

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Наумова Наталия Александровна

Должность: Ректор

Дата подписания: 24.10.2024 14:21:41

Уникальный идентификатор документа:

6b5279da4e034bffa679172803da5b7b590c6922

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПРОСВЕЩЕНИЯ»

(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПРОСВЕЩЕНИЯ)

Экономический факультет

Кафедра современных промышленных технологий, робототехники и компьютерной графики

Согласовано

деканом факультета

«21» июня 2023 г.

 /Фонина Т.Б./

Рабочая программа дисциплины

Программирование на языке Python

Направление подготовки

44.03.05 Педагогическое образование (с двумя профилями подготовки)

Профиль:

Технология и дополнительное образование

Квалификация

Бакалавр

Форма обучения

Очная

Согласовано учебно-методической комиссией
экономического факультета

Протокол «20» июня 2023 г. № 11

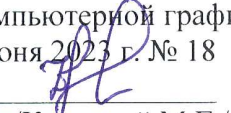
Председатель УМКом


/Сюзева О.В./

Рекомендовано кафедрой современных
промышленных технологий,
робототехники и компьютерной графики

Протокол от «13» июня 2023 г. № 18

Зав. кафедрой


/Корецкий М.Г./

Мытищи

2023

Автор-составитель:

Корецкий М.Г., кандидат педагогических наук, доцент, заведующий кафедрой современных промышленных технологий, робототехники и компьютерной графики Государственного университета просвещения

Лисевский А.А., ассистент кафедры современных промышленных технологий, робототехники и компьютерной графики Государственного университета просвещения.

Рабочая программа дисциплины «Программирование на языке Python» составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), утвержденного приказом МИНОБРНАУКИ России от 22.02.2018 г. № 125.

Дисциплина входит в часть, формируемую участниками образовательных отношений Блока 1 «Дисциплины (модули)» и является элективной дисциплиной.

Год начала подготовки (по учебному плану) 2023

СОДЕРЖАНИЕ

1. Планируемые результаты обучения.....	4
2. Место дисциплины в структуре образовательной программы.....	4
3. Объем и содержание дисциплины.....	4
4. Учебно-методическое обеспечение самостоятельной работы обучающихся.....	8
5. Фонд оценочных средств для проведения текущей и промежуточной аттестации по дисциплине.....	13
6. Учебно-методическое и ресурсное обеспечение дисциплины.....	29
7. Методические указания по освоению дисциплины.....	30
8. Информационные технологии для осуществления образовательного процесса по дисциплине.....	30
9. Материально-техническое обеспечение дисциплины.....	30

1. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

1.1. Цель и задачи дисциплины

Целью освоения дисциплины является формирование у студентов навыков, соответствующих видам профессиональной деятельности, на которые ориентирована программа, необходимых для решения следующих профессиональных задач.

Задачи дисциплины:

- построение математических моделей и исследование их аналитическими методами, разработка алгоритмов, методов, программного обеспечения, инструментальных средств по тематике проводимых научно-исследовательских проектов;
- разработка и применение современных высокопроизводительных вычислительных технологий, применение современных суперкомпьютеров в проводимых исследованиях;
- применение наукоемких математических и информационных технологий и пакетов программ для решения прикладных задач в области физики, химии, биологии, экономики, медицины, экологии;
- разработка архитектуры, алгоритмических и программных решений системного и прикладного программного обеспечения;
- развитие и использование математических и информационных инструментальных средств, автоматизированных систем в научной и практической деятельности.

1.2. Планируемые результаты обучения

В результате освоения данной дисциплины у обучающихся формируются следующие компетенции:

СПК-2. Способен организовывать содержательную практическую деятельность обучающихся с наукоемкой междисциплинарной и метадисциплинарной составляющей на основе применения личностно-ориентированного подхода, направленную на конструирование и программирование робототехнических комплексов.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина входит в часть, формируемую участниками образовательных отношений Блока 1 «Дисциплины (модули)» и является элективной дисциплиной.

Для освоения дисциплины «Программирование на языке Python» необходимы знания, умения и навыки приобретенные на предыдущих уровнях образования при изучении таких дисциплин как: «Методы математической обработки данных», «Основы робототехники», «Основы 3D-моделирования».

Во время освоения дисциплины «Программирование на языке Python» у обучающихся формируются знания, умения, навыки, которые можно использовать в процессе изучения дисциплин «Основы автоматизации и электроники».

3. ОБЪЕМ И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3.1. Объем дисциплины

Показатель объема дисциплины	Форма обучения
	Очная
Объем дисциплины в зачетных единицах	2
Объем дисциплины в часах	72
Контактная работа:	48.2
Лекции	24
Практические занятия	24
из них, в форме практической подготовки	24

Контактные часы на промежуточную аттестацию:	0,2
Зачет	0,2
Самостоятельная работа	16
Контроль	7,8

Форма промежуточной аттестации – зачет в 7 семестре.

