

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 32

УТВЕРЖДАЮ

Руководитель направления

доц., к.т.н., доц.

(должность, уч. степень, звание)

С.В. Солёный

(инициалы, фамилия)



(подпись)

«22» июня 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Алгоритмизация и программирование»
(Наименование дисциплины)

Код направления подготовки	13.03.02
Наименование направления подготовки	Электроэнергетика и электротехника
Наименование направленности	Цифровая энергетика
Форма обучения	заочная

Лист согласования рабочей программы дисциплины

Программу составил (а)

проф., д.т.н., доц.
(должность, уч. степень, звание)


(подпись, дата)

С.А. Сериков
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 32
«24» апреля 2023 г., протокол № 6

Заведующий кафедрой № 32


доц., к.т.н., доц.
(уч. степень, звание)


(подпись, дата)

С.В. Солёный
(инициалы, фамилия)

Ответственный за ОП ВО 13.03.02(03)

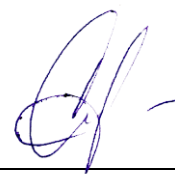
доц., к.т.н., доц.
(должность, уч. степень, звание)


(подпись, дата)

О.Я. Солёная
(инициалы, фамилия)

Заместитель директора института №3 по методической работе

старший преподаватель
(должность, уч. степень, звание)


(подпись, дата)

Н.В. Решетникова
(инициалы, фамилия)

Аннотация

Дисциплина «Алгоритмизация и программирование» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки 13.03.02 «Электроэнергетика и электротехника» направленности «Цифровая энергетика». Дисциплина реализуется кафедрой «№32».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

УК-1 «Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач»

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

Содержание дисциплины охватывает круг вопросов, связанных с изучением принципов, методов и инструментальных средств алгоритмизации, разработки и отладки программного обеспечения на языках программирования высокого уровня при решении задач в области электроэнергетики и электротехники.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы самостоятельная работа обучающегося.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 3 зачетных единицы, 108 часов.

Язык обучения по дисциплине «русский».

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Целью дисциплины является изучение принципов, методов и инструментальных средств алгоритмизации, разработки и отладки программного обеспечения на языках программирования высокого уровня при решении задач в области электроэнергетики и электротехники.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Универсальные компетенции	УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.Д.4 умеет использовать методы и средства анализа больших данных
Общепрофессиональные компетенции	ОПК-2 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-2.Д.1 выбирает языки программирования и среды разработки информационных систем и технологий ОПК-2.Д.2 разрабатывает алгоритмы и компьютерные программы, предназначенные для практического применения ОПК-2.Д.3 выполняет отладку и верификацию разработанных компьютерных программ

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- «Математика. Теория вероятностей и математическая статистика»,
- «Теория автоматического управления»,
- «Программное обеспечение мехатронных и робототехнических систем»,
- «Информационные технологии».

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и могут использоваться при изучении других дисциплин:

- «Управление роботами и робототехническими системами»,
- «Моделирование и исследование роботов и робототехнических систем»,
- «Системы с искусственным интеллектом».

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам
		№3
1	2	3
Общая трудоемкость дисциплины, ЗЕ/ (час)	3/ 108	3/ 108
Из них часов практической подготовки		
Аудиторные занятия, всего час.	16	16
в том числе:		
лекции (Л), (час)	6	6
практические/семинарские занятия (ПЗ), (час)		
лабораторные работы (ЛР), (час)	10	10
курсовой проект (работа) (КП, КР), (час)		
экзамен, (час)	9	9
Самостоятельная работа, всего (час)	83	83
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз.	Экз.

Примечание: ** кандидатский экзамен

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
Семестр 3					
Раздел 1. Язык программирования Питон (Python) и его возможности для использования в информационно – измерительных и управляющих робототехнических комплексах	0.5		1		5
Раздел 2. Типы и структуры данных в языке Python. Правила работы со структурами, их полями и методами. Алгебра логики. Системы счисления.	1		1		10
Раздел 3. Операторы языка Python	1		2		13
Раздел 4. Функции, модули в языке Python	1		2		15
Раздел 5. Принципы объектно-ориентированного программирования	1		2		20
Раздел 6. Организация ввода и вывода информации	1		1		10
Раздел 7. Использование подключаемых библиотек NumPy, Matplotlib, SciPy, Pandas, Scikit-learn при разработке робототехнических информационно-управляющих систем. Использование пакета OpenCV	0.5		1		10

Итого	6	0	10	0	83
-------	---	---	----	---	----

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	Язык программирования Питон (Python) и его возможности для использования в информационно – измерительных и управляющих робототехнических комплексах. Подготовка среды программирования. Получение помощи. Запуск программ Python. Первая программа. Переменные и выражения. Ввод / вывод. Форматированный ввод / вывод. Запись в файл, чтение из файла
2	Типы и структуры данных. Правила работы со структурами, их полями и методами. Алгебра логики. Системы счисления. Простые типы данных (целые числа, числа с плавающей точкой, строки комментариев). Структуры данных (списки, словари, кортежи, множества). Преобразование типов.
3	Операторы языка Python. Операторы отношения. Операторы if, if-else, if-elif-else, while, for, break, continue
4	Функции, модули. Обработка исключений. Локальные и глобальные переменные. Аргументы функции. Значения, возвращаемые функцией. Строки документации. Декораторы. Анонимные функции и их использование. Функции-генераторы.
5	Принципы объектно-ориентированного программирования. Создание класса. Наследование. Переменные (поля) класса и объекта. Методы класса и объекта. Метод <code>__init__</code> . Метод <code>__del__</code> .
6	Организация ввода и вывода информации. Работа с файлами и каталогами. Вывод графической информации. Обмен данными по протоколу HTTP, UDP. Обработка исключений.
7	Использование подключаемых библиотек NumPy, Matplotlib, SciPy, Pandas, Scikit-learn при разработке робототехнических информационно-управляющих систем. Использование пакета OpenCV

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Учебным планом не предусмотрено					
Всего					

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической	№ раздела
-------	---------------------------------	---------------------	---------------------	-----------

			подготовки, (час)	дисциплины
Семестр 3				
1	Подготовка среды программирования. Изучение возможностей дистрибутива Python Anaconda. Знакомство со средой программирования Spyder.	1		1
2	Типы и структуры данных.	1		2
3	Операторы языка Python	2		3
4	Организация ввода и вывода информации	2		4
5	Функции, модули в языке Python	2		5
6	Принципы объектно-ориентированного программирования	1		6
7	Использование подключаемых библиотек NumPy, Matplotlib, SciPy, Pandas, Scikit-learn при разработке робототехнических информационно-управляющих систем. Использование пакета OpenCV	1		7
Всего		10		

4.5. Курсовое проектирование/ выполнение курсовой работы
Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 3, час
1	2	3
Изучение теоретического материала дисциплины (ТО)	40	40
Курсовое проектирование (КП, КР)		
Расчетно-графические задания (РГЗ)		
Выполнение реферата (Р)		
Подготовка к текущему контролю успеваемости (ТКУ)	20	20
Домашнее задание (ДЗ)		
Контрольные работы заочников (КРЗ)	10	10
Подготовка к промежуточной аттестации (ПА)	13	13
Всего:	83	83

5. Перечень учебно-методического обеспечения

для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8– Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
УДК 004.438	Прохоренок Н.А. Python 3. Самое необходимое / Н.А. Прохоренок, В.А. Дронов. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2019. – 608 с.	
УДК 004.94	Аллен Б. Дауни Изучение сложных систем с помощью Python / пер. с англ. Д. А. Беликова. – М.: ДМК Пресс, 2019. – 160 с.	
УДК 004.43	Седер Наоми C28 Python. Экспресс-курс. 3-е изд. — СПб.: Питер, 2019. — 480 с.	
УДК: 681.3.07	Жерон Орельен. Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем, 2-е изд.: Пер. с англ.— СПб.: ООО «Диалектика», 2020.—1040 с.	
УДК 681.3.07	Рашка Себастьян, Мирджалили Вахид. Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2, 3-е изд. Пер. с англ. – СПб.: «Диалектика», 2020. – 848 с.	
УДК 004 ББК 32.973 П29	Петин В. А. Микрокомпьютеры Raspberry Pi. Практическое руководство. — СПб.: БХВ-Петербург, 2015. — 240 с.	

7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
www.guar.ru	Библиотека ГУАП
http://python.org .	Официальный сайт языка Python
https://www.anaconda.com/download/	Anaconda - Дистрибутив Python,

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
	Не предусмотрено

8.2. Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Лекционная аудитория	Б.М. 21-21
2	Компьютерный класс	Б.М. 31-04

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Экзамен	Список вопросов к экзамену; Примерный перечень вопросов для тестов.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«отлично» «зачтено»	– обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий.
«хорошо» «зачтено»	– обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий.

Оценка компетенции	Характеристика сформированных компетенций
5-балльная шкала	
«удовлетворительно» «зачтено»	<ul style="list-style-type: none"> – обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий.
«неудовлетворительно» «не зачтено»	<ul style="list-style-type: none"> – обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений.

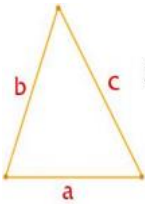
10.3. Типовые контрольные задания или иные материалы.

Вопросы для экзамена представлены в таблице 15.

Таблица 15 – Вопросы для экзамена

№ п/п	Перечень вопросов для экзамена	Код индикатора
1	<p>Тема: «Типы и структуры данных»</p> <p>1.1. Создайте скрипт, который обеспечивает:</p> <ul style="list-style-type: none"> – в ответ на запрос ввод с клавиатуры фамилии, имени и отчества одной строкой (<i>возможно, при неправильном использовании прописных и строчных букв</i>); – в ответ на запрос ввод с клавиатуры года поступления в университет; – вывод следующих сообщений: <ul style="list-style-type: none"> - фамилия: Ваша фамилия; - имя: Ваше имя; - отчество: Ваше отчество; - диплом бакалавра Вы получите в XXXX году; - защита магистерской диссертации может состояться в XXXX году; - псевдоним: Ваша фамилия с обратным следованием букв. <p><i>Примечание: фамилия, имя, отчество, псевдоним должны выводиться строчными буквами с первой прописной независимо от их ввода с клавиатуры. Для получения псевдонима используйте срез</i></p> <p>1.2. Свяжите переменную с любой строкой, состоящей не менее, чем из 15 символов. Извлеките из строки первый символ, затем последний, третий с начала и третий с конца. Измерьте длину строки. Извлеките из нее следующие срезы:</p> <ul style="list-style-type: none"> – первые восемь символов; – четыре символа из центра строки; 	УК-1.Д.4

	<p>– символы с индексами кратными трем. Определите количество вхождений первого символа в строку. Результаты выведете на экран.</p> <p>1.3. Создайте словарь, ставящий в соответствие идентификаторам пяти разных учебных групп количество студентов в этих группах. Выведите содержимое словаря на экран. Выведите информацию о количестве человек в какой либо группе. Внесите в словарь следующие изменения:</p> <ul style="list-style-type: none"> – в трёх группах изменилось количество студентов; – появились две новые группы; – одна из групп была расформирована. <p>Определите количество записей в словаре. Прдемонстрируйте проверку наличия записи в словаре. Выведите содержимое словаря на экран.</p> <p>1.4. Написать программу, которая определяет и выводит на экран позиции первого, второго и третьего вхождения буквы «у» в строку: <code>s = " " У лукоморья дуб зелёный, Златая цепь на дубе том. " "</code></p> <p>Определить сколько раз встречается буква «у» в строке.</p> <p>1.5. Напишите программу, принимающую от пользователя последовательность чисел, введенных с клавиатуры через запятую, и составляющую список и кортеж из этих чисел. Выведете на экран сумму первого и последнего элемента кортежа.</p> <p>1.6. Напишите программу, которая выводит на экран сумму цифр произвольного трёхзначного целого числа, введенного с клавиатуры</p>	
2	<p>Тема: «Операторы управления потоком вычислений»</p> <p>2.1. Напишите скрипт на языке программирования Python, выводящий ряд чисел Фибоначчи начиная с пятого члена ряда и заканчивая сотым.</p> <p>Числа Фибоначчи — ряд чисел, в котором каждое последующее число равно сумме двух предыдущих: 0, 1, 1, 2, 3, 5, 8, 13 и т.д.</p> <p>2.2. Напишите цикл, выводящий ряд четных чисел от 0 до 20. Затем, каждое третье число в ряде от -1 до -21.</p> <p>2.3. Напишите программу, которая запрашивает у пользователя значение вещественного аргумента x, а затем вычисляет и выводит на экран значение функции</p> $f = \begin{cases} x^2 & \text{при } -2,4 \leq x \leq 5,7, \\ 4 & \text{в противном случае.} \end{cases}$ <p>2.4. Напишите программу, определяющую и выводящую на печать все значения функции</p> $y(x) = x^2 + 3$	ОПК-2.Д.1

	<p>на интервале от 10 до 30 с шагом 2.</p> <p>2.5. С клавиатуры вводится произвольное число. Напишите программу, определяющую и выводящую на печать сумму квадратов четных цифр в числе.</p> <p>2.6. Напишите программу, которая предлагала бы пользователю решить пример 4*100-54.</p> <p>Если пользователь напишет правильный ответ, программа обеспечит выдачу поздравления и завершение работы.</p> <p>В противном случае – программа сообщит пользователю об ошибке и выдаст приглашение повторить ввод числа. При каждой попытке необходимо выдавать подсказки:</p> <ul style="list-style-type: none"> - «введено очень большое число», - «введено немного большее число», - «введено немного меньшее число», - «введено очень малое число». <p>При вводе пользователем ключевого слова «exit» программа обеспечивает выдачу соответствующего сообщения и завершение работы.</p>	
3	<p>3.1. Создайте функцию, переводящую градусы по шкале Цельсия (tc) в градусы по шкале Фаренгейта (tf) по формуле:</p> $tf = 9/5 * tc + 32.$ <p>Продемонстрируйте работу функции.</p> <p>3.2. Создайте функцию, принимающую в качестве аргумента список либо кортеж действительных чисел и возвращающую отношение их среднеарифметического значения к диапазону изменения элементов. Продемонстрируйте работу функции.</p> <p>3.3. Напишите функцию, осуществляющую проверку того, является ли палиндромом строка, переданная ей в качестве аргумента. Продемонстрируйте работу функции.</p> <p><i>(Палиндром это слово или фраза, которые одинаково читаются слева направо и справа налево)</i></p> <p>3.4. Найдите площадь треугольника с помощью формулы Герона.</p> <p>Длины сторон задаются вводом с клавиатуры. Реализовать вычисление площади необходимо в виде функции, на вход которой подаются три числа, на выходе – площадь.</p> <p>Функция находится в отдельном модуле, где происходит разделение между запуском и импортированием:</p> <ul style="list-style-type: none"> – при запуске модуля производится запрос у пользователя длин сторон, вычисляется площадь треугольника и её значение выводится на экран; – при импортировании модуля – запроса данных, вычисления площади и вывода значения на экран не происходит. <p>3.5. Напишите функцию, проверяющую все ли числа</p> <div style="display: flex; align-items: center; justify-content: center;">  <div style="margin-left: 20px;"> $S = \sqrt{p(p-a)(p-b)(p-c)}$ $p = \frac{a+b+c}{2}$ </div> </div>	ОПК-2.Д.2

	<p>последовательности, которая передаётся ей в качестве аргумента, уникальны. Продемонстрируйте работу функции.</p>	
4	<p>4.1. Функция <code>time()</code> из модуля <code>time</code> возвращает вещественное число, представляющее собой количество секунд, прошедшее с 1 января 1970 года. Напишите программу, отображающую это время в виде: дни : часы : минуты : секунды.</p> <p>4.2. Функция <code>localtime()</code> из модуля <code>time</code> при её вызове без указания параметра возвращает объект, представляющий собой локальное время. Возвращаемый функцией объект содержит следующие атрибуты (указаны тройки вида: имя атрибута – индекс – описание):</p> <ul style="list-style-type: none"> <code>tm_year</code> – 0 – год; <code>tm_mon</code> – 1 – месяц (число от 1 до 12); <code>tm_mday</code> – 2 – день месяца (число от 1 до 31); <code>tm_hour</code> – 3 – час (число от 0 до 23); <code>tm_min</code> – 4 – минуты (число от 0 до 59); <code>tm_sec</code> – 5 – секунды (число от 0 до 59, изредка до 61); <code>tm_wday</code> – 6 – день недели (число от 0 для понедельника до 6 для воскресенья); <code>tm_yday</code> – 7 – количество дней, прошедшее с начала года (число от 1 до 366); <code>tm_isdst</code> – 8 – флаг коррекции летнего времени (значения 0,1 или -1). <p>Напишите программу, выводящую день недели, дату и время таким образом, что бы день недели и месяц были написаны по-русски.</p> <p><i>Например: Сегодня: пятница 20 декабря 2019 года 18:45:40</i></p> <p>Тема: «Объектно-ориентированное программирование»</p> <p>4.3. Создайте класс <code>Cat</code>. Определите атрибуты <code>name</code> (имя), <code>color</code> (цвет) и <code>weight</code> (вес). Добавьте метод под названием <code>meow</code> («мяуканье»). Создайте два объекта класса <code>Cat</code>. Установите атрибуты объектов. Вызовите метод <code>meow</code> для объектов. Выведите на печать значения атрибутов объектов.</p> <p>4.4. Напишите код, описывающий класс <code>Animal</code>:</p> <ol style="list-style-type: none"> a). Добавьте атрибут имени животного. b). Добавьте конструктор класса <code>Animal</code>, выводящий: <ul style="list-style-type: none"> «Родилось животное с именем “Имя животного”». Если имя не задано – вместо него выводится ”Animal”. c). Добавьте метод <code>makeNoise()</code>, выводящий: <ul style="list-style-type: none"> «”Имя животного” говорит Гррр». d). Добавьте метод <code>eat()</code>, выводящий «Ням-Ням». e). Добавьте методы <code>getName()</code> и <code>setName()</code>, обеспечивающие получение и присвоение имени. <p>Пусть класс <code>Animal</code> будет родительским для класса <code>Cat</code>. Метод <code>makeNoise()</code> класса <code>Cat</code> выводит «”Имя животного” говорит Мяу». Конструктор класса <code>Cat</code> выводит «Родился кот», а также вызывает родительский конструктор.</p>	ОПК-2.Д.3

	<p>Пусть Animal будет родительским для класса Dog. Метод makeNoise () для Dog выводит «Имя животного» <i>говорит Гав</i>». Конструктор Dog выводит «<i>Родилась собака</i>», а также вызывает родительский конструктор.</p> <p><u>Основная программа:</u></p> <ul style="list-style-type: none"> – код, создающий кота, двух собак и одно простое животное. Дайте имя каждому животному. Одной собаке имя даётся аргументом при создании, остальным животным через вызов соответствующих методов. – код, вызывающий eat () и makeNoise () для каждого животного. 	
--	--	--

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.
Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
	Учебным планом не предусмотрено	

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения курсовой работы
	Учебным планом не предусмотрено

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов	Код индикатора
1	<p>1.1 Какой тип данных в <i>Python</i> представляет целые числа? Варианты ответов: а) complex, б) float, в) int, г) bool. Ответ: в</p> <p>1.2 Какие из представленных литералов чисел относятся к типу float? Варианты ответов: а) 1.7+4.3j, б) 5.0, в) 88, г) -.4 Ответ: б, г</p> <p>1.3 Что будет выведено на экран в результате выполнения следующего кода? x = 5; y = 2 z = x//y print(z) Варианты ответов: а) 2.5, б) 2, в) 3, г) ошибка. Ответ: б</p>	УК-1.Д.4

1.4 Что будет выведено на экран в результате выполнения инструкции

```
print(4 + 3.0) ?
```

Варианты ответов: а) 7.0, б) 4 + 3.0, в) 7, г) **ошибка**.

Ответ: а

1.5 Какое число будет выведено на экран в результате выполнения следующего кода?

```
x = 5; y = 2
z = x%y
print(z)
```

Варианты ответов: а) 2.5, б) 1, в) 0.1, г) **ошибка**.

Ответ: б

1.6 Что будет выведено на экран в результате выполнения следующего кода?

```
x = 5
x += x - x
print(x)
```

Варианты ответов: а) 5, б) 0, в) 10, г) **ошибка**.

Ответ: а

1.7 Выберите литералы чисел, представленные в шестнадцатеричной системе счисления.

Варианты ответов: а) 0X755, б) 16755, в) 0x1101, г) 0o1675.

Ответ: а, в

1.8 Что будет выведено на экран в результате выполнения следующего кода?

```
x = 3; y = 3.0; z = '3'
print(x + y + z)
```

Варианты ответов: а) 9.0, б) 9, в) 6.0 + 3, г) **ошибка**.

Ответ: г

1.9 Каким будет результат преобразования типа для инструкции `bool(-1)` ?

Варианты ответов: а) **False**, б) **True**, в) **None**, г) 1.

Ответ: б

1.10 Что будет выведено на экран в результате выполнения следующего кода?

```
a = [1, 2, 3]
b = a
a[0] = 10
```

```
print(b)
```

Варианты ответов:

а) 10, б) [1, 2, 3], в) [10, 2, 3], г) ошибка.

Ответ: в

1.11 Что будет выведено на экран в результате выполнения следующего кода?

```
x = '7'; y = '3'
z = x + y
print(z)
```

Варианты ответов: а) 7 + 3, б) 10, в) 73, г) ошибка.

Ответ: в

1.12 Выберите нужное описание функции `lst.extend(x)`.

Варианты ответов:

- а) добавление элемента `x` в конец списка `lst`;
- б) добавление кортежа или списка `x` в конец списка `lst`;
- в) определение количества элементов, равных `x`, в списке `lst`.

Ответ: а

1.13 В каком варианте представлена встроенная функция, которая обеспечивает применение одной и той же операции к элементам одного или нескольких списков или кортежей?

Варианты ответов: а) `range(n)`, б) `zip()`, в) `map()`.

Ответ: в

1.14 Что будет выведено на экран в результате выполнения следующего кода?

```
d = {1: 'one'}
d.update([(1, 'один'), (2, 'два')])
d[3] = 'три'
print(d)
```

Варианты ответов: а) {1: 'one', 2: 'два', 3: 'три'},
б) {},
в) {1: 'один', 2: 'два', 3: 'три'},
г) ошибка.

Ответ: в

1.15 Как удалить элемент `elem` из множества `st`?

Варианты ответов: а) `st.remove(elem)`, б) `st.discard(elem)`,
в) `st.pop(elem)`, г) никак.

Ответ: а, б

2

2.1 Какие из частей условной инструкции не являются обязательными?

ОПК-2.Д.1

Варианты ответов: а) `if`, б) `elif`,
в) `else`, г) **все части необязательны.**

Ответ: б, в

2.2 Что будет выведено на экран в результате выполнения следующего кода?

```
a, b = 5, 10
if a>b:
    print("a>b")
else:
    print("a<=b")
```

Варианты ответов: а) `a>b`, б) `a>=b`, в) `a<b`, г) `a<=b`.

Ответ: г

2.3 Какой из предлагаемых операторов можно использовать для инвертирования условия в инструкции `if`?

Варианты ответов: а) `and`, б) `or`, в) `not`, г) `is`.

Ответ: в

2.4 Что будет выведено на экран в результате выполнения следующего кода?

```
res = 0
if []: res += 1
if False: res += 1
if not None: res += 1
if -1: res += 1
print(res)
```

Варианты ответов: а) 1, б) 2, в) 3, г) **ошибка.**

Ответ: б

2.5 Что означает данная конструкция?

```
if x != -8:
    print('done')
```

Варианты ответов:

- а) если переменная `x` строго равна `-8`, то вывод строки `'done'`;
- б) если переменная `x` равна `-8`, то вывод строки `'done'`;
- в) если переменная `x` не равна `-8`, то вывод строки `'done'`;

Ответ: в

2.6 Что будет выведено на экран в результате выполнения следующего кода?

```
for i in range(3):
    print(-i, end=' ')
```

Варианты ответов: а) `-1 -2 -3`, б) `0 -1 -2 -3`,
в) `-1 -2 -3`, г) `0 -1 -2`.

Ответ: г

	<p>2.7 Какой из следующих операторов предназначен для прерывания цикла?</p> <p>Варианты ответов: а) <code>pass</code>, б) <code>continue</code>, в) <code>break</code>, г) <code>else</code>.</p> <p>Ответ: в</p> <p>2.8 Что будет выведено на экран в результате выполнения следующего кода?</p> <pre>d = [('a', 'b', 'c'), ('d', 'e', 'f')] for x, *y in d: print(x, y, end=' ')</pre> <p>Варианты ответов: а) <code>a (b, c) d (e, f)</code>, б) <code>a ('b', 'c') d ('e', 'f')</code>, в) <code>a [b, c] d [e, f]</code>, г) <code>a ['b', 'c'] d ['e', 'f']</code>.</p> <p>Ответ: а</p> <p>2.9 Что будет выведено на экран в результате выполнения следующего кода?</p> <pre>k = 4 while k: k -= 1 print(k, end=' ') if k>2: continue else: print(0)</pre> <p>Варианты ответов: а) <code>2 1 0</code>, б) <code>3 2 1 0</code>, в) <code>3 2 1 0 0</code>, г) <code>ошибка</code>.</p> <p>Ответ: в</p>	
3	<p>3.1 Что возвращает функция, в которой отсутствует инструкция <code>return</code>?</p> <p>Варианты ответов: а) <code>объект функции</code>, б) <code>None</code>, в) <code>False</code>, г) <code>ничего</code>.</p> <p>Ответ: б</p> <p>3.2 Что будет выведено на экран в результате выполнения следующего кода?</p> <pre>def my_func_1(): print(1, end=' ') my_func_2() def my_func_2(): print(2, end=' ') my_func_1()</pre> <p>Варианты ответов: а) <code>1 2</code>, б) <code>2 1</code>, в) <code>1</code>, г) <code>ошибка</code>.</p> <p>Ответ: г</p> <p>3.3. Что будет выведено на экран в результате выполнения следующего кода?</p> <pre>def func(*a, **b): print(a, b)</pre>	ОПК-2.Д.2

```
func(1, x=3, y=4)
```

Варианты ответов: а) (1,) {'x': 3, 'y': 4},
б) 1, 3, 4,
в) (1, 3, 4),
г) **ошибка.**

Ответ: а

3.4 Что будет выведено на экран в результате выполнения следующего кода?

```
x = 10
def func():
    global x
    x = 20

func()
print(x)
```

Варианты ответов: а) 10, б) 20, в) 30, г) **ошибка.**

Ответ: б

3.5 Что будет выведено на экран в результате выполнения следующего кода?

```
def func_1():
    x = 20
    def func_2():
        nonlocal x
        x = 30
    print(x)

func_1()
```

Варианты ответов: а) 20, б) 30, в) **ничего**, г) **ошибка.**

Ответ: а

3.6 Какой из предлагаемых операторов можно использовать для создания анонимной функции?

Варианты ответов: а) **def**,
б) **lambda**,
в) **yield**,
г) **все вышеуказанные.**

Ответ: б

3.7 Что будет выведено на экран в результате выполнения следующего кода?

```
ld = lambda x: if x>0: x**2
print(ld(3.0))
```

Варианты ответов: а) 9, б) 6.0, в) 9.0, г) **ошибка.**

Ответ: г

3.8 Что делает данная функция?

```
x = input('FF=')
```

	<p>Варианты ответов:</p> <ul style="list-style-type: none"> а) ввод значения переменной, тип переменной целое число; б) ввод значения переменной, тип переменной строка; в) вывод значения переменной. <p>Ответ: б</p> <p>3.9 Значениями по умолчанию могут быть снабжены параметры, находящиеся:</p> <p>Варианты ответов: а) в начале списка параметров; б) в конце списка параметров.</p> <p>Ответ: б</p>	
4	<p>4.1 Что будет выведено на экран в результате выполнения следующего кода?</p> <pre>def f_2(n): for i in range(1, n+1): yield i gen = f_2(30) res = next(gen) + next(gen) + next(f_2(30)) print(res)</pre> <p>Варианты ответов: а) 4, б) 3, в) 2, г) 1.</p> <p>Ответ: а</p> <p>4.2 Выберите верное утверждение, касающееся функций:</p> <p>Варианты ответов:</p> <ul style="list-style-type: none"> а) после того как функция определена, можно ее вызывать, передавая в квадратных скобках различные аргументы; б) переменные, объявленные внутри определения функции, являются глобальными; в) при определении одной функции внутри другой для изменения переменной внешней функции из внутренней используется зарезервированное слово nonglobal; г) при определении одной функции внутри другой для изменения переменной внешней функции из внутренней используется зарезервированное слово nonlocal. <p>Ответ: г</p> <p>4.3 Что будет выведено на экран в результате выполнения следующего кода?</p> <pre>def f(x): '''5''' return x res = f('3') + f.__doc__ print(res)</pre> <p>Варианты ответов: а) 3 + 5, б) 35,</p>	ОПК-2.Д.3

	<p style="text-align: center;">в) <code>3 + '''5'''</code>, г) ошибка.</p> <p>Ответ: б</p> <p>4.4 Что будет выведено на экран в результате выполнения следующего кода?</p> <pre style="text-align: center;">def factorial(n): if n == 0 or n == 1: return 1 else: return n*factorial(n - 1) print(factorial(5))</pre> <p>Варианты ответов: а) 0, б) 25, в) 120, г) ошибка.</p> <p>Ответ: в</p> <p>4.5 Какие функции в <i>Python</i> называют рекурсивными?</p> <p>Варианты ответов:</p> <ul style="list-style-type: none"> а) функция, которая вызывает саму себя внутри своего тела, б) функция, которая вызывает другую функцию внутри своего тела, в) функция, которая одновременно не имеет параметров и возвращаемого значения, г) рекурсивные функции в <i>Python</i> запрещены. <p>Ответ: а</p> <p>4.6 Что будет выведено на экран в результате выполнения следующего кода?</p> <pre style="text-align: center;">ld = lambda x, y=5: x/y if y != 0 else None print(ld(20))</pre> <p>Варианты ответов: а) 20, б) 4.0, в) None, г) ошибка.</p> <p>Ответ: б</p>
--	---

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
1	Изучение возможностей дистрибутива Python Anaconda. Знакомство со средой программирования Spyder. Настройка среды
2	Типы и структуры данных. Строки, списки, кортежи, словари, множества. Организация ввода и вывода информации
3	Операторы языка Python. Условная инструкция if/elif/else. Циклы for и while.
4	Функции и модули,.
5	Объектно-ориентированное программирование
6	Использование подключаемых библиотек NumPy, Matplotlib, SciPy, Pandas, Scikit-learn при разработке систем обработки информации

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

11.1. Методические указания для обучающихся по освоению лекционного материала

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

Лекция состоит из трёх основных частей: вступительной, основной и заключительной.

Вступительная часть определяет название темы, план и цель лекции. Она призвана заинтересовать и настроить аудиторию. В этой части лекции излагается актуальность, основная идея, связь данной лекции с предыдущими занятиями, ее основные вопросы. Введение должно быть кратким и целенаправленным.

В основной части лекции реализуется научное содержание темы, все главные узловые вопросы, проводится вся система доказательств с использованием наиболее целесообразных методических приёмов. Каждый учебный вопрос заканчивается краткими выводами, логически подводящими студентов к следующему вопросу лекции.

Заключительная часть имеет целью обобщать в кратких формулировках основные идеи лекции, логически завершая её как целостное творение.

Отдельные виды лекций могут иметь свои особенности как по содержанию, так и по структуре.

11.2. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

Лабораторные работы, предусмотренные в данном курсе, выполняются в компьютерном классе на персональном компьютере с использованием пакетов поддержки разработки и отладки программного обеспечения на языке Python.

В процессе подготовки к лабораторной работе необходимо изучить соответствующие методические указания и повторить лекционный материал, который относится к теме работы.

Перед началом выполнения работы необходимо создать отдельную папку для создаваемых в работе файлов и установить её в системе качестве текущей директории. Путь к данной папке не должен содержать имён, написанных кириллицей.

В процессе выполнения работы полученные результаты расчётов, листинги разрабатываемых программ, схемы и другие рабочие материалы должны сохраняться на диске для их дальнейшего использования при оформлении отчёта.

По окончании работы необходимо составить отчёт и подготовиться к его защите на следующем занятии.

Структура и форма отчета о лабораторной работе

Отчёт о лабораторной работе должен содержать:

- фамилию, имя и отчество студента, выполнившего работу;
- номер учебной группы;
- дату выполнения работы;
- название работы;
- цель работы;
- краткую формулировку задания на лабораторную работу;
- основные теоретические сведения и формулы, использовавшиеся в процессе выполнения работы;
- листинги программ, разработанные и отлаженные в процессе выполнения работы;
- схемы, разработанные в процессе выполнения работы;
- при использовании дополнительной литературы указать ссылки и привести список литературы;
- Выводы по работе.

Требования к оформлению отчета о лабораторной работе

Пример оформления титульного листа отчёта по лабораторной работе приведен на сайте университета ww.guap.ru.

При оформлении отчёта о лабораторной работе необходимо придерживаться требований ГОСТ 7.32-2017 «Отчёт о научно-исследовательской работе. Структура и правила оформления».

По каждой лабораторной работе должен быть подготовлен отчёт в бумажном и в электронном виде. После защиты лабораторных работ отчёты в бумажном виде с проставленными оценками хранятся на кафедре, а отчёты в электронной форме должны быть выложены в личном кабинете учащегося на сайте университета.

11.3. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся, являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

11.4. Методические указания для обучающихся по прохождению текущего контроля успеваемости

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Контроль качества знаний проводится в форме индивидуального собеседования по материалу отдельных разделов дисциплины, а также проверки отчётов о выполнении практических заданий.

Результаты текущего контроля могут учитываться при проведении промежуточной аттестации. Положительный результат текущего контроля успеваемости дает студенту дополнительный балл при проведении промежуточной аттестации. Результаты текущего контроля могут учитываться при проведении промежуточной аттестации.

Система оценок при проведении промежуточной аттестации осуществляется в соответствии с требованиями Положений «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программы высшего образования» и «О модульно-рейтинговой системе оценки качества учебной работы студентов в ГУАП».

Текущий контроль успеваемости проводится после завершения изучения каждого раздела дисциплины. Методы ТКУ в зависимости от изучаемого материала: проведение проверочных работ в виде решения задач или тестирование в системе LMS. Примерный перечень вопросов для тестирования, представленный в таблице 18, формируются исходя из содержания пройденного раздела. О конкретной дате ТКУ, методе проведения ТКУ, условиях успешного прохождения ТКУ преподаватель сообщает не позднее одной недели до текущего контроля успеваемости.

11.5. Методические указания для обучающихся по прохождению промежуточной аттестации

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она проводится в форме экзамена:

Экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Система оценок при проведении промежуточной аттестации осуществляется в соответствии с требованиями Положений «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программам высшего образования» и «О модульно-рейтинговой системе оценки качества учебной работы студентов в ГУАП».

Промежуточная аттестация оценивается по результатам текущего контроля успеваемости. Список вопросов (таблица 15) к промежуточной аттестации утверждается кафедрой и выдается студентам для ознакомления. В случае, если студент по уважительной причине не выполнил требования текущего контроля, ему предоставляется возможность сдать задолженности по пропущенным темам. Форма проведения промежуточной аттестации – письменная.

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой