) и методы класса. Конструкторы и деструкторы: конструктор - инициализация объекта при создании, деструктор - освобождение ресурсов (при необходимости). Инкапсуляция: сокрытие внутренней реализации, модификаторы доступа. Геттеры и сеттеры (методы доступа к закрытым полям). Наследование. Базовый (родительский) и производный (дочерний) класс. Расширение функциональности: добавление новых полей/методов. Переопределение методов родителя. Иерархия классов. Полиморфизм. Один интерфейс - разная реализация. Перегрузка методов (overload) - одинаковое имя, разные параметры. Переопределение методов (override) - изменение поведения в наследнике. Абстрактные классы и интерфейсы (базовое введение).	2
<b>Итого за семестр:</b>				<b>16</b>
<b>Итого:</b>				<b>16</b>

### 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
<b>2 семестр</b>				

1	Основы алгоритмизации и программирования	1. Введение в программирование	<p>Поколения языков программирования: машинные коды (1-е поколение), ассемблер (2-е поколение), высокоуровневые языки (Fortran, C, Pascal - 3-е поколение), специализированные и визуальные среды (4-е поколение). Понятие парадигмы программирования. Основные парадигмы: императивное программирование, декларативное программирование, функциональное программирование, логическое программирование, объектно-ориентированное программирование. Этапы жизненного цикла программы: постановка задачи (техническое задание), алгоритмизация (разработка алгоритма), написание исходного кода, компиляция и/или интерпретация, компоновка (линковка), отладка и тестирование, сопровождение и модификация. Различия между компиляторами и интерпретаторами: компиляция: исходный код, машинный код (зависимость от платформы), интерпретация: построчный анализ и исполнение, гибридные решения. Понятие интегрированной среды разработки и её компоненты.</p>	2
2	Основы алгоритмизации и программирования	1. Введение в программирование	<p>Поколения языков программирования: машинные коды (1-е поколение), ассемблер (2-е поколение), высокоуровневые языки (Fortran, C, Pascal - 3-е поколение), специализированные и визуальные среды (4-е поколение). Понятие парадигмы программирования. Основные парадигмы: императивное программирование, декларативное программирование, функциональное программирование, логическое программирование, объектно-ориентированное программирование. Этапы жизненного цикла программы: постановка задачи (техническое задание), алгоритмизация (разработка алгоритма), написание исходного кода, компиляция и/или интерпретация, компоновка (линковка), отладка и тестирование, сопровождение и модификация. Различия между компиляторами и интерпретаторами: компиляция: исходный код, машинный код (зависимость от платформы), интерпретация: построчный анализ и исполнение, гибридные решения. Понятие интегрированной среды разработки и её компоненты.</p>	2

3	Основы алгоритмизации и программирования	2. Основы алгоритмизации в программировании	<p>Понятие алгоритма и его свойства: дискретность (разбивка на шаги), детерминированность (однозначность), результативность (завершаемость), массовость (применимость к классу задач). Блок-схемы (ГОСТ): терминатор (овал) - начало и конец, процесс (прямоугольник) - действие, решение (ромб) - условие, ветвление, ввод/вывод (параллелограмм), предопределённый процесс - вызов подпрограммы, стрелки - поток управления. Псевдокод: смесь естественного языка и управляющих конструкций, отсутствие строгого синтаксиса, удобство для передачи идеи алгоритма без привязки к языку. Сравнение способов описания. Оценка сложности алгоритмов (сравнение эффективности независимо от скорости компьютера и языка). Временная сложность (количество элементарных операций). Пространственная сложность (объём используемой памяти). Асимптотический анализ: рост времени работы при увеличении входных данных N. Основные классы сложности (O-большое).</p>	2
4	Основы алгоритмизации и программирования	2. Основы алгоритмизации в программировании	<p>Понятие алгоритма и его свойства: дискретность (разбивка на шаги), детерминированность (однозначность), результативность (завершаемость), массовость (применимость к классу задач). Блок-схемы (ГОСТ): терминатор (овал) - начало и конец, процесс (прямоугольник) - действие, решение (ромб) - условие, ветвление, ввод/вывод (параллелограмм), предопределённый процесс - вызов подпрограммы, стрелки - поток управления. Псевдокод: смесь естественного языка и управляющих конструкций, отсутствие строгого синтаксиса, удобство для передачи идеи алгоритма без привязки к языку. Сравнение способов описания. Оценка сложности алгоритмов (сравнение эффективности независимо от скорости компьютера и языка). Временная сложность (количество элементарных операций). Пространственная сложность (объём используемой памяти). Асимптотический анализ: рост времени работы при увеличении входных данных N. Основные классы сложности (O-большое).</p>	2

5	Основные конструкции языков программирования	3. Конструкции языков программирования и ведение в структурированное программирование	Понятие переменной и константы. Простые типы данных. Литералы. Статическая типизация. Динамическая типизация. Явное и неявное приведение типов. Базовые управляющие конструкции: ветвления и циклы. Понятие подпрограммы. Различия между функцией и процедурой. Формальные и фактические параметры. Способы передачи параметров. Области видимости переменных. Стек вызовов и возврат из подпрограммы. Структуры данных: массивы, строки, записи.	2
6	Основные конструкции языков программирования	3. Конструкции языков программирования и ведение в структурированное программирование	Понятие переменной и константы. Простые типы данных. Литералы. Статическая типизация. Динамическая типизация. Явное и неявное приведение типов. Базовые управляющие конструкции: ветвления и циклы. Понятие подпрограммы. Различия между функцией и процедурой. Формальные и фактические параметры. Способы передачи параметров. Области видимости переменных. Стек вызовов и возврат из подпрограммы. Структуры данных: массивы, строки, записи.	2
7	Основные конструкции языков программирования	4. Введение в объектно-ориентированное программирование	Понятие класса и объекта: класс - чертёж, описание (тип данных), объект - экземпляр класса (конкретный представитель). Поля (свойства) и методы класса. Конструкторы и деструкторы: конструктор - инициализация объекта при создании, деструктор - освобождение ресурсов (при необходимости). Инкапсуляция: сокрытие внутренней реализации, модификаторы доступа. Геттеры и сеттеры (методы доступа к закрытым полям). Наследование. Базовый (родительский) и производный (дочерний) класс. Расширение функциональности: добавление новых полей/методов. Переопределение методов родителя. Иерархия классов. Полиморфизм. Один интерфейс - разная реализация. Перегрузка методов (overload) - одинаковое имя, разные параметры. Переопределение методов (override) - изменение поведения в наследнике. Абстрактные классы и интерфейсы (базовое введение).	2

8	Основные конструкции языков программирования	4. Введение в объектно-ориентированное программирование	Понятие класса и объекта: класс - чертёж, описание (тип данных), объект - экземпляр класса (конкретный представитель). Поля (свойства) и методы класса. Конструкторы и деструкторы: конструктор - инициализация объекта при создании, деструктор - освобождение ресурсов (при необходимости). Инкапсуляция: сокрытие внутренней реализации, модификаторы доступа. Геттеры и сеттеры (методы доступа к закрытым полям). Наследование. Базовый (родительский) и производный (дочерний) класс. Расширение функциональности: добавление новых полей/методов. Переопределение методов родителя. Иерархия классов. Полиморфизм. Один интерфейс - разная реализация. Перегрузка методов (overload) - одинаковое имя, разные параметры. Переопределение методов (override) - изменение поведения в наследнике. Абстрактные классы и интерфейсы (базовое введение).	2
			<b>Итого за семестр:</b>	<b>16</b>
			<b>Итого:</b>	<b>16</b>

#### 4.4. Содержание самостоятельной работы

Наименование раздела	Вид самостоятельной работы	Содержание самостоятельной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов
<b>2 семестр</b>			

<p>Основы алгоритмизации и программирования</p>	<p>Подготовка к практическим занятиям</p>	<p>Поколения языков программирования: машинные коды (1-е поколение), ассемблер (2-е поколение), высокоуровневые языки (Fortran, C, Pascal - 3-е поколение), специализированные и визуальные среды (4-е поколение). Понятие парадигмы программирования. Основные парадигмы: императивное программирование, декларативное программирование, функциональное программирование, логическое программирование, объектно-ориентированное программирование. Этапы жизненного цикла программы: постановка задачи (техническое задание), алгоритмизация (разработка алгоритма), написание исходного кода, компиляция и/или интерпретация, компоновка (линковка), отладка и тестирование, сопровождение и модификация. Различия между компиляторами и интерпретаторами: компиляция: исходный код, машинный код (зависимость от платформы), интерпретация: построчный анализ и исполнение, гибридные решения. Понятие интегрированной среды разработки и её компоненты. Понятие алгоритма и его свойства: дискретность (разбивка на шаги), детерминированность (однозначность), результативность (завершаемость), массовость (применимость к классу задач). Блок-схемы (ГОСТ): терминатор (овал) - начало и конец, процесс (прямоугольник) - действие, решение (ромб) - условие, ветвление, ввод/вывод (параллелограмм), предопределённый процесс - вызов подпрограммы, стрелки - поток управления. Псевдокод: смесь естественного языка и управляющих конструкций, отсутствие строгого синтаксиса, удобство для передачи идеи алгоритма без привязки к языку. Сравнение способов описания. Оценка сложности алгоритмов (сравнение эффективности независимо от скорости компьютера и языка). Временная сложность (количество элементарных операций). Пространственная сложность (объём используемой памяти). Асимптотический анализ: рост времени работы при увеличении входных данных N. Основные классы сложности (O-большое).</p>	<p>48</p>
---	---	---	-----------

Основные конструкции языков программирования	Подготовка к практическим занятиям	<p>Понятие переменной и константы. Простые типы данных. Литералы. Статическая типизация. Динамическая типизация. Явное и неявное приведение типов. Базовые управляющие конструкции: ветвления и циклы. Понятие подпрограммы. Различия между функцией и процедурой. Формальные и фактические параметры. Способы передачи параметров. Области видимости переменных. Стек вызовов и возврат из подпрограммы. Структуры данных: массивы, строки, записи. Понятие класса и объекта: класс - чертёж, описание (тип данных), объект - экземпляр класса (конкретный представитель). Поля (свойства) и методы класса. Конструкторы и деструкторы: конструктор - инициализация объекта при создании, деструктор - освобождение ресурсов (при необходимости). Инкапсуляция: сокрытие внутренней реализации, модификаторы доступа. Геттеры и сеттеры (методы доступа к закрытым полям). Наследование. Базовый (родительский) и производный (дочерний) класс. Расширение функциональности: добавление новых полей/методов. Переопределение методов родителя. Иерархия классов. Полиморфизм. Один интерфейс - разная реализация. Перегрузка методов (overload) - одинаковое имя, разные параметры. Переопределение методов (override) - изменение поведения в наследнике. Абстрактные классы и интерфейсы (базовое введение).</p>	48
<b>Итого за семестр:</b>			<b>96</b>
<b>Итого:</b>			<b>96</b>

### 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	Алгоритмизация и программирование: учебное пособие / , Тюменский индустриальный университет, сост. Сорокин Г.Г.: 2022.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  133639">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  133639</a>	Электронный ресурс

2	Алгоритмизация и программирование: учебное пособие / Дорохова Т.Ю., Ильина И.Е., Ай Пи Ар Медиа: 2022.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  122425">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  122425</a>	Электронный ресурс
3	Алгоритмизация. Технология разработки программного обеспечения: учебно-методическое пособие / Китайцева Е.Х., МИСИ-МГСУ, ЭБС АСВ: 2021.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  126184">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  126184</a>	Электронный ресурс
4	Алгоритмические языки и программирование. В 2 томах. Т.1. Основы алгоритмизации и программирования: учебник / Антипов О.В., Дмитриева Т.А., Москвитина О.А., Парфилова Н.И., КУРС: 2024.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  144767">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  144767</a>	Электронный ресурс
Дополнительная литература		
5	Алгоритмизация и программирование (C++): учебно-методическое пособие / Ремизова О.И., Издательский Дом МИСиС: 2021.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  129485">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  129485</a>	Электронный ресурс
6	Основы алгоритмизации и программирования на языке Pascal: практикум / Исаев А.Л., Профобразование, Ай Пи Ар Медиа: 2023.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  130049">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  130049</a>	Электронный ресурс
7	Основы алгоритмизации и программирования на языке Python: учебник / Андреева О.В., Ремизова О.И., Издательский Дом МИСиС: 2022.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  129510">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  129510</a>	Электронный ресурс
8	Основы алгоритмизации и программирования: учебное пособие / Чурина Т.Г., Нестеренко Т.В., Профобразование, Ай Пи Ар Медиа: 2020.- Режим доступа: <a href="https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  96017">https://elib.samgtu.ru/getinfo?uid=els_samgtu  iprbooks  96017</a>	Электронный ресурс

Доступ обучающихся к ЭР НТБ СамГТУ ([elib.samgtu.ru](http://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 информационной образовательной среды университета.

<b>№ п/п</b>	<b>Наименование</b>	<b>Производитель</b>	<b>Способ распространения</b>
1	Visual Studio Code (VS Code)	Microsoft (Зарубежный)	Свободно распространяемое
2	Microsoft Office	Microsoft (Зарубежный)	Лицензионное
3	Matlab	MathWorks (Зарубежный)	Лицензионное
4	Образовательная платформа «Юрайт»	ООО «ЭЛЕКТРОННОЕ ИЗДАТЕЛЬСТВО ЮРАЙТ» (Отечественный)	Лицензионное

5	МойОфис Образование	ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ» (Отечественный)	Лицензионное
---	---------------------	---	--------------

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

№ п/п	Наименование	Краткое описание	Режим доступа
1	eLIBRARY.ru	<a href="http://www.eLIBRARY.ru/">http://www.eLIBRARY.ru/</a>	Российские базы данных ограниченного доступа
2	Первые шаги в программировании	<a href="http://www.firststeps.ru.">http://www.firststeps.ru.</a>	Ресурсы открытого доступа
3	Сайты научно – технической библиотеки ФГБОУ СамГТУ	<a href="http://lib.sumgtu.ru/">http://lib.sumgtu.ru/</a>	Ресурсы открытого 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 закреплён 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 завершённой. 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 при работе на лабораторном занятии

Проведение лабораторной работы делится на две условные части: теоретическую и практическую.

Необходимыми структурными элементами занятия являются проведение лабораторной работы, проверка усвоенного материала, включающая обсуждение теоретических основ выполняемой работы.

Перед лабораторной работой, как правило, проводится технико-теоретический инструктаж по использованию необходимого оборудования. Преподаватель корректирует деятельность обучающегося в процессе выполнения работы (при необходимости). После завершения лабораторной работы подводятся итоги, обсуждаются результаты деятельности.

Возможны следующие формы организации лабораторных работ: фронтальная, групповая и индивидуальная. При фронтальной форме выполняется одна и та же работа (при этом возможны различные варианты заданий). При групповой форме работа выполняется группой (командой). При индивидуальной форме обучающимися выполняются индивидуальные работы.

По каждой лабораторной работе имеются методические указания по их выполнению, включающие необходимый теоретический и практический материал, содержащие элементы и последовательную инструкцию по проведению выбранной работы, индивидуальные варианты заданий, требования и форму отчётности по данной работе.

## 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.

Приложение 1 к рабочей программе дисциплины  
Б1.О.03.01 «Алгоритмизация и основы  
программирования»

**Фонд оценочных средств  
по дисциплине  
Б1.О.03.01 «Алгоритмизация и основы программирования»**

<b>Код и направление подготовки (специальность)</b>	13.03.02 Электроэнергетика и электротехника
<b>Направленность (профиль)</b>	Электроэнергетика
<b>Квалификация</b>	Бакалавр
<b>Форма обучения</b>	Очная
<b>Год начала подготовки</b>	2026
<b>Институт / факультет</b>	Кафедры филиала ФГБОУ ВО "СамГТУ" в г. Новокуйбышевске
<b>Выпускающая кафедра</b>	кафедра "Электроэнергетика, электротехника и автоматизация технологических процессов" (НФ- ЭЭиАТП)
<b>Кафедра-разработчик</b>	кафедра "Информатика и системы управления" (НФ-ИиСУ)
<b>Объем дисциплины, ч. / з.е.</b>	144 / 4
<b>Форма контроля (промежуточная аттестация)</b>	Зачет

**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			
Информационная культура	ОПК-2 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-2.1 Алгоритмизирует решение задач и реализует алгоритмы с использованием программных средств.	Владеть навыками разработки алгоритмом и компьютерных программ
			Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
			Уметь разрабатывать алгоритмы для конкретных задач
		ОПК-2.2 Принимает участие в процессе разработки, отладки и тестирования компьютерных программ, пригодных для практического применения.	Владеть навыками разработки, отладки и тестирования компьютерных программ, пригодных для практического применения
		Знать языки программирования, основы объектно-ориентированного программирования на примере алгоритмического языка	
		Уметь разрабатывать алгоритмы для конкретных задач, работать в среде программирования, реализовывать построенные алгоритмы в виде программ на конкретном языке программирования, выполнять проверку, отладку кода программы.	

## Матрица соответствия оценочных средств запланированным результатам обучения

Код индикатора достижения компетенции	Результаты обучения	Оценочные средства	Текущий контроль успеваемости	Промежуточная аттестация
<b>Основы алгоритмизации и программирования</b>				
ОПК-2.1 Алгоритмизирует решение задач и реализует алгоритмы с использованием программных средств.	<b>Уметь</b> разрабатывать алгоритмы для конкретных задач	оценочные средства промежуточного контроля	Нет	Да
	<b>Владеть</b> навыками разработки алгоритмом и компьютерных программ	оценочные средства промежуточного контроля	Нет	Да
	<b>Знать</b> понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	оценочные средства промежуточного контроля	Нет	Да
	<b>Уметь</b> разрабатывать алгоритмы для конкретных задач	практические задания	Да	Нет
	<b>Владеть</b> навыками разработки алгоритмом и компьютерных программ	практические задания	Да	Нет
	<b>Знать</b> понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	тест	Да	Нет
ОПК-2.2 Принимает участие в процессе разработки, отладки и тестирования компьютерных программ, пригодных для практического применения.	<b>Знать</b> языки программирования, основы объектно- ориентированного программирования на примере алгоритмического языка	тест	Да	Нет
	<b>Владеть</b> навыками разработки, отладки и тестирования компьютерных программ, пригодных для практического применения	практические задания	Да	Нет
	<b>Уметь</b> разрабатывать алгоритмы для конкретных задач, работать в среде программирования, реализовывать построенные алгоритмы в виде программ на конкретном языке программирования, выполнять проверку, отладку кода программы.	практические задания	Да	Нет
	<b>Знать</b> языки программирования, основы объектно- ориентированного программирования на примере алгоритмического языка	оценочные средства промежуточного контроля	Нет	Да
	<b>Владеть</b> навыками разработки, отладки и тестирования компьютерных программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да

	<b>Уметь</b> разрабатывать алгоритмы для конкретных задач, работать в среде программирования, реализовывать построенные алгоритмы в виде программ на конкретном языке программирования, выполнять проверку, отладку кода программы.	оценочные средства промежуточного контроля	Нет	Да
<b>Основные конструкции языков программирования</b>				
ОПК-2.1 Алгоритмизирует решение задач и реализует алгоритмы с использованием программных средств.	<b>Уметь</b> разрабатывать алгоритмы для конкретных задач	оценочные средства промежуточного контроля	Нет	Да
	<b>Знать</b> понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	оценочные средства промежуточного контроля	Нет	Да
	<b>Владеть</b> навыками разработки алгоритмом и компьютерных программ	оценочные средства промежуточного контроля	Нет	Да
	<b>Уметь</b> разрабатывать алгоритмы для конкретных задач	практические задания	Да	Нет
	<b>Знать</b> понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	тест	Да	Нет
	<b>Владеть</b> навыками разработки алгоритмом и компьютерных программ	практические задания	Да	Нет
ОПК-2.2 Принимает участие в процессе разработки, отладки и тестирования компьютерных программ, пригодных для практического применения.	<b>Уметь</b> разрабатывать алгоритмы для конкретных задач, работать в среде программирования, реализовывать построенные алгоритмы в виде программ на конкретном языке программирования, выполнять проверку, отладку кода программы.	практические задания	Да	Нет
	<b>Знать</b> языки программирования, основы объектно- ориентированного программирования на примере алгоритмического языка	тест	Да	Нет
	<b>Владеть</b> навыками разработки, отладки и тестирования компьютерных программ, пригодных для практического применения	практические задания	Да	Нет
	<b>Уметь</b> разрабатывать алгоритмы для конкретных задач, работать в среде программирования, реализовывать построенные алгоритмы в виде программ на конкретном языке программирования, выполнять проверку, отладку кода программы.	оценочные средства промежуточного контроля	Нет	Да
	<b>Знать</b> языки программирования, основы объектно- ориентированного программирования на примере алгоритмического языка	оценочные средства промежуточного контроля	Нет	Да
	<b>Владеть</b> навыками разработки, отладки и тестирования компьютерных программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да

**Типовые задания для промежуточной аттестации по дисциплине  
Б1.О.03.01 Алгоритмизация и основы программирования**  
(шифр и наименование дисциплины)

**для направления подготовки 13.03.02 Электроэнергетика и электротехника**  
(шифр и наименование направления подготовки, специальности)

**2026 ГОД ПРИЕМА**

(год приема на образовательную программу)

**Контролируемая (ые) компетенция(и):**

**ОПК-2 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения**

(шифр и наименование компетенции(й))

**Спецификация тестовых заданий**

Содержание дисциплины (разделы / темы)	Число заданий									
	закрытые			открытые				комбинированные		всего
	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	
<b>Раздел 1. Основы алгоритмизации и программирования</b>	<b>10</b>	<b>5</b>	<b>8</b>	<b>11</b>	<b>12</b>	<b>7</b>				
Тема 1. Введение в программирование	5	3	4	6	6	3				27
Тема 2. Основы алгоритмизации в программировании	5	2	4	5	6	4				26
<b>Раздел 2. Основные конструкции языков программирования</b>	<b>9</b>	<b>8</b>	<b>6</b>	<b>7</b>	<b>12</b>	<b>5</b>				<b>47</b>
Тема 3. Конструкции языков программирования и ведение в структурированное программирование	4	4	2	4	6	3				23
Тема 4. Введение в объектно-ориентированное программирование	5	4	4	3	6	2				24
<b>Итого</b>	<b>19</b>	<b>13</b>	<b>14</b>	<b>18</b>	<b>24</b>	<b>12</b>				<b>100</b>

**Количество заданий в комплекте оценочных материалов**

Код компетенции	Наименование компетенции	Количество заданий
ОПК-2	ОПК-2 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	100

**Сценарии выполнения диагностических заданий**

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

### 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

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
<b>ОПК-2 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</b>					
1.	<b>Прочитайте и дополните фразу:</b> Языки программирования первого поколения, представляющие собой последовательности двоичных или шестнадцатеричных кодов, которые процессор выполняет непосредственно, называются _____.	машинными кодами	Задание открытого типа на дополнение	2	1
2.	<b>Прочитайте и дополните фразу:</b> Языки программирования четвертого поколения (4GL) включают в себя _____, предназначенные для быстрой разработки приложений с использованием визуальных компонентов.	визуальные среды	Задание открытого типа на дополнение	2	1
3.	<b>Прочитайте и дополните фразу:</b> Процесс преобразования всей программы из исходного кода в машинный код, который затем может быть выполнен независимо от компилятора, называется _____.	компиляцией	Задание открытого типа на дополнение	2	1
4.	<b>Прочитайте и дополните фразу:</b> Совокупность программных средств, используемых для разработки программного обеспечения, объединяющая редактор кода, компилятор/интерпретатор, отладчик и другие инструменты, называется _____.	интегрированной средой разработки	Задание открытого типа на дополнение	2	1
5.	<b>Прочитайте и дополните фразу:</b> Этап жизненного цикла программы, на котором разрабатывается формальный документ, содержащий цели, функции, требования к производительности и условиям эксплуатации будущей программы, называется _____.	постановкой задачи	Задание открытого типа на дополнение	2	1
6.	<b>Прочитайте и дополните фразу:</b> Парадигма программирования, в которой программа представляет	императивным	Задание открытого типа на дополнение	2	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	собой последовательность инструкций, изменяющих состояние памяти компьютера (переменные), называется _____ программированием.				
7.	<b>Прочитайте вопрос и дайте развернутый ответ.</b> Назовите три основных отличия языков программирования третьего поколения (высокоуровневых) от языков первого (машинные коды) и второго (ассемблер) поколений.	1) Абстракция от аппаратуры. 2) Машинная независимость. 3) Производительность труда разработчика.	Задание открытого типа с развернутым ответом	4	1
8.	<b>Прочитайте вопрос и дайте развернутый ответ.</b> Перечислите три основных этапа жизненного цикла программы, связанных с преобразованием кода (после разработки алгоритма).	1) Написание исходного кода. 2) Компиляция. 3) Компоновка (линковка).	Задание открытого типа с развернутым ответом	4	1
9.	<b>Прочитайте вопрос и дайте развернутый ответ.</b> Назовите не менее трёх компонентов интегративной среды разработки.	1) Редактор исходного кода. 2) Компилятор и/или интерпретатор. 3) Отладчик (debugger). 4) Средства автоматизации сборки. 5) Система управления версиями.	Задание открытого типа с развернутым ответом	4	1
10.	<b>Упорядочите поколения языков программирования в порядке их исторического появления (от первого к четвёртому):</b> 1. Высокоуровневые языки (Fortran, C, Pascal). 2. Машинные коды. 3. Ассемблер. 4. Специализированные и визуальные среды (SQL, Delphi). Ответ запишите в виде последовательности цифр через запятую слева направо.	2,3,1,4	Задание закрытого типа на установление последовательности	1	1
11.	<b>Упорядочите этапы жизненного цикла программы, связанные с написанием и исполнением кода, в их логической последовательности:</b> 1. Компоновка (линковка). 2. Написание исходного кода. 3. Компиляция. 4. Исполнение (запуск программы). Ответ запишите в виде последовательности цифр через запятую слева направо.	2,3,1,4	Задание закрытого типа на установление последовательности	1	1
12.	<b>Упорядочите этапы создания программы от возникновения идеи до готового продукта:</b> 1. Отладка и тестирование. 2. Постановка задачи (техническое задание). 3. Алгоритмизация (разработка алгоритма). 4. Написание исходного кода. Ответ запишите в виде	2,3,4,1	Задание закрытого типа на установление последовательности	1	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы												
	последовательности цифр через запятую слева направо.																
13.	<b>Упорядочите виды трансляции от наиболее ранней (исторически) к более современным:</b> 1. Компиляция в байт-код с JIT. 2. Чистая интерпретация. 3. Компиляция в машинный код. Ответ запишите в виде последовательности цифр через запятую слева направо.	3,2,1	Задание закрытого типа на установление последовательности	1	1												
14.	<b>Упорядочите этапы выполнения программы в интерпретируемом языке (например, Python):</b> 1. Интерпретатор читает и анализирует очередную строку исходного кода. 2. Пользователь запускает скрипт (исходный файл). 3. Интерпретатор выполняет соответствующее действие. 4. Переход к следующей строке до конца файла. Ответ запишите в виде последовательности цифр через запятую слева направо.	2,1,3,4	Задание закрытого типа на установление последовательности	1	1												
15.	<b>Упорядочите компоненты современной IDE в порядке их типичного использования в сессии разработки:</b> 1. Отладчик для поиска ошибок. 2. Редактор для написания кода. 3. Компилятор для сборки проекта. Ответ запишите в виде последовательности цифр через запятую слева направо.	2,3,1	Задание закрытого типа на установление последовательности	1	1												
16.	<b>Прочитайте текст вопроса и соотнесите поколения языков программирования с их характерными представителями:</b> <u>Поколения:</u> 1) 3-е поколение (3GL); 2) 2-е поколение (2GL); 3) 4-е поколение (4GL). <u>Представители:</u> А) Ассемблер. Б) Visual Basic, SQL, Delphi. В) Fortran, C, Pascal. Запишите выбранные буквы под соответствующими цифрами: <table border="1" data-bbox="352 1644 472 1704"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3				<table border="1" data-bbox="812 1234 932 1294"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>В</td><td>А</td><td>Б</td></tr> </table>	1	2	3	В	А	Б	Задание закрытого типа на установление соответствия	1	1
1	2	3															
1	2	3															
В	А	Б															
17.	<b>Прочитайте текст вопроса и соотнесите парадигмы программирования с их ключевыми характеристиками:</b> <u>Парадигмы:</u> 1) Функциональное программирование; 2) Логическое программирование; 3) Объектно-ориентированное программирование. <u>Характеристики:</u> А) Программа состоит из объектов, обменивающихся сообщениями;	<table border="1" data-bbox="812 1709 932 1769"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>Б</td><td>В</td><td>А</td></tr> </table>	1	2	3	Б	В	А	Задание закрытого типа на установление соответствия	1	1						
1	2	3															
Б	В	А															

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы												
	<p>используются понятия классов, наследования, полиморфизма.  Б) Вычисления рассматриваются как вычисление значений функций; используются неизменяемые данные и рекурсия (например, Haskell, Lisp).  В) Программа - это набор фактов и правил логического вывода (например, Prolog).  Запишите выбранные буквы под соответствующими цифрами:</p> <table border="1" data-bbox="352 539 470 600"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3													
1	2	3															
18.	<p><b>Прочитайте текст вопроса и соотнесите этапы жизненного цикла программы с их содержанием:</b>  <u>Этапы:</u>  1) Отладка и тестирование;  2) Сопровождение и модификация;  3) Алгоритмизация.  <u>Содержание:</u>  А) Разработка формального плана решения задачи (блок-схема, псевдокод).  Б) Выявление и исправление ошибок, проверка работоспособности.  В) Внесение изменений в программу после её передачи в эксплуатацию.  Запишите выбранные буквы под соответствующими цифрами:</p> <table border="1" data-bbox="352 1099 470 1160"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3				<table border="1" data-bbox="812 600 930 660"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>Б</td><td>В</td><td>А</td></tr> </table>	1	2	3	Б	В	А	Задание закрытого типа на установление соответствия	1	1
1	2	3															
1	2	3															
Б	В	А															
19.	<p><b>Прочитайте текст вопроса и соотнесите типы трансляции программ с их описанием:</b>  <u>Типы трансляции:</u>  1) Компиляция (АОТ);  2) Интерпретация;  3) JIT-компиляция.  <u>Описание:</u>  А) Построчный анализ и выполнение кода без создания отдельного исполняемого файла.  Б) Полное преобразование исходного кода в машинный код до запуска программы.  В) Компиляция часто выполняемых участков байт-кода в машинный код во время выполнения программы.  Запишите выбранные буквы под соответствующими цифрами:</p> <table border="1" data-bbox="352 1684 470 1744"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3				<table border="1" data-bbox="812 1155 930 1216"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>Б</td><td>А</td><td>В</td></tr> </table>	1	2	3	Б	А	В	Задание закрытого типа на установление соответствия	1	1
1	2	3															
1	2	3															
Б	А	В															
20.	<p><b>Прочитайте вопрос и выберите верный ответ:</b>  Укажите к какому поколению языков программирования относится язык Ассемблер  А) 1-е поколение (машинные коды);  Б) 2-е поколение;  В) 3-е поколение (высокоуровневые);  Г) 4-е поколение (визуальные среды).</p>	Б	Задание закрытого типа с однозначным выбором варианта ответа	1	1												
21.	<p><b>Прочитайте вопрос и выберите верный ответ:</b>  В основе языка Prolog лежит</p>	В	Задание закрытого типа с однозначным	1	1												

<b>№ задания</b>	<b>Содержание задания</b>	<b>Ответ на задание</b>	<b>Тип задания</b>	<b>Уровень сложности (балл)</b>	<b>№ Темы</b>
	парадигма программирования: А) Императивное программирование; Б) Функциональное программирование; В) Логическое программирование; Г) Объектно-ориентированное программирование.		выбором варианта ответа		
22.	<b>Прочитайте вопрос и выберите верный ответ:</b> Результатом работы компоновщика (линковщика) является: А) Исходный код на языке высокого уровня; Б) Объектный код (obj-файл); В) Исполняемый файл (exe, elf); Г) Документация программы.	В	Задание закрытого типа с однозначным выбором варианта ответа	1	1
23.	<b>Прочитайте вопрос и выберите верный ответ:</b> За компиляцией непосредственно следует этап жизненного цикла программы: А) Отладка и тестирование; Б) Написание исходного кода; В) Компоновка (линковка); Г) Постановка задачи.	В	Задание закрытого типа с однозначным выбором варианта ответа	1	1
24.	<b>Прочитайте вопрос и выберите верный ответ:</b> Укажите что из перечисленного является примером языка четвертого поколения (4GL) А) Assembler; Б) C; В) SQL; Г) Pascal.	В	Задание закрытого типа с однозначным выбором варианта ответа	1	1
25.	<b>Прочитайте и выберите два верных ответа:</b> Укажите какие из перечисленных языков программирования относятся к высокоуровневым (3-е поколение): А) Fortran; Б) Ассемблер; В) Pascal; Г) Машинные коды.	А, В	Задание закрытого типа с многозначным выбором варианта ответа	1	1
26.	<b>Прочитайте и выберите два верных ответа:</b> К <b>компиляции</b> (в отличие от интерпретации) относятся характеристики: А) Программа выполняется быстрее, так как машинный код выполняется напрямую процессором; Б) Программа может быть запущена только при наличии исходного кода и интерпретатора; В) Требуется время для полной сборки проекта перед запуском; Г) Ошибка в одной строке может остановить программу до того, как она будет выполнена.	А, В	Задание закрытого типа с многозначным выбором варианта ответа	1	1
27.	<b>Прочитайте и выберите два верных ответа:</b> На этапе сопровождения и модификации программы выполняются задачи: А) Исправление ошибок, обнаруженных пользователями после	А, В	Задание закрытого типа с многозначным выбором варианта ответа	1	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	внедрения; Б) Разработка блок-схемы алгоритма; В) Адаптация программы к изменениям операционной среды (новое оборудование, обновление ОС); Г) Написание исходного кода (кодирование).				
28.	<b>Прочитайте и дополните фразу:</b> Свойство алгоритма, означающее, что процесс решения задачи должен быть разбит на отдельные законченные шаги (действия), называется _____ .	дискретность ю	Задание открытого типа на дополнение	2	2
29.	<b>Прочитайте и дополните фразу:</b> Свойство алгоритма, означающее, что при одних и тех же исходных данных он должен давать один и тот же результат (отсутствие неоднозначности), называется _____ .	детерминированностью	Задание открытого типа на дополнение	2	2
30.	<b>Прочитайте и дополните фразу:</b> Свойство алгоритма, означающее, что он должен завершать свою работу за конечное число шагов, называется _____ .	результативностью	Задание открытого типа на дополнение	2	2
31.	<b>Прочитайте и дополните фразу:</b> В блок-схемах алгоритмов (по ГОСТ) фигура в форме _____ обозначает начало или конец алгоритма.	овала	Задание открытого типа на дополнение	2	2
32.	<b>Прочитайте и дополните фразу:</b> Временная сложность алгоритма оценивает количество _____ операций, выполняемых алгоритмом в зависимости от размера входных данных N.	элементарных (или базовых)	Задание открытого типа на дополнение	2	2
33.	<b>Прочитайте и дополните фразу:</b> Символ <b>O(f(N))</b> (O-большое) в асимптотическом анализе алгоритмов обозначает _____ функции сложности при стремлении N к бесконечности.	верхнюю асимптотическую границу (или порядок роста)	Задание открытого типа на дополнение	2	2
34.	<b>Прочитайте вопрос и дайте развернутый ответ.</b> Перечислите четыре основных свойства алгоритма.	1) Дискретность. 2) Детерминированность (определённость). 3) Результативность (конечность). 4) Массовость.	Задание открытого типа с развернутым ответом	4	2
35.	<b>Прочитайте вопрос и дайте развернутый ответ.</b> Опишите псевдокод как способ описания алгоритма.	Псевдокод - это способ описания алгоритма, использующий смесь естественного языка, математических обозначений и управляющих конструкций языков программирования (циклы, условия), но без жёстких	Задание открытого типа с развернутым ответом	4	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
		синтаксических правил.			
36.	<p><b>Прочитайте вопрос и дайте развернутый ответ.</b> Перечислите не менее трех основных классов сложности алгоритмов по возрастанию времени работы (от наилучшего к наихудшему)</p>	<p>1) <math>O(1)</math> - константная сложность. 2) <math>O(\log N)</math> - логарифмическая сложность. 3) <math>O(N)</math> - линейная сложность. 4) <math>O(N \log N)</math> - линейно-логарифмическая сложность. 5) <math>O(N^2)</math> - квадратичная сложность. 6) <math>O(2^N)</math> - экспоненциальная сложность.</p>	Задание открытого типа с развернутым ответом	4	2
37.	<p><b>Прочитайте вопрос и дайте развернутый ответ.</b> Назовите не менее трех основные фигуры блок-схемы согласно ГОСТ</p>	<p>Основные фигуры блок-схем: 1) Терминатор (овал). 2) Процесс (прямоугольник). 3) Решение (ромб). 4) Ввод/вывод (параллелограмм). 5) Предопределённый процесс (прямоугольник с вертикальными боковыми линиями). 6) Стрелки (линии потока).</p>	Задание открытого типа с развернутым ответом	4	2
38.	<p><b>Упорядочите свойства алгоритма в порядке их значимости для формального доказательства корректности (от фундаментальных к обеспечивающим):</b> 1. Результативность. 2. Детерминированность. 3. Дискретность. Ответ запишите в виде последовательности цифр через запятую слева направо.</p>	3,2,1	Задание закрытого типа на установление последовательности	1	2
39.	<p><b>Упорядочите классы сложности алгоритмов по возрастанию времени работы (от самого быстрого к самому медленному):</b> 1. <math>O(N^2)</math>. 2. <math>O(1)</math>. 3. <math>O(N \log N)</math>. 4. <math>O(N)</math>. 5. <math>O(\log N)</math>. Ответ запишите в виде последовательности цифр через запятую слева направо.</p>	2,5,4,3,1	Задание закрытого типа на установление последовательности	1	2
40.	<p><b>Упорядочите этапы создания блок-схемы алгоритма в их логической последовательности:</b> 1. Определить порядок соединения блоков стрелками. 2. Нарисовать блоки для каждого действия. 3. Определить необходимые блоки-решения (условия). 4. Уточнить результаты и корректность переходов.</p>	2,3,1,4	Задание закрытого типа на установление последовательности	1	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы												
	<p>Ответ запишите в виде последовательности цифр через запятую слева направо.</p>																
41.	<p><b>Упорядочите способы описания алгоритмов по степени нарастания формализации (от наименее формального к наиболее формальному):</b>  1. На естественном языке.  2. Псевдокод.  3. Блок-схема по ГОСТ.  Ответ запишите в виде последовательности цифр через запятую слева направо.</p>	1,2,3	Задание закрытого типа на установление последовательности	1	2												
42.	<p><b>Упорядочите этапы анализа временной сложности алгоритма для поиска асимптотической оценки O-большое:</b>  1. Сохранить только доминирующий член (с наибольшей степенью роста).  2. Определить количество элементарных операций в зависимости от N.  3. Отбросить константные множители.  Ответ запишите в виде последовательности цифр через запятую слева направо.</p>	2,1,3	Задание закрытого типа на установление последовательности	1	2												
43.	<p><b>Прочитайте текст вопроса и соотнесите свойства алгоритма с их определениями:</b>  <u>Свойства:</u>  1) Дискретность;  2) Детерминированность;  3) Массовость.  <u>Определения:</u>  А) Применимость алгоритма ко всем задачам данного класса (для любых допустимых исходных данных).  Б) Разбивка процесса на последовательность отдельных шагов.  В) Однозначность и воспроизводимость результата при одинаковых исходных данных.  Запишите выбранные буквы под соответствующими цифрами:</p> <table border="1" data-bbox="352 1536 472 1592"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3				<table border="1" data-bbox="815 1021 935 1077"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>Б</td><td>В</td><td>А</td></tr> </table>	1	2	3	Б	В	А	Задание закрытого типа на установление соответствия	1	2
1	2	3															
1	2	3															
Б	В	А															
44.	<p><b>Прочитайте текст вопроса и соотнесите фигуры блок-схем с их обозначениями:</b>  <u>Фигуры:</u>  1) Процесс (действие);  2) Решение (условие);  3) Ввод/вывод.  <u>Обозначения:</u>  А) Ромб;  Б) Параллелограмм;  В) Прямоугольник.  Запишите выбранные буквы под соответствующими цифрами:</p> <table border="1" data-bbox="352 1962 472 2018"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3				<table border="1" data-bbox="815 1603 935 1659"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>В</td><td>А</td><td>Б</td></tr> </table>	1	2	3	В	А	Б	Задание закрытого типа на установление соответствия	1	2
1	2	3															
1	2	3															
В	А	Б															
45.	<p><b>Прочитайте текст вопроса и соотнесите классы сложности с их</b></p>	<table border="1" data-bbox="815 2029 935 2085"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>В</td><td>А</td><td>Б</td></tr> </table>	1	2	3	В	А	Б	Задание закрытого типа на	1	2						
1	2	3															
В	А	Б															

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы												
	<p><b>асимптотическими записями (О-большое):</b>  <u>Классы:</u>            1) Линейная сложность;            2) Квадратичная сложность;            3) Логарифмическая сложность.  <u>Записи:</u>            А) <math>O(N^2)</math>;            Б) <math>O(\log N)</math>;            В) <math>O(N)</math>.            Запишите выбранные буквы под соответствующими цифрами:</p> <table border="1" data-bbox="352 566 472 622"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3					установление соответствия								
1	2	3															
46.	<p><b>Прочитайте текст вопроса и соотнесите этапы описания алгоритма с их характеристиками:</b>            Способы:            1) Блок-схема;            2) Псевдокод;            3) Программный код.            Характеристики:            А) Строгий синтаксис, готов к выполнению на компьютере.            Б) Визуальное представление с использованием геометрических фигур.            В) Компактная запись, акцент на логике, отсутствие жёсткого синтаксиса.            Запишите выбранные буквы под соответствующими цифрами:</p> <table border="1" data-bbox="352 1126 472 1182"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3				<table border="1" data-bbox="815 633 935 689"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>Б</td><td>В</td><td>А</td></tr> </table>	1	2	3	Б	В	А	Задание закрытого типа на установление соответствия	1	2
1	2	3															
1	2	3															
Б	В	А															
47.	<p><b>Прочитайте вопрос и выберите верный ответ:</b>            Если при одних и тех же исходных данных программа иногда выдаёт один результат, а иногда - другой, нарушается свойство алгоритма:            А) Дискретность;            Б) Детерминированность;            В) Результативность;            Г) Массовость.</p>	Б	Задание закрытого типа с однозначным выбором варианта ответа	1	2												
48.	<p><b>Прочитайте вопрос и выберите верный ответ:</b>            Укажите какой символ в блок-схеме используется для обозначения начала и конца алгоритма?            А) Прямоугольник;            Б) Ромб;            В) Овал (терминатор);            Г) Параллелограмм.</p>	В	Задание закрытого типа с однозначным выбором варианта ответа	1	2												
49.	<p><b>Прочитайте вопрос и выберите верный ответ:</b>            Временная сложность <math>O(N^2)</math> для алгоритма означает:            А) Алгоритм всегда выполняется за 2 секунды;            Б) Время выполнения алгоритма пропорционально квадрату размера входных данных;            В) Алгоритм требует <math>N^2</math> единиц памяти;            Г) Алгоритм не может обработать более 100 элементов.</p>	Б	Задание закрытого типа с однозначным выбором варианта ответа	1	2												

<b>№ задания</b>	<b>Содержание задания</b>	<b>Ответ на задание</b>	<b>Тип задания</b>	<b>Уровень сложности (балл)</b>	<b>№ Темы</b>
50.	<b>Прочитайте вопрос и выберите верный ответ:</b> Укажите что из перечисленного является преимуществом псевдокода перед блок-схемой: А) Графическое отображение логики; Б) Компактность и отсутствие необходимости рисования; В) Возможность непосредственного выполнения на компьютере; Г) Строгая стандартизация по ГОСТ.	Б	Задание закрытого типа с однозначным выбором варианта ответа	1	2
51.	<b>Прочитайте вопрос и выберите верный ответ:</b> Укажите какой класс сложности имеет алгоритм бинарного поиска в отсортированном массиве? А) $O(1)$ ; Б) $O(N)$ ; В) $O(\log N)$ ; Г) $O(N^2)$ .	В	Задание закрытого типа с однозначным выбором варианта ответа	1	2
52.	<b>Прочитайте и выберите два верных ответа:</b> Укажите какие из перечисленных фигур используются в блок-схемах для обозначения безусловного (не зависящего от условий) действия? А) Прямоугольник (процесс); Б) Ромб (решение); В) Параллелограмм (ввод/вывод); Г) Овал (терминатор).	А, В	Задание закрытого типа с многозначным выбором варианта ответа	1	2
53.	<b>Прочитайте и выберите два верных ответа:</b> Для алгоритмов с логарифмической сложностью $O(\log N)$ верны утверждения: А) Логарифмические алгоритмы очень эффективны для больших объёмов данных; Б) При увеличении $N$ в 10 раз время работы увеличивается примерно в 10 раз; В) Примером является бинарный поиск (деление массива пополам на каждом шаге); Г) $Lg(N)$ растёт быстрее, чем $N$ (при $N > 1$ ).	А, В	Задание закрытого типа с многозначным выбором варианта ответа	1	2
54.	<b>Прочитайте и дополните фразу:</b> Именованная область памяти, хранящая значение определённого типа, которое может изменяться в процессе выполнения программы, называется _____.	переменной	Задание открытого типа на дополнение	2	3
55.	<b>Прочитайте и дополните фразу:</b> Способ типизации, при котором тип переменной проверяется на этапе компиляции и не может измениться во время выполнения программы, называется _____.	статической типизацией	Задание открытого типа на дополнение	2	3
56.	<b>Прочитайте и дополните фразу:</b> Управляющая конструкция, которая повторяет выполнение блока операторов до тех пор, пока выполняется некоторое условие, называется _____.	циклом	Задание открытого типа на дополнение	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
57.	<b>Прочитайте и дополните фразу:</b> Переменные, которые передаются в подпрограмму при её вызове и используются внутри неё как данные, называются _____.	фактическим и параметрами	Задание открытого типа на дополнение	2	3
58.	<b>Прочитайте и дополните фразу:</b> Область действия переменной, в которой она доступна для использования, называется _____.	областью видимости переменной	Задание открытого типа на дополнение	2	3
59.	<b>Прочитайте и дополните фразу:</b> Способ передачи параметров в подпрограмму, при котором в подпрограмму передаётся копия значения аргумента (изменения внутри подпрограммы не влияют на оригинал), называется _____.	передачей по значению	Задание открытого типа на дополнение	2	3
60.	<b>Прочитайте вопрос и дайте развернутый ответ.</b> Укажите три основные базовые управляющие конструкции структурированного программирования, сформулированные Э. Дейкстрой.	1) Следование (последовательность). 2) Ветвление (выбор). 3) Повторение (цикл).	Задание открытого типа с развернутым ответом	4	3
61.	<b>Прочитайте вопрос и дайте развернутый ответ.</b> Опишите стек вызовов (call stack) и его роль при вызове подпрограмм.	Стек вызовов - это структура данных, которая хранит информацию об активных подпрограммах (функциях/процедурах) в процессе выполнения программы.	Задание открытого типа с развернутым ответом	4	3
62.	<b>Прочитайте вопрос и дайте развернутый ответ.</b> Назовите четыре простых типа данных, которые обычно присутствуют в языках программирования	1) Целый. 2) Вещественный. 3) Символьный. 4) Логический.	Задание открытого типа с развернутым ответом	4	3
63.	<b>Упорядочите этапы жизненного цикла переменной в программе от её создания до уничтожения:</b> 1. Уничтожение переменной (освобождение памяти). 2. Инициализация (присвоение начального значения). 3. Объявление (определение типа и имени). 4. Использование в вычислениях. Ответ запишите в виде последовательности цифр через запятую слева направо.	3,2,4,1	Задание закрытого типа на установление последовательности	1	3
64.	<b>Упорядочите уровни видимости переменных в языках программирования от наиболее широкой области к наиболее узкой (локальной):</b> 1. Локальная переменная (внутри функции). 2. Глобальная переменная (файл/модуль). 3. Переменная уровня блока (внутри {	2,1,3	Задание закрытого типа на установление последовательности	1	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы												
	}). Ответ запишите в виде последовательности цифр через запятую слева направо.																
65.	<b>Упорядочите способы приведения типов по степени риска потери данных (от наименее рискованного к наиболее рискованному):</b> 1. Явное приведение от большего типа к меньшему (например, double → int). 2. Автоматическое (неявное) приведение от меньшего типа к большему (например, int → double). 3. Явное приведение между несовместимыми типами (например, int → pointer). Ответ запишите в виде последовательности цифр через запятую слева направо.	2,1,3	Задание закрытого типа на установление последовательности	1	3												
66.	<b>Упорядочите этапы выполнения подпрограммы в их логической последовательности:</b> 1. Выполнение тела подпрограммы. 2. Передача управления в подпрограмму (вызов). 3. Возврат значения (для функции) и передача управления обратно. 4. Помещение в стек адреса возврата и фактических параметров. Ответ запишите в виде последовательности цифр через запятую слева направо.	2,4,1,3	Задание закрытого типа на установление последовательности	1	3												
67.	<b>Прочитайте текст вопроса и соотнесите понятия с их определениями:</b> Понятия: 1) Переменная; 2) Константа; 3) Литерал. Определения: А) Непосредственное представление значения в исходном коде (например, 25, "hello"). Б) Именованная область памяти с неизменяемым значением. В) Именованная область памяти с изменяемым значением. Запишите выбранные буквы под соответствующими цифрами: <table border="1" data-bbox="352 1615 472 1675"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3				<table border="1" data-bbox="815 1151 930 1211"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>В</td><td>Б</td><td>А</td></tr> </table>	1	2	3	В	Б	А	Задание закрытого типа на установление соответствия	1	3
1	2	3															
1	2	3															
В	Б	А															
68.	<b>Прочитайте текст вопроса и соотнесите виды циклов с их описанием:</b> <u>Виды циклов:</u> 1) Цикл с предусловием (while); 2) Цикл с постусловием (do-while); 3) Цикл со счётчиком (for). <u>Описание:</u> А) Тело цикла выполняется <b>хотя бы один раз</b> , затем проверяется условие продолжения. Б) Тело цикла выполняется, пока истинно условие, которое проверяется <b>перед каждой</b>	<table border="1" data-bbox="815 1682 930 1742"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>Б</td><td>А</td><td>В</td></tr> </table>	1	2	3	Б	А	В	Задание закрытого типа на установление соответствия	1	3						
1	2	3															
Б	А	В															

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы						
	<p>итераций.  В) Количество итераций определяется переменной-счётчиком, которая изменяется на каждом шаге.  Запишите выбранные буквы под соответствующими цифрами:</p> <table border="1" data-bbox="352 427 472 488"> <tr> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>	1	2	3							
1	2	3									
69.	<p><b>Прочитайте вопрос и выберите верный ответ:</b>  В упрощённой теореме Бёма-Якопини (достаточно двух конструкций) отсутствует базовая управляющая конструкция структурированного программирования:  А) Следование;  Б) Ветвление;  В) Цикл for;  Г) Цикл while.</p>	В	Задание закрытого типа с однозначным выбором варианта ответа	1	3						
70.	<p><b>Прочитайте вопрос и выберите верный ответ:</b>  Укажите что из перечисленного является неявным приведением типа (coercion):  А) int x = (int)3.14;  Б) float y = 5; (автоматическое преобразование int в float);  В) char c = (char)65;  Г) void* ptr = &amp;x;</p>	Б	Задание закрытого типа с однозначным выбором варианта ответа	1	3						
71.	<p><b>Прочитайте вопрос и выберите верный ответ:</b>  К динамической типизации применима характеристика  А) Тип переменной должен быть объявлен до её использования;  Б) Ошибки типов обнаруживаются только во время выполнения программы;  В) Переменная не может изменить тип после инициализации;  Г) Тип переменной указывается в объявлении явно.</p>	Б	Задание закрытого типа с однозначным выбором варианта ответа	1	3						
72.	<p><b>Прочитайте вопрос и выберите верный ответ:</b>  Укажите какой тип цикла гарантирует выполнение тела хотя бы один раз, независимо от условия  А) while (с предусловием);  Б) do-while (с постусловием);  В) for (счётный);  Г) repeat-until.</p>	Б	Задание закрытого типа с однозначным выбором варианта ответа	1	3						
73.	<p><b>Прочитайте и выберите два верных ответа:</b>  К статической типизации относятся характеристики:  А) Тип переменной проверяется на этапе компиляции;  Б) Одна переменная может хранить значения разных типов;  В) До выполнения программы все типы уже известны;  Г) Python является примером такого подхода.</p>	А, В	Задание закрытого типа с многозначным выбором варианта ответа	1	3						

<b>№ задания</b>	<b>Содержание задания</b>	<b>Ответ на задание</b>	<b>Тип задания</b>	<b>Уровень сложности (балл)</b>	<b>№ Темы</b>
74.	<b>Прочитайте и выберите два верных ответа:</b> Для передачи параметров по ссылке в C++ (reference) верны утверждения: А) Изменение параметра внутри функции изменяет оригинальную переменную; Б) В функцию передаётся копия значения; В) Позволяет функции возвращать более одного результата; Г) Работает только со скалярными типами int и float.	А, В	Задание закрытого типа с многозначным выбором варианта ответа	1	3
75.	<b>Прочитайте и выберите два верных ответа:</b> Укажите какие из перечисленных структур данных являются статическими (фиксированного размера): А) Массив, объявленный как int arr[100]; Б) Связный список (linked list); В) Запись (структура) с фиксированным числом полей; Г) Динамический массив (Vector, ArrayList).	А, В	Задание закрытого типа с многозначным выбором варианта ответа	1	3
76.	<b>Прочитайте и выберите два верных ответа:</b> Для стека вызовов (call stack) верны утверждения: А) Он работает по принципу LIFO (последним пришёл — первым ушёл); Б) Он используется только для передачи параметров, но не хранения локальных переменных; В) Каждый вызов функции создаёт новый кадр (frame) в стеке; Г) Стек вызовов не имеет ограничений по размеру.	А, В	Задание закрытого типа с многозначным выбором варианта ответа	1	3
77.	<b>Прочитайте и дополните фразу:</b> В объектно-ориентированном программировании _____ является шаблоном (чертежом, описанием), определяющим структуру и поведение объектов.	класс	Задание открытого типа на дополнение		4
78.	<b>Прочитайте и дополните фразу:</b> Процесс создания нового класса на основе существующего, при котором дочерний класс наследует поля и методы родительского, называется _____.	наследованием	Задание открытого типа на дополнение		4
79.	<b>Прочитайте и дополните фразу:</b> Принцип ООП, заключающийся в сокрытии внутренней реализации объекта и предоставлении доступа к его состоянию только через открытые методы, называется _____.	инкапсуляцией	Задание открытого типа на дополнение		
80.	<b>Прочитайте и дополните фразу:</b> Специальный метод класса, который автоматически вызывается при создании объекта (экземпляра класса) и служит для инициализации его полей, называется _____.	конструктором	Задание открытого типа на дополнение		

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы						
81.	<b>Прочитайте и дополните фразу:</b> Методы, предназначенные для чтения (get) и изменения (set) значений приватных (закрытых) полей класса, называются _____ и _____.	геттерами и сеттерами	Задание открытого типа на дополнение								
82.	<b>Прочитайте и дополните фразу:</b> Свойство полиморфизма, позволяющее создавать несколько методов с одинаковым именем, но разными параметрами (типом или количеством), называется _____.	перегрузкой методов	Задание открытого типа на дополнение								
83.	<b>Прочитайте вопрос и дайте развернутый ответ.</b> Укажите три основных принципа объектно-ориентированного программирования.	1) Инкапсуляция 2) Наследование 3) Полиморфизм	Задание открытого типа с развернутым ответом								
84.	<b>Прочитайте вопрос и дайте развернутый ответ.</b> Назовите основные модификаторы доступа (спецификаторы видимости) в ООП (например, в C++, Java, C#)	1) public (открытый). 2) private (закрытый). 3) protected (защищенный).	Задание открытого типа с развернутым ответом								
85.	<b>Упорядочите этапы создания и использования объекта в ООП в их логической последовательности:</b> 1. Вызов метода объекта (например, object.method()). 2. Объявление класса (поля, методы, конструктор). 3. Создание объекта (экземпляра) с помощью конструктора (new ClassName()). Ответ запишите в виде последовательности цифр через запятую слева направо.	2,3,1	Задание закрытого типа на установление последовательности								
86.	<b>Упорядочите уровни доступа (модификаторы видимости) от наиболее открытого к наиболее закрытому:</b> 1. public. 2. protected. 3. private. Ответ запишите в виде последовательности цифр через запятую слева направо.	1,2,3	Задание закрытого типа на установление последовательности								
87.	<b>Упорядочите шаги при создании класса с инкапсуляцией:</b> 1. Создание приватных полей. 2. Создание публичных методов (геттеров/сеттеров) для доступа к полям. 3. Определение класса и его публичного интерфейса. Ответ запишите в виде последовательности цифр через запятую слева направо.	3,1,2	Задание закрытого типа на установление последовательности								
88.	<b>Прочитайте текст вопроса и соотнесите понятия ООП с их определениями:</b> <u>Понятия:</u>	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>В</td> <td>А</td> <td>Б</td> </tr> </table>	1	2	3	В	А	Б	Задание закрытого типа на установление соответствия		
1	2	3									
В	А	Б									

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы												
	1) Класс; 2) Объект; 3) Поле класса. <u>Определения:</u> А) Конкретный экземпляр класса, имеющий собственное состояние. Б) Переменная, принадлежащая классу, хранящая его состояние (данные). В) Шаблон (тип данных), описывающий структуру и поведение будущих объектов. Запишите выбранные буквы под соответствующими цифрами: <table border="1" data-bbox="352 622 470 680"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3													
1	2	3															
89.	<b>Прочитайте текст вопроса и соотнесите принципы ООП с их описанием:</b> <u>Принципы:</u> 1) Инкапсуляция; 2) Наследование; 3) Полиморфизм. <u>Описание:</u> А) Создание нового класса на основе существующего с добавлением или переопределением функциональности. Б) Скрытие внутренней реализации объекта, предоставление доступа через открытый интерфейс. В) Способность объектов разных классов реагировать на один и тот же вызов метода по-своему. Запишите выбранные буквы под соответствующими цифрами: <table border="1" data-bbox="352 1234 470 1292"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3				<table border="1" data-bbox="813 685 932 743"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>Б</td><td>А</td><td>В</td></tr> </table>	1	2	3	Б	А	В	Задание закрытого типа на установление соответствия		
1	2	3															
1	2	3															
Б	А	В															
90.	<b>Прочитайте текст вопроса и соотнесите виды методов с их характеристиками:</b> <u>Виды:</u> 1) Геттер (getter); 2) Сеттер (setter); 3) Конструктор. <u>Характеристики:</u> А) Метод, возвращающий значение приватного поля (чтение). Б) Метод, вызываемый при создании объекта для инициализации полей. В) Метод, изменяющий значение приватного поля (запись). Запишите выбранные буквы под соответствующими цифрами: <table border="1" data-bbox="352 1736 470 1794"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td></td><td></td><td></td></tr> </table>	1	2	3				<table border="1" data-bbox="813 1296 932 1355"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>А</td><td>В</td><td>Б</td></tr> </table>	1	2	3	А	В	Б	Задание закрытого типа на установление соответствия		
1	2	3															
1	2	3															
А	В	Б															
91.	<b>Прочитайте текст вопроса и соотнесите модификаторы доступа с их областью видимости:</b> <u>Модификаторы:</u> 1) public; 2) private; 3) protected. <u>Область видимости:</u> А) Доступен только внутри самого класса.	<table border="1" data-bbox="813 1798 932 1856"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>Б</td><td>А</td><td>В</td></tr> </table>	1	2	3	Б	А	В	Задание закрытого типа на установление соответствия								
1	2	3															
Б	А	В															

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы						
	Б) Доступен везде. В) Доступен внутри класса и внутри классов-наследников. Запишите выбранные буквы под соответствующими цифрами: <table border="1" data-bbox="352 376 472 432"> <tr> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>	1	2	3							
1	2	3									
92.	<b>Прочитайте вопрос и выберите верный ответ:</b> Укажите какой принцип ООП позволяет использовать объект Dog и объект Cat с единым методом makeSound(), где каждый выводит свой звук: А) Инкапсуляция; Б) Наследование; В) Полиморфизм; Г) Абстракция.	В	Задание закрытого типа с однозначным выбором варианта ответа								
93.	<b>Прочитайте вопрос и выберите верный ответ:</b> Конструктор класса – это: А) Метод, который вызывается перед удалением объекта; Б) Специальный метод, вызываемый при создании объекта для его инициализации; В) Метод для доступа к приватному полю; Г) Механизм создания нового класса на основе существующего.	Б	Задание закрытого типа с однозначным выбором варианта ответа								
94.	<b>Прочитайте вопрос и выберите верный ответ:</b> Укажите какой модификатор доступа обеспечивает максимальный уровень сокрытия (наименьшую видимость): А) private; Б) protected; В) public; Г) internal.	А	Задание закрытого типа с однозначным выбором варианта ответа								
95.	<b>Прочитайте вопрос и выберите верный ответ:</b> Примером перегрузки (overload) метода является: А) В классе А и его наследнике В есть метод method(), делающий разное; Б) В одном классе есть методы calculate(int a) и calculate(double a); В) В классе есть приватное поле и публичный метод для его чтения; Г) Класс В наследует метод method() от класса А.	Б	Задание закрытого типа с однозначным выбором варианта ответа								
96.	<b>Прочитайте вопрос и выберите верный ответ:</b> Геттеры и сеттеры в инкапсуляции используются: А) Для создания объектов класса; Б) Для перегрузки операторов; В) Для контролируемого доступа к приватным полям (чтение и запись); Г) Для реализации множественного наследования.	В	Задание закрытого типа с однозначным выбором варианта ответа								
97.	<b>Прочитайте и выберите два верных ответа:</b> К абстрактному классу относятся	А, В	Задание закрытого типа с многозначным								

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>характеристики:</p> <p>А) Нельзя создать объект (экземпляр) абстрактного класса;</p> <p>Б) Все методы абстрактного класса обязательно должны быть абстрактными;</p> <p>В) Абстрактный класс может содержать реализованные методы и поля;</p> <p>Г) Абстрактный класс не может иметь конструктор.</p>		выбором варианта ответа		
98.	<p><b>Прочитайте и выберите два верных ответа:</b></p> <p>Укажите какие из следующих утверждений верны для деструктора класса (в языках с управляемой памятью, таких как C#/Java):</p> <p>А) Используется для освобождения неуправляемых ресурсов (файлы, сетевые соединения);</p> <p>Б) Вызывается автоматически при создании объекта;</p> <p>В) Не может быть перегружен (имеет только одну версию без параметров);</p> <p>Г) Обязательно должен быть определён для каждого класса.</p>	А, В	Задание закрытого типа с многозначным выбором варианта ответа		
99.	<p><b>Прочитайте и выберите два верных ответа:</b></p> <p>К статической типизации относятся характеристики:</p> <p>А) Тип переменной проверяется на этапе компиляции;</p> <p>Б) Переменная может менять свой тип во время выполнения;</p> <p>В) Тип переменной фиксирован и не может измениться;</p> <p>Г) Python является примером такого подхода.</p>	А, В	Задание закрытого типа с многозначным выбором варианта ответа		
100.	<p><b>Прочитайте и выберите два верных ответа:</b></p> <p>Укажите какие из перечисленных утверждений верны для алгоритмов с логарифмической сложностью <math>O(\log N)</math>:</p> <p>А) Логарифмические алгоритмы очень эффективны для больших объёмов данных;</p> <p>Б) При увеличении <math>N</math> в 10 раз время работы увеличивается примерно в 10 раз;</p> <p>В) Примером является бинарный поиск (деление массива пополам на каждом шаге);</p> <p>Г) <math>\lg(N)</math> растёт быстрее, чем <math>N</math> (при <math>N &gt; 1</math>).</p>	А, В	Задание закрытого типа с многозначным выбором варианта ответа		

## **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
<b>«Зачтено»</b>	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
<b>«Не зачтено»</b>	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
<b>«Отлично»</b>	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
<b>«Хорошо»</b>	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
<b>«Удовлетворительно»</b>	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
<b>«Не удовлетворительно»</b>	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