

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Заболотный, Глеб Иванович

Должность: Директор филиала

Дата подписания: 23.06.2025 10:07:51

Уникальный программный ключ:

476db7d4accb36ef8130172be235477473d63457266ce26b7e9e40f733b8b08

МИНОБРАЗОВАНИЯ РОССИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

«Самарский государственный технический университет»

(ФГБОУ ВО «СамГТУ»)

УТВЕРЖДАЮ:

Директор филиала ФГБОУ ВО
"СамГТУ" в г. Новокуйбышевске

_____ / Г.И. Заболотный

" ____ " _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.03.01 «Алгоритмизация и основы программирования»

**Код и направление подготовки
(специальность)**

13.03.02 Электроэнергетика и
электротехника

Направленность (профиль)

Электроэнергетика

Квалификация

Бакалавр

Форма обучения

Очная

Год начала подготовки

2025

Институт / факультет

Кафедры филиала ФГБОУ ВО "СамГТУ" в г.
Новокуйбышевске

Выпускающая кафедра

кафедра "Электроэнергетика,
электротехника и автоматизация
технологических процессов" (НФ- ЭЭиАТП)

Кафедра-разработчик

кафедра "Информатика и системы
управления" (НФ-ИиСУ)

Объем дисциплины, ч. / з.е.

144 / 4

**Форма контроля (промежуточная
аттестация)**

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:

Старший преподаватель

(должность, степень, ученое звание)

С.П Минеев

(ФИО)

Заведующий кафедрой

А.В. Волкодаева, кандидат
экономических наук, доцент

(ФИО, степень, ученое звание)

СОГЛАСОВАНО:

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

А.А Малафеев, кандидат
экономических наук, доцент

(ФИО, степень, ученое звание)

Руководитель образовательной
программы

А.А. Складчиков, кандидат
технических наук

(ФИО, степень, ученое звание)

Заведующий выпускающей кафедрой

А.А. Складчиков, кандидат
технических наук

(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	7
4.3 Содержание практических занятий	7
4.4. Содержание самостоятельной работы	8
5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)	9
6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения	9
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем	10
8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)	10
9. Методические материалы	11
10. Фонд оценочных средств по дисциплине (модулю)	13

**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.	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.	Знать эволюцию языков программирования, их классификацию; основные элементы языка Python, операторы и операции, управляющие структуры, структуры данных, файлы; подпрограммы, составление библиотек подпрограмм; объектно-ориентированную модель программирования, основные принципы объектноориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.
			Уметь разрабатывать алгоритмы для конкретных задач; работать в среде программирования; реализовывать построенные алгоритмы в виде программ на конкретном языке программирования (Python); оформлять код программы в соответствии со стандартом кодирования (PEP8), 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
Аудиторная контактная работа (всего), в том числе:	48	48
Лабораторные работы	16	16
Лекции	16	16
Практические занятия	16	16
Самостоятельная работа (всего), в том числе:	96	96
подготовка к практическим занятиям	96	96
Итого: час	144	144
Итого: з.е.	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	Введение в программирование	4	4	4	24	36
2	Основные конструкции языков программирования	4	4	4	24	36
3	Введение в структурированное программирование	4	4	4	24	36
4	Введение в объектно-ориентированное программирование	4	4	4	24	36

		Итого	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
2 семестр				
1	Введение в программирование	Языки программирования	Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы. Жизненный цикл программы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере.	2
2	Введение в программирование	Типы данных	Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.	2
3	Основные конструкции языков программирования	Операторы языка программирования	Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор. Условный оператор. Оператор выбора. Массивы. Двумерные массивы. Строки. Структурированный тип данных – множество. Операции над множествами. Комбинированный тип данных – запись.	2
4	Основные конструкции языков программирования	Функции и циклы	Определение функции. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы. Стандартные процедуры и функции для работы со строками.	2
5	Введение в структурированное программирование	Процедуры и функции	Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций. Рекурсия. Программирование рекурсивных алгоритмов.	2
6	Введение в структурированное программирование	Структуризация в программировании. Модульное программирование	Основы структурного программирования. Методы структурного программирования. Модульное программирование. Понятие модуля. Структура модуля. Стандартные модули.	2
7	Введение в объектно-ориентированное программирование	Введение в ООП	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс. Основные принципы ООП: абстракция, инкапсуляция, наследование, полиморфизм. Классы объектов.	2

8	Введение в объектно-ориентированное программирование	Интегрированная среда разработчика	Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.	2
Итого за семестр:				16
Итого:				16

4.2 Содержание лабораторных занятий

№ занятия	Наименование раздела	Тема лабораторного занятия	Содержание лабораторного занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
2 семестр				
1	Введение в программирование	Введение в язык программирования Python	Среда разработки Jupyter. Основные типы данных, команды ввода и вывода данных.	2
2	Введение в программирование	Математические операции в Python	Основные математические операции в Python	2
3	Основные конструкции языков программирования	Структура ветвления в Python	Конструкции if, if-else, if-elif-else	2
4	Основные конструкции языков программирования	Работа с циклами в Python	Цикл for, while	2
5	Введение в структурированное программирование	Операции над списками в Python	Тип данных list, работа со списками	2
6	Введение в структурированное программирование	Работа со строками в Python	Операции и методы обработки строк	2
7	Введение в объектно-ориентированное программирование	Функции и процедуры в Python	Синтаксис процедур и функций при составлении программы	2
8	Введение в объектно-ориентированное программирование	Работа с многомерными массивами	Создания программ обработки многомерных массивов	2
Итого за семестр:				16
Итого:				16

4.3 Содержание практических занятий

№ занятия	Наименование раздела	Тема практического занятия	Содержание практического занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
2 семестр				

1	Введение в программирование	Языки программирования	Знакомство со средой программирования	2
2	Введение в программирование	Типы данных	Работа со строками. Использование стандартных процедур и функций для работы со строками. Работа с данными типа множество. Операции над множествами.	2
3	Основные конструкции языков программирования	Операторы языка программирования	Составление программ линейной структуры. Составление программ разветвляющейся структуры. Условный оператор. Составление программ разветвляющейся структуры. Оператор выбора.	2
4	Основные конструкции языков программирования	Функции и циклы	Цикл с предусловием, цикл с параметром	2
5	Введение в структурированное программирование	Процедуры и функции	Организация процедур. Использование процедур. Организация функций. Применение рекурсивных функций.	2
6	Введение в структурированное программирование	Модульное программирование	Программирование модуля. Создание библиотеки подпрограмм.	2
7	Введение в объектно-ориентированное программирование	Интегрированная среда разработчика	Создание простого проекта	2
8	Введение в объектно-ориентированное программирование	Основные принципы ООП	Создание программы с использованием классов	2
Итого за семестр:				16
Итого:				16

4.4. Содержание самостоятельной работы

Наименование раздела	Вид самостоятельной работы	Содержание самостоятельной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов
2 семестр			
Введение в программирование	Языки программирования	Эволюция языков программирования. Классификация программного обеспечения ПК	12
Введение в программирование	Типы данных	Типы данных, определяемые программистом. Перечисляемый и интервальный типы	12
Основные конструкции языков программирования	Операторы языка программирования	Создание и удаление динамических переменных. Структуры данных на основе указателей.	12
Основные конструкции языков программирования	Функции и циклы	Создание цикла с постусловием в Python	12
Введение в структурированное программирование	Процедуры и функции	Вероятностные алгоритмы. Дополнительные сведения о процедурах и функциях	12
Введение в структурированное программирование	Структуризация в программировании	Стили структурного программирования	12

Введение в объектно-ориентированное программирование	Основные принципы ООП	Принципы SOLID	12
Введение в объектно-ориентированное программирование	Интегрированная среда разработчика	Классы ООП: виды, назначение, свойства, методы, события. Объявления класса. Создание наследованного класса	12
Итого за семестр:			96
Итого:			96

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. Язык Python; Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2020.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 102400	Электронный ресурс
2	Язык программирования Python; Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 97589	Электронный ресурс
Дополнительная литература		
3	Простой Python просто с нуля; СОЛОН-Пресс, 2019.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 94940	Электронный ресурс

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

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

При проведении лекционных занятий используется мультимедийное оборудование.

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

№ п/п	Наименование	Производитель	Способ распространения
1	PyCharm Community Edition	JetBrains (Зарубежный)	Свободно распространяемое
2	Python	PSF (Зарубежный)	Свободно распространяемое
3	Microsoft Office	Microsoft (Зарубежный)	Лицензионное
4	Jupyter Notebook	Open-source (Зарубежный)	Свободно распространяемое

5	Microsoft Windows	Microsoft (Зарубежный)	Лицензионное
---	-------------------	---------------------------	--------------

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

№ п/п	Наименование	Краткое описание	Режим доступа
1	Научная электронная библиотека	http://elibrary.ru/	Ресурсы открытого доступа
2	eLIBRARY.ru	http://www.eLIBRARY.ru/	Российские базы данных ограниченного доступа

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

Лекционные занятия

302 (учебный корпус) Наименование: Учебная аудитория для проведения занятий лекционного типа, текущего контроля и промежуточной аттестации. Перечень специализированного оборудования: Технические средства обучения, служащие для представления учебной информации большой аудитории, набор демонстрационного оборудования: экран, проектор, переносной ноутбук. Специализированная мебель: 22 ученических стола, 44 ученических стула, доска, стол, кафедра и стул для преподавателя

Практические занятия

101 (учебный корпус)

Наименование:

Компьютерный класс – учебная аудитория для проведения занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации – лаборатория специализированных программных продуктов.

Перечень специализированного оборудования:

Технические средства обучения, служащие для представления учебной информации большой аудитории, набор демонстрационного оборудования: экран, проектор, переносной ноутбук. Лаборатория, оснащенная компьютерной техникой с возможностью подключения к сети «Интернет» и с доступом в электронную информационно-образовательную среду СамГТУ; компьютерами - 12 шт., оборудованная учебной мебелью: 12 компьютерных столов, 12 стульев, стол и стул преподавателя, доска.

Самостоятельная работа

212 (учебный корпус)

Наименование:

Помещение для самостоятельной работы – учебная аудитория для курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций.

Перечень специализированного оборудования:

Аудитория, оснащенная компьютерной техникой с возможностью подключения к сети «Интернет» и с доступом в

электронную информационно-образовательную среду СамГТУ.

Оборудование: 3 компьютера с выходом в сеть Интернет.

Специализированная мебель: 3 компьютерных стола, 3 стула

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 отчетности 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»**

Код и направление подготовки (специальность)	13.03.02 Электроэнергетика и электротехника
Направленность (профиль)	Электроэнергетика
Квалификация	Бакалавр
Форма обучения	Очная
Год начала подготовки	2025
Институт / факультет	Кафедры филиала ФГБОУ ВО "СамГТУ" в г. Новокуйбышевске
Выпускающая кафедра	кафедра "Электроэнергетика, электротехника и автоматизация технологических процессов" (НФ- ЭЭиАТП)
Кафедра-разработчик	кафедра "Информатика и системы управления" (НФ-ИиСУ)
Объем дисциплины, ч. / з.е.	144 / 4
Форма контроля (промежуточная аттестация)	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.	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.	Знать эволюцию языков программирования, их классификацию; основные элементы языка Python, операторы и операции, управляющие структуры, структуры данных, файлы; подпрограммы, составление библиотек подпрограмм; объектно-ориентированную модель программирования, основные принципы объектноориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.
			Уметь разрабатывать алгоритмы для конкретных задач; работать в среде программирования; реализовывать построенные алгоритмы в виде программ на конкретном языке программирования (Python); оформлять код программы в соответствии со стандартом кодирования (PEP8), 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
Введение в программирование				
ОПК-2.1 Алгоритмизирует решение задач и реализует алгоритмы с использованием программных средств.	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Перечень вопросов к зачету	Нет	Да
	Уметь разрабатывать алгоритмы для конкретных задач	Перечень вопросов к зачету	Нет	Да
ОПК-2.2 Принимает участие в процессе разработки, отладки и тестирования компьютерных программ, пригодных для практического применения.	Уметь разрабатывать алгоритмы для конкретных задач; работать в среде программирования; реализовывать построенные алгоритмы в виде программ на конкретном языке программирования (Python); оформлять код программы в соответствии со стандартом кодирования (PEP8), выполнять проверку, отладку кода программы.	Перечень вопросов к зачету	Нет	Да
	Знать эволюцию языков программирования, их классификацию; основные элементы языка Python, операторы и операции, управляющие структуры, структуры данных, файлы; подпрограммы, составление библиотек подпрограмм; объектно-ориентированную модель программирования, основные принципы объектноориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.	Перечень вопросов к зачету	Нет	Да
Основные конструкции языков программирования				
ОПК-2.1 Алгоритмизирует решение задач и реализует алгоритмы с использованием программных средств.	Уметь разрабатывать алгоритмы для конкретных задач	Перечень вопросов к зачету	Нет	Да
	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Перечень вопросов к зачету	Нет	Да

