

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Заболотный, Г.И.
Должность: Директор филиала
Дата подписания: 02.10.2023 16:12:38
Уникальный программный ключ:
476db7d4accb36ef8130172be235477473d63457266ce26b7e9e40f733b8b08

МИНОБРНАУКИ 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»

Код и направление подготовки (специальность)	13.03.02 Электроэнергетика и электротехника
Направленность (профиль)	Электроэнергетика
Квалификация	Бакалавр
Форма обучения	Заочная
Год начала подготовки	2022
Институт / факультет	Кафедры филиала ФГБОУ ВО "СамГТУ" в г. Новокуйбышевске
Выпускающая кафедра	кафедра "Электроэнергетика, электротехника и автоматизация технологических процессов" (НФ- ЭЭиАТП)
Кафедра-разработчик	кафедра "Электроэнергетика, электротехника и автоматизация технологических процессов" (НФ- ЭЭиАТП)
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

Е.М. Шишков, кандидат
технических 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

Е.М. Шишков, кандидат
технических наук, доцент

(ФИО, степень, ученое звание)

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы	4
2. Место дисциплины (модуля) в структуре образовательной программы	7
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	8
4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий	8
4.1 Содержание лекционных занятий	8
4.2 Содержание лабораторных занятий	9
4.3 Содержание практических занятий	9
4.4. Содержание самостоятельной работы	9
5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)	11
6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения	11
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем	11
8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)	12
9. Методические материалы	12
10. Фонд оценочных средств по дисциплине (модулю)	14

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
			Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
			Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
			Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
			Уметь разрабатывать алгоритмы для конкретных задач
			Уметь разрабатывать алгоритмы для конкретных задач
			Уметь разрабатывать алгоритмы для конкретных задач
			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 объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции

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 объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.

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 объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.

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 объектно-ориентированного 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.

			<p>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.</p>
			<p>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.</p>
			<p>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.</p>

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

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	7 семестр часов / часов в электронной форме
Аудиторная контактная работа (всего), в том числе:	6	6
Лекции	2	2
Практические занятия	4	4
Самостоятельная работа (всего), в том числе:	136	136
выполнение задач, заданий, упражнений (в том числе разноуровневых)	88	88
подготовка к зачету	36	36
подготовка к лекциям	4	4
подготовка к практическим занятиям	8	8
Контроль	2	2
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	2	0	0	34	36
2	Основные конструкции языков программирования	0	0	4	34	38
3	Введение в структурированное программирование	0	0	0	34	34
4	Введение в объектно-ориентированное программирование	0	0	0	34	34
	Контроль	0	0	0	0	2
	Итого	2	0	4	136	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
7 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
Итого за семестр:				2
Итого:				2

4.2 Содержание лабораторных занятий

Учебные занятия не реализуются.

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
7 семестр				
1	Основные конструкции языков программирования	Операторы языка программирования Составление	Составление программ линейной структуры. Составление программ разветвляющейся структуры. Условный оператор. Составление программ разветвляющейся структуры. Оператор выбора.	2
2	Основные конструкции языков программирования	Функции и циклы	Цикл с предусловием, цикл с параметром	2
Итого за семестр:				4
Итого:				4

4.4. Содержание самостоятельной работы

Наименование раздела	Вид самостоятельной работы	Содержание самостоятельной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов
7 семестр			

Введение в программирование	Языки программирования	Эволюция языков программирования. Классификация программного обеспечения ПК	17
Введение в программирование	Типы данных	Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.	17
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.	17
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.	17
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.	17
Введение в структурированное программирование	Структуризация в программировании. Модульное программирование	Основы структурного программирования. Методы структурного программирования. Модульное программирование. Понятие модуля. Структура модуля. Стандартные модули.	17
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	17
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.	17
Итого за семестр:			136
Итого:			136

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	Python	PSF (Зарубежный)	Свободно распространяемое
2	PyCharm Community Edition	JetBrains (Зарубежный)	Свободно распространяемое
3	Jupyter Notebook	Open-source (Зарубежный)	Свободно распространяемое
4	Microsoft Windows	Microsoft (Зарубежный)	Лицензионное
5	Microsoft Office	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
---	--------------------------------	---	---------------------------

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	Заочная
Год начала подготовки	2022
Институт / факультет	Кафедры филиала ФГБОУ ВО "СамГТУ" в г. Новокуйбышевске
Выпускающая кафедра	кафедра "Электроэнергетика, электротехника и автоматизация технологических процессов" (НФ- ЭЭиАТП)
Кафедра-разработчик	кафедра "Электроэнергетика, электротехника и автоматизация технологических процессов" (НФ- ЭЭиАТП)
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
			Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
			Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
			Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
			Уметь разрабатывать алгоритмы для конкретных задач
			Уметь разрабатывать алгоритмы для конкретных задач
			Уметь разрабатывать алгоритмы для конкретных задач
			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 объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции

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 объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.

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 объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.

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 объектно-ориентированного 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.

		<p>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.</p>
		<p>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.</p>
		<p>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.</p>

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	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
	Уметь разрабатывать алгоритмы для конкретных задач			
	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции			
	Уметь разрабатывать алгоритмы для конкретных задач			
	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	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.			
	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 объектно-ориентированного 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.			

	<p>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 объектно- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции</p>			
	<p>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.</p>	Перечень вопросов к зачету	Нет	Да
	<p>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 объектно- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.</p>			
	<p>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.</p>			
Основные конструкции языков программирования				
ОПК-2.1 Алгоритмизирует решение задач и реализует алгоритмы с использованием программных средств	<p>Уметь разрабатывать алгоритмы для конкретных задач</p>			
		Перечень вопросов к зачету	Нет	Да
	<p>Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции</p>			
	<p>Уметь разрабатывать алгоритмы для конкретных задач</p>			

	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Перечень вопросов к зачету	Нет	Да
	Уметь разрабатывать алгоритмы для конкретных задач			
	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции			
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.			
	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 объектно- ориентированного 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.	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 объектно- ориентированного 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.			

	<p>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 объектно- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.</p>	Перечень вопросов к зачету	Нет	Да
	<p>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.</p>			
	<p>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 объектно- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.</p>			
Введение в структурированное программирование				
ОПК-2.1 Алгоритмизирует решение задач и реализует алгоритмы с использованием программных средств	<p>Уметь разрабатывать алгоритмы для конкретных задач</p>			
	<p>Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции</p>	Перечень вопросов к зачету	Нет	Да
	<p>Уметь разрабатывать алгоритмы для конкретных задач</p>			
	<p>Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции</p>			
	<p>Уметь разрабатывать алгоритмы для конкретных задач</p>	Перечень вопросов к зачету	Нет	Да
	<p>Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции</p>			
	<p>Уметь разрабатывать алгоритмы для конкретных задач</p>			

ОПК-2.2 Принимает участие в процессе разработки, отладки и тестирования компьютерных программ, пригодных для практического применения	<p>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 объектно- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.</p>	Перечень вопросов к зачету	Нет	Да
	<p>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.</p>			
	<p>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 объектно- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.</p>			
	<p>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.</p>			
	<p>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 объектно- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.</p>			
	<p>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.</p>	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.			
	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 объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции			
Введение в объектно-ориентированное 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			
	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции			
	Уметь разрабатывать алгоритмы для конкретных задач			
	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции			
	Уметь разрабатывать алгоритмы для конкретных задач			
	Знать понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции			
	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.			

<p>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 объектно- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.</p>			
	Перечень вопросов к зачету	Нет	Да
<p>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.</p>			
<p>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 объектно- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции</p>			
<p>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.</p>	Перечень вопросов к зачету	Нет	Да
<p>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 объектно- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и абстракции.</p>			