3.2. Содержание дисциплины

Наименование тем дисциплины с кратким содержанием	Количество часов		
	Лекции	Практические занятия	
		Общее кол-во	Из них, в форме практической подготовки
Тема 1. История создания. Области применения и перспективы. Инструкции и структура программы..	2	2	2
Тема 2. Установка Python. Доступ к документации. Ввод и вывод данных. Первая программа на языке Python.	2	2	2
Тема 3. Переменные: именованное переменных, присваивание значения переменным, удаление переменных. Операторы: математические операторы, двоичные операторы, приоритет выполнения операторов.	2	2	2
Тема 4. Типы данных: числовые типы, строки, списки, кортежи, множества, диапазоны, словари. Операторы для работы с последовательностями и отображениями.	2	2	2
Тема 5. Простые и составные инструкции в Python. Условные операторы и циклы.	2	2	2
Тема 6. Встроенные и пользовательские функции. Итераторы и функции-генераторы. Декораторы функций.	2	2	2
Тема 7. Объектно-ориентированное программирование. Основы программирования классов. Создание экземпляров класса. Конструктор и деструктор класса. Основы программирования классов. Создание экземпляров класса. Конструктор и деструктор класса. Перегрузка операторов, Наследование. Композиция.	4	4	4
Тема 8. Графический интерфейс пользователя (GUI). Событийно-ориентированное программирование. Инструменты для создания графического интерфейса пользователя: модуль стандартной библиотеки tkinter, фреймворк PyQt.	4	4	4

Тема 9. Web-фреймворк Flask. Разработка web-приложения.	4	4	4
Итого:	24	24	24

ПРАКТИЧЕСКАЯ ПОДГОТОВКА

Тема	Задание на практическую подготовку	Количество часов
Тема 1. История создания. Области применения и перспективы. Инструкции и структура программы.	1. Изучить история создания языка программирования Python. 2. Изучить области применения и перспективы языка программирования Python. 3. Изучить инструкции и структура программы на языке программирования Python	2
Тема 2. Установка Python. Доступ к документации. Ввод и вывод данных. Первая программа на языке Python.	1. Изучить этапы установки Python. 2. Изучить способы доступа к документации языка Python 3. Напишите программу, которая выводит на экран текст «Здравствуй, мир!» (без кавычек).. 4. В популярном сериале «Остаться в живых» использовалась последовательность чисел 4 8 15 16 23 42, которая принесла героям удачу и помогла сорвать джекпот в лотерее. Напишите программу, которая выводит данную последовательность чисел с одним пробелом между ними. 5. Измените предыдущую программу так, чтобы каждое число последовательности 4 8 15 16 23 42 печаталось на отдельной строке.	2
Тема 3. Переменные: именованное присваивание переменным, удаление переменных. Операторы: математические операторы, двоичные операторы, приоритет выполнения операторов.	1. Напишите программу вывода на экран трех последовательно идущих чисел, каждое на отдельной строке. Первое число вводит пользователь, остальные числа вычисляются в программе. 2. Напишите программу, которая считывает три целых числа и выводит на экран их сумму. Каждое число записано в отдельной строке. 3. Напишите программу, вычисляющую объём куба и площадь его полной поверхности по введённому значению длины ребра 4. Напишите программу, которая считывает целое число, после чего на экран выводится следующее и предыдущее целое число с пояснительным текстом	2

Тема 4. Типы данных: числовые типы, строки, списки, кортежи, множества, диапазоны, словари. Операторы для работы с последовательностями и отображениями.	1. Напишите программу, которая находит полное число метров по заданному числу сантиметров 2. Напишите программу для пересчёта величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах	2
Тема 5. Простые и составные инструкции в Python. Условные операторы и циклы.	1. Пользователь вводит два числа с клавиатуры. Вывести на экран yes, если они отличаются друг от друга на 135, иначе вывести на экран No 2. Дано число. Если оно больше 100 или меньше -100, то вывести на экран символ —, иначе вывести на экран символ + 3. Пользователь вводит номер месяца (от 1 до 12). Вывести название сезона года на экран (зима, весна, лето, осень); 4. Дано три числа. Найти количество положительных чисел среди них	2
Тема 6. Встроенные и пользовательские функции. Итераторы и функции-генераторы. Декораторы функций.	1.Реализуйте итератор колоды карт (52 штуки) CardDeck. Каждая карта представлена в виде строки типа 2 Пик. При вызове функции next() будет представлена следующая карта. По окончании перебора всех элементов возникнет ошибка StopIteration 2. Создайте функцию infinite(lst, tries), которая будет проходить по элементам списка lst (целые числа) заданное количество раз (tries) циклически. Один раз - один элемент списка. После вывода последнего значения последовательности процедура начнется с самого начала. Например, если в списке 2 элемента, а функция получила значение 3, то сначала выведется первый объект, потом последний, а потом опять первый. Результат работы функции представьте в виде строки, состоящей из tries количества символов	2
Тема 7. Объектно-ориентированное программирование. Основы программирования классов. Создание экземпляров класса. Конструктор и деструктор класса. Основы программирования классов. Создание экземпляров класса. Конструктор и	1. Создайте класс Soda (для определения типа газированной воды), принимающий 1 аргумент при инициализации (отвечающий за добавку к выбираемому лимонаду). В этом классе реализуйте метод show_my_drink(), выводящий на печать Газировка и {ДОБАВКА} в случае наличия добавки, а иначе отобразится следующая фраза: Обычная газировка. 2. Николаю требуется проверить, возможно	4

деструктор класса. Перегрузка операторов, Наследование. Композиция.	ли из представленных отрезков условной длины сформировать треугольник. Для этого он решил создать класс TriangleChecker, принимающий только положительные числа. С помощью метода is_triangle() возвращаются следующие значения (в зависимости от ситуации): – Ура, можно построить треугольник!; – С отрицательными числами ничего не выйдет!; – Нужно вводить только числа!; – Жаль, но из этого треугольник не сделать.	
Тема 8. Графический интерфейс пользователя (GUI). Событийно-ориентированное программирование. Инструменты для создания графического интерфейса пользователя: модуль стандартной библиотеки tkinter, фреймворк PyQt.	1. Разработка игры «Тетрис» при помощи библиотеки tkinter и фреймворка PyQt	4
Тема 9. Web-фреймворк Flask. Разработка web-приложения.	1. Создание простого веб-приложение на python с базой данных для авторизации пользователей	4

4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Темы для самостоятельного изучения	Изучаемые вопросы	Количество часов	Формы самостоятельной работы	Методическое обеспечение	Формы отчетности
Тема 1. История создания. Области применения и перспективы. Инструкции и структура программы.	История создания. Области применения и перспективы. Инструкции и структура программы	1	Подготовка конспекта, подготовка сообщения, подготовка к тесту	Учебно-методическое обеспечение дисциплины	Конспект, сообщение, тест
Тема 2. Установка Python. Доступ к документации. Ввод и вывод данных. Первая	Установка Python. Доступ к документации. Ввод и вывод	1	Подготовка конспекта, подготовка сообщения, подготовка к тесту	Учебно-методическое обеспечение дисциплины	Конспект, сообщение, тест

программа на языке Python.	данных. Первая программа на языке Python				
Тема 3. Переменные: именование переменных, присваивание значения переменным, удаление переменных. Операторы: математические операторы, двоичные операторы, приоритет выполнения операторов.	Переменные: именование переменных, присваивание значения переменным, удаление переменных. Операторы: математические операторы, двоичные операторы, приоритет выполнения операторов	2	Подготовка конспекта, подготовка сообщения, подготовка к тесту	Учебно-методическое обеспечение дисциплины	Конспект, сообщение, тест
Тема 4. Типы данных: числовые типы, строки, списки, кортежи, множества, диапазоны, словари. Операторы для работы с последовательностями и отображениями.	Типы данных: числовые типы, строки, списки, кортежи, множества, диапазоны, словари. Операторы для работы с последовательностями и отображениями.	2	Подготовка конспекта, подготовка сообщения, подготовка к тесту	Учебно-методическое обеспечение дисциплины	Конспект, сообщение, тест
Тема 5. Простые и составные инструкции в	Простые и составные инструкции	2	Подготовка конспекта, подготовка	Учебно-методическое обеспечение	Конспект, сообщение, тест

Python. Условные операторы и циклы.	и в Python. Условные операторы и циклы.		сообщения, подготовка к тесту	дисциплины	
Тема 6. Встроенные и пользовательские функции. Итераторы и функции-генераторы. Декораторы функций.	Встроенные и пользовательские функции. Итераторы и функции-генераторы. Декораторы функций.	2	Подготовка конспекта, подготовка сообщения, подготовка к тесту	Учебно-методическое обеспечение дисциплины	Конспект, сообщение, тест
Тема 7. Объектно-ориентированное программирование. Основы программирования классов. Создание экземпляров класса. Конструктор и деструктор класса. Основы программирования классов. Создание экземпляров класса. Конструктор и деструктор класса. Перегрузка операторов, Наследование. Композиция.	Объектно-ориентированное программирование. Основы программирования классов. Создание экземпляров класса. Конструктор и деструктор класса. Основы программирования классов. Создание экземпляров класса. Конструктор и деструктор класса. Перегрузка операторов, Наследование. Композиция	2	Подготовка конспекта, подготовка сообщения, подготовка к тесту	Учебно-методическое обеспечение дисциплины	Конспект, сообщение, тест
Тема 8. Графический интерфейс	Графический интерфейс	2	Подготовка конспекта, подготовка	Учебно-методическое обеспечение	Конспект, сообщение, тест

пользователя (GUI). Событийно-ориентированное программирование. Инструменты для создания графического интерфейса пользователя: модуль стандартной библиотеки tkinter, фреймворк PyQt.	пользователя (GUI). Событийно-ориентированное программирование. Инструменты для создания графического интерфейса пользователя: модуль стандартной библиотеки tkinter, фреймворк PyQt		сообщения, подготовка к тесту	дисциплины	
Тема 9. Web-фреймворк Flask. Разработка web-приложения.	Web-фреймворк Flask. Разработка web-приложения	2	Подготовка конспекта, подготовка сообщения, подготовка к тесту	Учебно-методическое обеспечение дисциплины	Конспект, сообщение, тест
Итого		16			

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

5.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Код и наименование компетенции	Этапы формирования компетенции	Формы учебной работы по формированию компетенций в процессе освоения образовательной программы
СПК-2. Способен организовывать содержательную практическую деятельность обучающихся с наукоёмкой межпредметной и метапредметной составляющей на основе применения личностно-ориентированного подхода, направленную на конструирование и программирование робототехнических комплектов.	Когнитивный	1. Работа на учебных занятиях 2. Самостоятельная работа
	Операционный	1. Работа на учебных занятиях 2. Самостоятельная работа
	Деятельностный	1. Работа на учебных занятиях 2. Самостоятельная работа

5.2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

СПК-2. Способен организовывать содержательную практическую деятельность обучающихся с наукоёмкой межпредметной и метапредметной составляющей на основе применения личностно-ориентированного подхода, направленную на конструирование и программирование робототехнических комплектов.

Оцениваемые компетенции	Этапы формирования компетенции	Уровни освоения составляющей компетенции	Описание показателей	Критерии оценивания	Шкала оценивания
СПК-2.	Когнитивный	пороговый	Способен организовывать содержательную практическую деятельность обучающихся	Наличие знаний по конструированию и программированию робототехнических комплектов	41-60

		продвинутый	с наукоемкой межпредметной и метапредметной составляющей на основе применения личностно-ориентированного подхода, направленную на конструирование и программирование робототехнических комплектов.	Наличие фундаментальных знаний по конструированию и программированию робототехнических комплектов	81 - 100
	Операционный	пороговый	Способен организовывать содержательную практическую деятельность обучающихся с наукоемкой межпредметной и метапредметной составляющей на основе применения личностно-ориентированного подхода, направленную на конструирование и программирование робототехнических комплектов.	Владение первичными умениями решения задач по конструированию и программированию робототехнических комплектов.	41-60
		продвинутый	составляющей на основе применения личностно-ориентированного подхода, направленную на конструирование и программирование робототехнических комплектов.	Владение умениями создавать задачи по конструированию и программированию робототехнических комплектов. Практическая подготовка	81 - 100

	Деятельност ный	пороговый	Способен организовывать содержательную практическую деятельность обучающихся с наукоемкой межпредметной и метапредметной составляющей на основе применения личностно-ориентированного подхода, направленную на конструирование и программирование робототехнических комплектов.	Способность создавать простые задачи по конструированию и программированию робототехнических комплектов. Практическая подготовка.	41-60
		Продвину тый	Способен организовывать содержательную практическую деятельность обучающихся с наукоемкой межпредметной и метапредметной составляющей на основе применения личностно-ориентированного подхода, направленную на конструирование и программирование робототехнических комплектов.	Способность создавать комплексные задачи по конструированию и программированию робототехнических комплектов. Практическая подготовка.	81 - 100

Шкала оценивания конспектов

Конспекты оцениваются по шкале от 0 до 1 балла.

Максимальное количество баллов – 9 (9 конспектов по 1 баллу)

Показатель	Балл
Выполнено	1 балл
Не выполнено	0 баллов

Шкала оценивания сообщения

Критерии оценивания	Баллы
если представленное сообщение свидетельствует о проведенном самостоятельном исследовании с привлечением различных источников информации; логично, связно и полно раскрывается тема; заключение содержит логично вытекающие из содержания выводы.	25-32 баллов

если представленное сообщение свидетельствует о проведенном самостоятельном исследовании с привлечением двух-трех источников информации; логично, связно и полно раскрывается тема; заключение содержит логично вытекающие из содержания выводы.	13-24 балла
если представленное сообщение свидетельствует о проведенном исследовании с привлечением одного источника информации; тема раскрыта не полностью; отсутствуют выводы.	1-12 баллов
если сообщение отсутствует	0 баллов

Шкала оценивания теста

Написание теста оценивается по шкале от 1 до 34 баллов. Освоение компетенций зависит от результата написания теста:

Критерии оценивания	Баллы
компетенции считаются освоенными на высоком уровне (оценка отлично)	23-34 баллов (80-100% правильных ответов)
компетенции считаются освоенными на базовом уровне (оценка хорошо);	15-19 баллов (70-75 % правильных ответов)
компетенции считаются освоенными на удовлетворительном уровне (оценка удовлетворительно);	7-11 - баллов (50-65 % правильных ответов)
компетенции считаются не освоенными (оценка неудовлетворительно).	1-3 баллов (менее 50 % правильных ответов)

Шкала оценивания практической подготовки

Критерии оценивания	Баллы
Высокая активность на практической подготовке, выполнены все задания, предусмотренные практической подготовкой	3-5 баллов
Средняя активность на практической подготовке, выполнены от 1 до 5 заданий, предусмотренных практической подготовкой	1-2 баллов
Низкая активность на практической подготовке, не выполнены задания, предусмотренные практической подготовкой	0 баллов

5.3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Примерные темы для конспектирования

1. История создания языка программирования Python.

2. Области применения и перспективы языка программирования Python.
3. Инструкции и структура программы на языке программирования Python.
4. Способы доступа к документации языка Python.
5. Переменные: именование переменных, присваивание значения переменным, удаление переменных.
6. Типы данных: числовые типы, строки, списки, кортежи, множества, диапазоны, словари.
7. Операторы для работы с последовательностями и отображениями.
8. Встроенные и пользовательские функции.
9. Итераторы и функции-генераторы. Декораторы функций.
10. Объектно-ориентированное программирование. Основы программирования классов.
11. Конструктор и деструктор класса. Основы программирования классов.
12. Перегрузка операторов, Наследование. Композиция.
13. Инструменты для создания графического интерфейса пользователя: модуль стандартной библиотеки tkinter, фреймворк PyQt.
14. Web-фреймворк Flask. Разработка web-приложения.
15. Операторы: математические операторы, двоичные операторы, приоритет выполнения операторов.