ОПК-2.2 Принимает участие в процессе разработки, отладки и тестирования компьютерных программ, пригодных для практического применения.	Знать эволюцию языков программирования, их классификацию; основные элементы языка Python, операторы и операции, управляющие структуры, структуры данных, файлы; подпрограммы, составление библиотек подпрограмм; объектно-ориентированную модель программирования, основные принципы объектноориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.	Перечень вопросов к зачету	Нет	Да
	Уметь разрабатывать алгоритмы для конкретных задач; работать в среде программирования; реализовывать построенные алгоритмы в виде программ на конкретном языке программирования (Python); оформлять код программы в соответствии со стандартом кодирования (PEP8), выполнять проверку, отладку кода программы.	Перечень вопросов к зачету	Нет	Да
Введение в структурированное программирование				
ОПК-2.1 Алгоритмизирует решение задач и реализует алгоритмы с использованием программных средств.	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Перечень вопросов к зачету	Нет	Да
	Уметь разрабатывать алгоритмы для конкретных задач	Перечень вопросов к зачету	Нет	Да
ОПК-2.2 Принимает участие в процессе разработки, отладки и тестирования компьютерных программ, пригодных для практического применения.	Уметь разрабатывать алгоритмы для конкретных задач; работать в среде программирования; реализовывать построенные алгоритмы в виде программ на конкретном языке программирования (Python); оформлять код программы в соответствии со стандартом кодирования (PEP8), выполнять проверку, отладку кода программы.	Перечень вопросов к зачету	Нет	Да
	Знать эволюцию языков программирования, их классификацию; основные элементы языка Python, операторы и операции, управляющие структуры, структуры данных, файлы; подпрограммы, составление библиотек подпрограмм; объектно-ориентированную модель программирования, основные принципы объектноориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.	Перечень вопросов к зачету	Нет	Да
Введение в объектно-ориентированное программирование				

ОПК-2.1 Алгоритмизирует решение задач и реализует алгоритмы с использованием программных средств.	Уметь разрабатывать алгоритмы для конкретных задач	Перечень вопросов к зачету	Нет	Да
	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Перечень вопросов к зачету	Нет	Да
ОПК-2.2 Принимает участие в процессе разработки, отладки и тестирования компьютерных программ, пригодных для практического применения.	Уметь разрабатывать алгоритмы для конкретных задач; работать в среде программирования; реализовывать построенные алгоритмы в виде программ на конкретном языке программирования (Python); оформлять код программы в соответствии со стандартом кодирования (PEP8), выполнять проверку, отладку кода программы.	Перечень вопросов к зачету	Нет	Да
	Знать эволюцию языков программирования, их классификацию; основные элементы языка Python, операторы и операции, управляющие структуры, структуры данных, файлы; подпрограммы, составление библиотек подпрограмм; объектно-ориентированную модель программирования, основные принципы объектноориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.	Перечень вопросов к зачету	Нет	Да

1. Алгоритм это

- это устройство или одушевлённое существо (человек), способное понять и выполнить команды, составляющие алгоритм
- это точное описание порядка действий, которые должен выполнить исполнитель для решения задачи за конечное время

2. Выберите шесть свойств алгоритма:

- Наглядность
- Дискретность
- Детерминированность
- Понятность
- Конечность
- Окончателность
- Корректность
- Массовость

3. Как выглядит комментарий в Python:

- "это пустая программа"
- #это пустая программа
- 'это пустая программа'

4. Что такое "переменная"

- это величина, имеющая имя, тип и значение. Значение переменной нельзя изменять во время работы программы.
- это знак проверки условия
- это величина, значение которой нельзя менять во время работы программы.
- это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.

5. Какие типы переменных используются в Python:

- Строчные
- Символьная строка

- Вещественное число
- Кинетическое число
- Целое число
- Логическая

6. Команда для присваивания нового значения переменной

- Оператор
- Конъюнкция
- Оператор присваивания
- Условный оператор

7. Что будет в результате работы программы:

```
a=input()
```

```
b=input()
```

```
c=a+b
```

```
print (c)
```

Входные данные a=21 b=33

- 54
- 21
- 33
- 2133

8. Как ввести целое число с клавиатуры:

- a=input()
- a=int(input())
- a=10
- a=str(input())

9. Что будет в результате выполнения программы (чему равно a и b):

```
a=5
```

```
b=a+2
```

```
a=(a+2)*(b-3)
```

```
b=b+1
```

- $a=2 \ b=20$
- $a=8 \ b=28$
- $a=28 \ b=8$

10. Возведение в степень в Python:

- `//`
- `**`
- `%`
- `+`
- `^`

11. Целочисленное деление:

- `//`
- `%`
- `/`
- `*`

12. Остаток от деления:

- `//`
- `/`
- `%`
- `^`

13. Что будет в результате такого деления:

```
a=1234
```

```
d=a%10
```

```
print(d)
```

- 5
- 4
- 3
- 2

14. Что выведет вторая d:

```
a = 1234
```

```
d = a % 10; print( d )
```

```
a = a // 10
```

```
d = a % 10; print( d )
```

- 3
- 4
- 5
- 6

15.Как импортировать библиотеку случайных чисел:

- from random import
- random.randint
- random.random
- import random

16.Знак равенства в Python:

- !=
- ><
- ==
- <=

17.Что выведет на экран следующая программа

```
cost = 1500
```

```
if cost<1000:
```

```
    print("Скидок нет")
```

```
elif cost<2000:
```

```
    print("Скидка 2%")
```

```
elif cost<5000:
```

```
    print("Скидка 5%")
```

```
else:
```

```
    print("Скидка 10%")
```

- Скидок нет

- Скидка 5%
- Скидка 10%
- Скидка 2%
- программа написана неправильно

18.Как определить длину строки:

- `n=len(a)`
- `n=length(a)`
- `print(s[5])`

19.Команда которая определяет цифру в строке:

- `if c.isupper()`
- `if c.isdigit()`
- `if c.isalpha()`
- `if c.islower()`
- `if c in ["a","б"]`

20.Ввести строку и определить сколько в строке цифр:

- `s = input() k = 0 for c in s: if c in range(10) k -= 1`
- `s = input() k = 0 for c in s: if c.isdigit(): k -= 1`
- `s = input() k = 0 for c in s: if c.isdigit(): k += 1`
- `s = input() k = 0 for c in s: if c in range(10) k += 1`

21.Найти среднее арифметическое элементов массива, которые оканчиваются на цифру 5

- `count = 0 summa = 0 for x in A: if x % 10 == 5: count -= 1 summa -= x print ( summa/count )`
- `count = 0 summa = 0 for x in A: if x % 10 == 5: count += 1 summa += x print ( summa/count )`
- `count = 10 summa = 10 for x in A: if x % 10 == 5: count += 1 summa += x print ( summa/count )`
- `count = 30 summa = 40 for x in A: if x % 10 == 5: count += 1 summa += x print ( summa/count )`

22. В массиве записаны данные о росте баскетболистов. Сколько из них имеет рост больше 180 см, но меньше 190 см?

- `count = 0` for `x` in `A`: if `180 > x` and `x > 190`: `count += 1`
- `count = 0` for `x` in `A`: if `190 < x` and `x < 180`: `count += 1`
- `count = 0` for `x` in `A`: if `180 < x` and `x < 190`: `count += 1`

23. Что выведет программа:

Список:

`A = [2,3,1,4,6,5]`

Вывод:

`print( A[0]+2*A[5])`

- 14
- 10
- 11
- 12

24. Что выведет программа:

`A = [2,3,1,4,6,5]`

`for k in range(6):`

`A[k] += 2`

`print(2*A[3]+3*A[4])`

- 30
- 42
- 36
- 15

25. Как ввести числа в одной строке:

- `data = input() s = data.split() print(s)`
- `data = input() s = data.split() A=[ int(x) for x in s ]`
- `data = input() if s>x: A=[ int(x) for x in s ]`
- `data = input() s = data.math() A=[ int(x) for x in s ]`

26. Как правильно заполнить массив 20 числами (начиная с 0)

- $N = 10$ $A = [0]*N$ for i in $\text{range}(N)$: $A[i] = i$
- $N = 30$ $A = [0]*N$ for i in $\text{range}(N)$: $A[i] = i$
- $N = 20$ $A = [0]*N$ for i in $\text{range}(N)$: $A[i] = i$
- $N = 5$ $A = [0]*N$ for i in $\text{range}(N)$: $A[i] = i$

27. Что выведет программа

```
A = [2,3,1,4,6,5]
```

```
A[1] = A[0] + A[5]
```

```
print(3*A[1]+A[4])
```

- 20
- 21
- 7
- 27

28. Что выведет программа:

```
Дан массив: A = [4, 3, 6, 8, 1]
```

```
summ=0
```

```
for x in A:
```

```
    if x%2 == 0:
```

```
        summ+=x
```

```
print(summ)
```

- 18
- 20
- 22
- 15

29. Дан массив:

```
A = [2, 3, 1, 4, 6, 5]
```

```
print( A[0]+2*A[5] )
```

- 9
- 12
- 15

- 35

30. Укажите правильный приоритет операций:

- Варианты ответов
- Скобки
- Возведение в степень
- Умножение и деление
- Сложение и вычитание

31. Что выведет каждая из переменных X1, X2, X3, X4

$a = 9; b = 6$

$x1 = 3 // 4$

$x2 = a // b$

$x3 = -3 // 4$

$x4 = -a // b$

Показатели и критерии оценивания сформированности компетенций (промежуточного контроля)

На этапе промежуточной аттестации используется система оценки успеваемости обучающихся, которая позволяет преподавателю оценить уровень освоения материала обучающимися. Критерии оценивания сформированности планируемых результатов обучения (дескрипторов) представлены в карте компетенции ОПОП.

Форма оценки знаний: оценка - 5 «отлично»; 4 «хорошо»; 3 «удовлетворительно»; 2 «неудовлетворительно». Лабораторные работы, практические занятия, практика оцениваются: «зачет», «незачет». Возможно использование балльно-рейтинговой оценки.

Шкала оценивания:

«Зачет» – выставляется, если сформированность заявленных дескрипторов компетенций на 51 % и более оценивается не ниже «удовлетворительно» при условии отсутствия критерия «неудовлетворительно». Выставляется, когда обучающийся показывает хорошие знания изученного учебного материала; самостоятельно, логично и последовательно излагает и интерпретирует материалы учебного курса; полностью раскрывает смысл предлагаемого вопроса; владеет основными терминами и понятиями изученного курса; показывает умение переложить теоретические знания на предполагаемый практический опыт.

«Отлично» – выставляется, если сформированность заявленных дескрипторов компетенций 85 % более (в соответствии с картами компетенций ОПОП) 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 дескрипторов компетенций на 61 % и более (в соответствии с картами компетенций ОПОП) 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 дескрипторов компетенций 51 % и более (в соответствии с картами компетенций ОПОП) 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» «Незачет» – выставляется, если сформированность заявленных дескрипторов компетенций менее чем 51 % (в соответствии с картами компетенций ОПОП): 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.

Соответствие критериев оценивания сформированности планируемых результатов обучения (дескрипторов) системам оценок представлено в табл.

Таблица 11

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

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.