Примерные темы сообщений

1. Введение в программирование на языке Python: основные принципы и понятия.
2. История развития языка программирования Python: от создания до современных технологий.
3. Основные особенности и преимущества языка Python по сравнению с другими языками программирования.
4. Синтаксис и основные элементы языка Python.
5. Работа с переменными и типами данных в Python.
6. Как выполнять ввод и вывод данных в Python.
7. Управляющие конструкции в Python: условные операторы и циклы.
8. Функции и процедуры в Python: объявление, определение и вызов.
9. Объектно-ориентированное программирование в Python: классы, объекты, наследование и полиморфизм.
10. Обработка исключений в Python: try-except блоки, обработка ошибок.
11. Модули и пакеты в Python: создание и использование.
12. Работа с файлами и потоками ввода-вывода в Python.
13. Работа с базами данных в Python: подключение, запросы и обработка результатов.
14. Работа с веб-сервисами и API при помощи Python.
15. Разработка графических интерфейсов с использованием библиотек в Python, например Tkinter или PyQt.
16. Работа с веб-скрапингом и парсингом данных в Python.
17. Создание и работа с виртуальными окружениями в Python.
18. Асинхронное программирование в Python: использование асинхронных функций и корутин.
19. Применение машинного обучения и искусственного интеллекта с помощью библиотек, таких как TensorFlow или scikit-learn, в Python.
20. Разработка веб-приложений и RESTful API на фреймворке Django в Python.

Примерный тест

1. Каким образом объявляется переменная в Python?
 - a) var
 - b) int

- c) `x = 5`
- d) `declare x = 5`

2. Какой оператор используется для выполнения условных операций в Python?

- a) `for`
- b) `while`
- c) `if`
- d) `switch`

3. Какой цикл используется для выполнения повторяющихся операций определенное количество раз?

- a) `for`
- b) `while`
- c) `if`
- d) `switch`

4. Как объявить функцию в Python?

- a) `func my_function():`
- b) `function my_function():`
- c) `def my_function():`
- d) `void my_function():`

5. Какой оператор используется для ввода данных пользователем в Python?

- a) `print`
- b) `input`
- c) `read`
- d) `scan`

6. Каким образом можно узнать длину строки в Python?

- a) `length()`
- b) `count()`
- c) `size()`
- d) `len()`

7. Какие типы данных поддерживает Python?

- a) `int`, `float`, `string`
- b) `boolen`, `list`, `tuple`
- c) `dict`, `set`, `complex`
- d) Все вышеперечисленное

8. Какой оператор используется для объединения двух или более списков в Python?

- a) `concat()`
- b) `merge()`
- c) `join()`
- d) `+`

9. Какой оператор используется для добавления элемента в список в Python?

- a) `append()`
- b) `add()`
- c) `insert()`
- d) `push()`

10. Какая функция позволяет получить наибольшее значение из списка чисел в Python?
- a) min()
 - b) max()
 - c) largest()
 - d) biggest()
11. Какой оператор используется для удаления элемента из списка в Python?
- a) remove()
 - b) erase()
 - c) delete()
 - d) pop()
12. Как объявить пустой словарь в Python?
- a) dict()
 - b) {}
 - c) new dict()
 - d) empty dict()
13. Каким образом можно объединить два словаря в Python?
- a) merge()
 - b) join()
 - c) concat()
 - d) update()
14. Какая структура данных в Python позволяет хранить уникальные элементы без порядка?
- a) list
 - b) tuple
 - c) set
 - d) dictionary
15. Какой оператор используется для проверки на принадлежность элемента к множеству в Python?
- a) in
 - b) contains()
 - c) exists()
 - d) isin()
16. Каким образом можно объявить кортеж в Python?
- a) (1, 2, 3)
 - b) [1, 2, 3]
 - c) {1, 2, 3}
 - d) {1: "one", 2: "two", 3: "three"}
17. Какой оператор используется для объединения строк в Python?
- a) concat()
 - b) merge()
 - c) join()
 - d) +
18. Какой символ используется в Python для обозначения комментария?
- a) //
 - b) --

- c) /*
- d) #

19. Какой модуль позволяет работать с датой и временем в Python?

- a) datetime
- b) time
- c) date
- d) chrono

20. Какой модуль в Python позволяет работать с регулярными выражениями?

- a) regex
- b) re
- c) regexpr
- d) regexru

Задание на практическую подготовку

1. Изучить история создания языка программирования Python.
2. Изучить области применения и перспективы языка программирования Python.
3. Изучить инструкции и структура программы на языке программирования Python
4. Изучить этапы установки Python.
5. Изучить способы доступа к документации языка Python
6. Напишите программу, которая выводит на экран текст «Здравствуй, мир!» (без кавычек)..
7. В популярном сериале «Остаться в живых» использовалась последовательность чисел 4 8 15 16 23 42, которая принесла героям удачу и помогла сорвать джекпот в лотерее. Напишите программу, которая выводит данную последовательность чисел с одним пробелом между ними.
8. Измените предыдущую программу так, чтобы каждое число последовательности 4 8 15 16 23 42 печаталось на отдельной строке
9. Напишите программу вывода на экран трех последовательно идущих чисел, каждое на отдельной строке. Первое число вводит пользователь, остальные числа вычисляются в программе.
10. Напишите программу, которая считывает три целых числа и выводит на экран их сумму. Каждое число записано в отдельной строке.
11. Напишите программу, вычисляющую объём куба и площадь его полной поверхности по введённому значению длины ребра
12. Напишите программу, которая считывает целое число, после чего на экран выводится следующее и предыдущее целое число с пояснительным текстом
13. Напишите программу, которая находит полное число метров по заданному числу сантиметров
14. Напишите программу для пересчёта величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах
15. Пользователь вводит два числа с клавиатуры. Вывести на экран yes, если они отличаются друг от друга на 135, иначе вывести на экран No
16. Дано число. Если оно больше 100 или меньше -100, то вывести на экран символ —, иначе вывести на экран символ +
17. Пользователь вводит номер месяца (от 1 до 12). Вывести название сезона года на экран (зима, весна, лето, осень);
18. Дано три числа. Найти количество положительных чисел среди них
19. Реализуйте итератор колоды карт (52 штуки) CardDeck. Каждая карта представлена в виде строки типа 2 Пик. При вызове функции next() будет представлена следующая карта. По окончании перебора всех элементов возникнет ошибка StopIteration

20. Создайте функцию `infinite(lst, tries)`, которая будет проходить по элементам списка `lst` (целые числа) заданное количество раз (`tries`) циклически. Один раз - один элемент списка. После вывода последнего значения последовательности процедура начнется с самого начала.

Например, если в списке 2 элемента, а функция получила значение 3, то сначала выведется первый объект, потом последний, а потом опять первый. Результат работы функции представьте в виде строки, состоящей из `tries` количества символов

21. Создайте класс `Soda` (для определения типа газированной воды), принимающий 1 аргумент при инициализации (отвечающий за добавку к выбираемому лимонаду). В этом классе реализуйте метод `show_my_drink()`, выводящий на печать Газировка и {ДОБАВКА} в случае наличия добавки, а иначе отобразится следующая фраза: Обычная газировка.

22. Николаю требуется проверить, возможно ли из представленных отрезков условной длины сформировать треугольник. Для этого он решил создать класс `TriangleChecker`, принимающий только положительные числа. С помощью метода `is_triangle()` возвращаются следующие значения (в зависимости от ситуации):

- Ура, можно построить треугольник!;
- С отрицательными числами ничего не выйдет!;
- Нужно вводить только числа!;
- Жаль, но из этого треугольник не сделать.

23. Разработка игры «Тетрис» при помощи библиотеки `tkinter` и фреймворка `PyQt`

24. Создание простого веб-приложение на `python` с базой данных для авторизации пользователей

Примерные вопросы к зачету

1. Что такое язык программирования `Python` и для чего он используется?
2. Каким образом можно установить `Python` на компьютер?
3. Как объявить переменную в `Python` и какие типы данных поддерживает язык?
4. Каким образом можно выполнить вывод на экран в `Python`?
5. Как преобразовать строку в число в `Python`?
6. Каким образом можно выполнить ввод данных пользователем в `Python`?
7. Каким образом можно выполнить математические операции в `Python`?
8. Каким образом можно проверить условие в `Python`?
9. Каким образом можно выполнить цикл в `Python`?
10. Как создать функцию в `Python` и как ее вызвать?
11. Каким образом можно выполнить обработку ошибок в `Python`?
12. Каким образом можно работать с файлами в `Python`?
13. Каким образом можно импортировать модуль в `Python`?
14. Что такое список (`list`) в `Python` и как его создать?
15. Каким образом можно добавить элемент в список в `Python`?
16. Каким образом можно удалить элемент из списка в `Python`?
17. Что такое кортеж (`tuple`) в `Python` и как его создать?
18. Что такое множество (`set`) в `Python` и как его создать?
19. Что такое словарь (`dictionary`) в `Python` и как его создать?
20. Каким образом можно выполнить сортировку списка (`list`) в `Python`?
21. Каким образом можно выполнить генерацию списка в `Python`?
22. Каким образом можно выполнить разделение строки на элементы списка в `Python`?
23. Каким образом можно объединить два списка (`list`) в `Python`?
24. Что такое индексация и срезы (`slicing`) в `Python`, и как их использовать?
25. Что такое модуль (`module`) в `Python`, и как его создать и использовать?

26. Что такое объектно-ориентированное программирование (ООП) в Python, и как использовать классы и объекты?
27. Что такое наследование в Python, и как его использовать?
28. Что такое полиморфизм в Python, и как его использовать?
29. Что такое исключения (exceptions) в Python, и как их обрабатывать?
30. Что такое рекурсия в Python, и как ее использовать?
31. Что такое декораторы (decorators) в Python, и как их использовать?
32. Что такое lambda-функции (анонимные функции) в Python, и как их использовать?
33. Что такое генератор (generator) в Python, и как его использовать?
34. Что такое веб-скрапинг (web scraping) в Python, и как его выполнить?
35. Что такое регулярные выражения (regular expressions) в Python, и как их использовать?
36. Что такое модуль requests в Python, и как его использовать для работы с HTTP запросами?
37. Что такое модуль pandas в Python, и как его использовать для работы с данными?
38. Что такое модуль os в Python, и как его использовать для работы с операционной системой?
39. Что такое виртуальное окружение (virtual environment) в Python, и как его использовать?
40. Что такое модуль tkinter в Python, и как его использовать для создания графического интерфейса пользователя?

5.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Требования по написанию конспекта

Конспект – это краткая письменная фиксация основных фактических данных, идей, понятий и определений, устно излагаемых преподавателем или представленных в литературном источнике. Такой вид аналитической обработки материала должен отражать логическую связь частей прослушанной или прочитанной информации. Результат конспектирования – хорошо структурированная запись, позволяющая обучающемуся с течением времени без труда и в полном объеме восстановить в памяти нужные сведения

Сообщение

Сообщение – продукт самостоятельной работы студента, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы.

Требования по оформлению сообщения

Последовательность подготовки сообщения:

1. Подберите и изучите литературу по теме.
 2. Составьте план сообщения.
 3. Выделите основные понятия.
 4. Введите в текст дополнительные данные, характеризующие объект изучения.
 5. Оформите текст письменно.
 6. Подготовьте устное выступление с сообщением на учебном занятии
- Само выступление должно состоять из трех частей – вступления (10-15%

общего времени), основной части (60-70%) и заключения (20-25%).

Требования к оформлению текста

Общий объем не должен превышать 5 страниц формата А 4, абзац должен равняться 1,25 см.

Поля страницы: левое - 3 см., правое - 1,0 см., нижнее 2 см., верхнее - 2 см. Текст печатается через 1,5 интервала. Если текст набирается в текстовом редакторе Microsoft Word, рекомендуется использовать шрифты: Times New Roman, размер шрифта - 14 пт.

После заголовка, располагаемого посередине строки, не ставится точка. Не допускается подчеркивание заголовка и переносы в словах заголовка.

Страницы нумеруются в нарастающем порядке. Номера страниц ставятся внизу листа по центру, размер шрифта - 12 пт

Титульный лист включается в общую нумерацию, но номер страницы на нем не проставляется (это не относится к содержанию сообщения).

Требования к тесту

Предлагаемые тестовые задания предназначены для повторения пройденного материала и закрепления знаний, главная цель тестов - систематизировать знания студентов. Во всех тестовых заданиях необходимо выбрать правильный из предлагаемых ответов, завершить определение либо вставить недостающий термин. Текущий контроль знаний в виде тестирования, проводится в рамках практического занятия.

Написание теста оценивается по шкале от 1 до 34 баллов. Освоение компетенций зависит от результата написания теста.

Требования к зачету

Промежуточная аттестация по дисциплине, определяющая степень усвоения знаний, умений и навыков студентов и характеризующая этапы формирования компетенций по учебному материалу дисциплины, проводится в виде зачета.

Требования к зачету: зачет по дисциплине «Программирование на языке Python» проводится в конце 1 семестра. На зачете для демонстрации сформированных знаний, умений, навыков и компетенций студент должен ответить на два вопроса, связанных с изучаемыми в течение семестра темами.

Выбор формы и порядок проведения зачета осуществляется кафедрой профессионального и технологического образования. Оценка знаний студента в процессе зачета осуществляется исходя из следующих критериев:

а) умение сформулировать определения понятий, данных в вопросе, с использованием специальной терминологии, показать связи между понятиями;

б) способность дать развернутый ответ на поставленный вопрос с соблюдением логики изложения материала; проанализировать и сопоставить различные точки зрения на поставленную проблему;

в) умение аргументировать собственную точку зрения, иллюстрировать высказываемые суждения и умозаключения практическими примерами на компьютере;

При оценке студента на зачете преподаватель руководствуется следующими критериями:

Шкала оценивания зачета

Баллы	Критерия оценивания
20-15	при полных, исчерпывающих, аргументированных ответах на все основные и дополнительные зачетные вопросы, отличающихся логической последовательностью и четкостью в выражении мыслей и обоснованностью выводов, демонстрирующих знания источников и литературы, понятийного аппарата и умение ими пользоваться при ответе.
14-8	при полных, исчерпывающих, аргументированных ответах на все основные и дополнительные зачетные вопросы, отличающихся логической последовательностью и четкостью в выражении мыслей и обоснованностью выводов, демонстрирующих знания источников и литературы, понятийного аппарата и умение ими пользоваться при ответе.
7-4	при неполных, ответах на все основные и дополнительные зачетные вопросы, демонстрирующих знания источников и литературы, понятийного аппарата и умение ими пользоваться при ответе.
0-3	Студент слабо разбирается в сути материала, не имеет прочных знаний по материалу; на поставленные вопросы отвечает неправильно, допускает грубые ошибки.

Соотношение вида работ и количества баллов в рамках процедуры оценивания

Вид работы	количество баллов
Конспект	до 9 баллов
Сообщение	до 32 баллов
Тест	до 34 балла
Практическая подготовка	до 5 баллов
Зачет	до 20 баллов

Итоговая шкала оценивания результатов освоения дисциплины

При выставлении итоговой оценки преподавателем учитывается работа студента в течение всего срока освоения дисциплины, а также баллы, полученные на промежуточной аттестации.

Цифровое выражение	Выражение в баллах БРС	Словесное выражение	Описание оценки в требованиях к уровню и объему компетенций
5	81-100	зачтено	Освоен продвинутый уровень всех составляющих компетенций СПК-2
4	61-80	зачтено	Освоен повышенный уровень всех составляющих компетенций СПК-2
3	41-60	зачтено	Освоен базовый уровень всех составляющих компетенций СПК-2
2	до 40	не зачтено	Не освоен базовый уровень всех составляющих компетенций СПК-2

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И РЕСУРСНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Основная литература

1. Сузи, Р. А. Язык программирования Python : учебное пособие / Р. А. Сузи. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 350 с. — ISBN 978-5-4497-0705-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/97589.html>
2. Гребенникова, Н. И. Программирование на языке высокого уровня : лабораторный практикум / Н. И. Гребенникова, М. Ю. Сергеев, Т. И. Сергеева. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2021. — 94 с. — ISBN 978-5-7731-0946-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/111478.html>
3. Дроботун, Н. В. Алгоритмизация и программирование. Язык Python : учебное пособие / Н. В. Дроботун, Е. О. Рудков, Н. А. Баев. — Санкт-Петербург : Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2020. — 119 с. — ISBN 978-5-7937-1829-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102400.html>

6.2. Дополнительная литература

1. Шакирьянов, Э. Д. Компьютерное зрение на Python. Первые шаги / Э. Д. Шакирьянов. — Москва : Лаборатория знаний, 2021. — 161 с. — ISBN 978-5-00101-944-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/103032.html>
2. Амоа, К. А. Разработка программных пакетов на языке Python : учебное пособие / К. А. Амоа, Н. А. Рындин, Ю. С. Скворцов. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. — 61 с. — ISBN 978-5-7731-0887-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/108184.html>
3. Карякин, М. И. Визуализация механических систем, процессов и явлений: проектные задания с использованием Vpython : учебное пособие / М. И. Карякин. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2021. — 244 с. — ISBN 978-5-9275-3827-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/117178.html>
4. Букунов, С. В. Объектно ориентированное программирование на языке Python : учебное пособие / С. В. Букунов, О. В. Букунова. — Санкт-Петербург : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2020. — 119 с. — ISBN 978-5-9227-1128-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/117194.html>
5. Протодяконов А.В. Алгоритмы Data Science и их практическая реализация на Python [Электронный ресурс]: учебное пособие/ Протодяконов А.В., Пылов П.А., Садовников В.Е.— Электрон. текстовые данные.— Москва, Вологда: Инфра-Инженерия, 2022.— 392 с
6. Теория и реализация языков программирования : учебное пособие / В. А. Серебряков, М. П. Галочкин, Д. Р. Гончар, М. Г. Фуругян. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 372 с. — ISBN 978-5-4497-0944-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102068.html>
7. Щерба, А. В. Программирование на Python®: первые шаги / А. В. Щерба. — Москва : Лаборатория знаний, 2022. — 251 с. — ISBN 978-5-93208-578-3. — Текст :

электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/120878.html>

8. Мартин, О. Байесовский анализ на Python / О. Мартин ; перевод А. В. Снастин. — Москва : ДМК Пресс, 2020. — 340 с. — ISBN 978-5-97060-768-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/124544.html>

6.3. Ресурсы информационно-телекоммуникационной сети «Интернет»

1. <http://mon.gov.ru> - Министерство образования и науки РФ;
2. <http://www.ed.gov.ru> - Федеральное агентство по образованию;
3. <http://www.fasi.gov.ru> - Федеральное агентство по науке и образованию;
4. <http://www.edu.ru> - Федеральный портал «Российское образование»;
5. <http://old.obrnadzor.gov.ru> - Федеральная служба по надзору в сфере образования и науки;
6. <http://www.garant.ru> - информационно-правовой портал «Гарант»
7. <http://federalbook.ru/projects/fso/fso.html> - Федеральный справочник «Образование в России»;
8. <http://www.school.edu.ru> - Российский общеобразовательный портал;
9. <http://www.openet.edu.ru> - Российский портал открытого образования;
10. <http://www.ict.edu.ru> - портал по информационно-коммуникационным технологиям в образовании;
11. <http://www.fero.ru> - портал Федерального Интернет-экзамена в сфере профессионального образования.
12. <http://pedagogic.ru> - педагогическая библиотека;
13. <http://www.ug.ru> - «Учительская газета»;
14. <http://1september.ru> - издательский дом «Первое сентября»;
15. <http://www.pedpro.ru> - журнал «Педагогика»;
16. http://www.informika.ru/about/informatization_pub/about/276 - научно-методический журнал «Информатизация образования и науки»;
17. <http://www.vovr.ru> - научно-педагогический журнал Министерства образования и науки РФ «Высшее образование в России»;
18. <http://www.hetoday.org> - журнал «Высшее образование сегодня».
19. http://www.prosvetitelstvo.ru/library/articles/?ELEMENT_ID=933. - Портал «Просветительство»
20. <http://www.znanie.org/> - Общество «Знание» России
21. <http://www.gpntb.ru> - Государственная публичная научно-техническая библиотека.
22. <http://www.rsl.ru> - Российская национальная библиотека.
23. <http://www.gpntb.ru> - Публичная электронная библиотека.

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

1. Методические рекомендации по подготовке к практическим занятиям.
2. Методические рекомендации по организации самостоятельной работы по дисциплинам.

8. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Лицензионное программное обеспечение:

Microsoft Windows

Microsoft Office

Kaspersky Endpoint Security

Информационные справочные системы:

Система ГАРАНТ

Система «КонсультантПлюс»

Профессиональные базы данных

fgosvo.ru – Портал Федеральных государственных образовательных стандартов высшего образования

pravo.gov.ru - Официальный интернет-портал правовой информации

www.edu.ru – Федеральный портал Российское образование

Свободно распространяемое программное обеспечение, в том числе отечественного производства

ОМС Плеер (для воспроизведения Электронных Учебных Модулей)

7-zip

Google Chrome

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение дисциплины включает в себя:

- учебные аудитории для проведения учебных занятий, оснащенные оборудованием и техническими средствами обучения: учебной мебелью, доской, демонстрационным оборудованием, персональными компьютерами, проектором;
- помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде.