

Федеральное государственное бюджетное образовательное учреждение высшего образования «Тамбовский государственный университет имени Г.Р. Державина»
Инженерно-технический институт
Кафедра теоретической и экспериментальной физики

УТВЕРЖДАЮ:
И.о. директора института



Т. В. Пасько
«17» июня 2026 г.

РАБОЧАЯ ПРОГРАММА

по дисциплине Б1.В.ДВ.05.07 Программирование устройств телекоммуникаций на языках высокого уровня

Направление подготовки/специальность: 03.03.02 - Физика

Профиль/направленность/специализация: Физические основы робототехнических систем

Уровень высшего образования: бакалавриат

Квалификация: Бакалавр

год набора: 2026

Автор программы:

Кандидат технических наук, Чистилин Денис Анатольевич

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 03.03.02 - Физика (уровень бакалавриата) (приказ Министерства науки и высшего образования РФ от «07» августа 2020 г. № 891).

Рабочая программа принята на заседании Кафедры теоретической и экспериментальной физики «16» июня 2026 г. Протокол № 11

Рассмотрена и одобрена на заседании Ученого совета Инженерно-технического института, Протокол от «17» июня 2026 г. № 10.

СОДЕРЖАНИЕ

1. Цели и задачи дисциплины.....	4
2. Место дисциплины в структуре ОП бакалавриата.....	6
3. Объем и содержание дисциплины.....	6
4. Контроль знаний обучающихся и типовые оценочные средства.....	20
5. Методические указания для обучающихся по освоению дисциплины (модуля).....	25
6. Учебно-методическое и информационное обеспечение дисциплины.....	27
7. Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы.....	28

1. Цели и задачи дисциплины

1.1 Цель дисциплины – формирование компетенций:

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

1.2 Типы задач профессиональной деятельности, к которым готовятся обучающиеся в рамках освоения дисциплины:

- научно-исследовательский
- педагогический

1.3 Дисциплина ориентирована на подготовку обучающихся к профессиональной деятельности в сферах: 01 Образование и наука (в сферах: реализации образовательных программ среднего общего образования, среднего профессионального образования, высшего образования и дополнительных профессиональных программ; научных исследований и научно-конструкторских разработок), 40 Сквозные виды деятельности в промышленности (в сферах: фундаментальных основ физики живых систем и физико-химической биологии, применения диагностического и лечебного оборудования, участия в инновационных и опытно-конструкторских разработках; эксплуатации электронных приборов и систем различного назначения; мониторинга параметров материалов; мониторинга состояния окружающей среды)

1.4 В результате освоения дисциплины у обучающихся должны быть сформированы:

Обобщенные трудовые функции / трудовые функции / трудовые или профессиональные действия (при наличии профстандарта)	Код и наименование компетенции ФГОС ВО, необходимой для формирования трудового или профессионального действия	Индикаторы достижения компетенций
	УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	Организует процесс изучения и практического применения языков высокого уровня для программирования телекоммуникационных устройств, эффективно распределяя временные ресурсы на этапах отладки, оптимизации и анализа кода

1.5 Согласование междисциплинарных связей дисциплин, обеспечивающих освоение компетенций:

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

№ п/п	Наименование дисциплин, определяющих междисциплинарные связи	Форма обучения				
		Очная (семестр)				
		1	2	3	4	8
1	CAD-системы автоматического проектирования		+			
2	CAE-системы автоматического проектирования			+		
3	CAM-системы автоматического проектирования				+	

4	Алгоритмы машинного обучения			+		
5	Безопасность компьютерных сетей			+		
6	Бэкэнд - программирование			+		
7	Введение в высшую математику	+				
8	Введение в спортивное программирование		+			
9	Высшая математика		+			
10	Вычислительные среды обработки данных		+			
11	Графический дизайн			+		
12	Демонстрационный эксперимент в школе		+			
13	Инструменты визуализации данных				+	
14	Комбинаторный анализ		+			
15	Компьютерная графика				+	
16	Математическое и компьютерное моделирование			+		
17	Математическое моделирование и визуализация научных данных средствами языка Python				+	
18	Методы решения геометрических задач				+	
19	Организация школьного кабинета физики			+		
20	Программирование Java		+			
21	Программирование на PHP				+	
22	Программирование устройств телекоммуникаций на языках низкого уровня			+		
23	Проективная геометрия			+		
24	Решение олимпиадных физических задач				+	
25	Системы многоуровневой защиты информации				+	

26	Спортивное программирование: уровень 1			+		
27	Спортивное программирование: уровень 2				+	
28	Стандарты в области информационной безопасности		+			
29	Физика и химия твёрдого тела		+			
30	Философия					+
31	Цифровая культура	+				
32	Цифровая обработка изображений		+			
33	Экспертные системы		+			

2. Место дисциплины в структуре ОП бакалавриата:

Дисциплина «Программирование устройств телекоммуникаций на языках высокого уровня» относится к части, формируемой участниками образовательных отношений, учебного плана ОП по направлению подготовки 03.03.02 - Физика.

Дисциплина «Программирование устройств телекоммуникаций на языках высокого уровня» изучается в 4 семестре.

3. Объем и содержание дисциплины

3.1. Объем дисциплины: 2 з.е.

Очная: 2 з.е.

Вид учебной работы	Очная (всего часов)
Общая трудоёмкость дисциплины	144
Контактная работа	64
Лабораторные (Лаб. раб.)	64
Самостоятельная работа (СР)	80
Зачет	-

3.2. Содержание курса:

№ темы	Название раздела/темы	Вид учебной работы, час.		Формы текущего контроля
		Лаб · раб.	СР	
		О	О	
4 семестр				
1	Переменные. Типы данных. Преобразование типов данных.	2	4	Отчет по лабораторноиу занятию

2	Условные операторы и циклы.	4	4	Отчет по лабораторному занятию
3	Строки и двоичные данные.	2	4	Отчет по лабораторному занятию
4	Функции и методы для работы со строками.	4	4	Отчет по лабораторному занятию
5	Списки. Операции над списками	4	4	Отчет по лабораторному занятию
6	Кортежи, множества и диапазоны	4	4	Отчет по лабораторному занятию
7	Словари. Операции и методы для работы со словарями.	4	4	Отчет по лабораторному занятию
8	Работа с датой и временем.	2	4	Отчет по лабораторному занятию
9	Пользовательские функции.	4	4	Отчет по лабораторному занятию
10	Работа с файлами и каталогами.	2	4	Отчет по лабораторному занятию

Тема 1. Переменные. Типы данных. Преобразование типов данных.

Лабораторные работы.

Лабораторная работа №1

Определение переменной. Имена переменной и зарезервированные слова. Типы переменных. Область видимости: локальные и глобальные переменные. Для чего нужны? Определение: переменная, объект, ссылки. Присвоение переменной. Примеры. Определение идентификатора объекта, описание его использования и примеры. Виды работ с переменной, их описание и примеры.

Задачи

1. Реализовать приложение для вычисления периметра и площади прямоугольника. Приложение запрашивает у пользователя два целых числа — длины сторон прямоугольника. Пример работы приложения:

Input a: 5

Input b: 4

P = 18

S = 20

2. Напишите программу, которая считывает значения двух переменных a и b, затем меняет их значения местами (то есть в переменной a должно быть записано то, что раньше хранилось в b, а в переменной b записано то, что раньше хранилось в a). Затем выведите значения переменных.

3. Напишите программу, которая приветствует пользователя, выводя слово Hello, введенное имя и знаки препинания по образцу. Программа должна считывать в строковую переменную значение и писать соответствующее приветствие.

Пример:

Your name: Harry

Hello, Harry!

4. Даны два целых числа: а и b. Найти их среднее арифметическое по формуле:

$$a+b/2$$

5. Электронные часы показывают время в формате часы(h): минуты(m): секунды(s), найти количество секунд прошедшее с начала суток.

Дополнительные задачи

1. Напишите программу, которая по данному числу n от 1 до 9 выводит на экран n пингвинов. Изображение одного пингвина имеет размер 5×9 символов, между двумя соседними пингвинами также имеется пустой (из пробелов) столбец. Для упрощения рисования скопируйте пингвина из примера в среду разработки.

Пример:

Penguins: 3

```

  ~ ~ ~
(o o) (o o) (o o)
 / V \ / V \ / V \
/( _ )\/( _ )\/( _ )\
 ^ ^ ^ ^ ^ ^ ^ ^ ^

```

2. Непокойно сейчас на стапелях шестого дока межгалактического порта планеты Торна. Всего через месяц закончится реконструкция малого броненосущего корвета «Эния». И снова этому боевому кораблю и его доблестной команде предстоят тяжелые бои за контроль над плутониевыми рудниками Сибелиуса. Работа не прекращается ни на секунду, лазерные сварочные аппараты работают круглые сутки. От непрерывной работы плавятся шарниры роботов-ремонтников. Но задержаться нельзя ни на секунду. И вот в этой суматохе обнаруживается, что термозащитные панели корвета вновь требуют срочной обработки сульфидом тория. Известно, что на обработку одного квадратного метра панели требуется 1 нанограмм сульфида. Всего необходимо обработать N прямоугольных панелей размером A на B метров. Вам необходимо как можно скорее подсчитать, сколько всего сульфида необходимо на обработку всех панелей «Энии». И не забудьте, что панели требуют обработки с обеих сторон.

3. Решите задачу 2 для целых чисел без использования дополнительной переменной.

Задания для самостоятельной работы.

1. Дано целое десятичное число. Выведите его последнюю цифру.

2. Дано целое десятичное число. Найдите число десятков в его десятичной записи.

3. Дано трехзначное число. Найдите сумму его цифр.

4. Пирожок в столовой стоит а рублей и b копеек. Определите, сколько рублей и копеек нужно заплатить за n пирожков.

5. В школе решили набрать три новых математических класса. Так как занятия по математике у них проходят в одно и то же время, было решено выделить кабинет для каждого класса и купить в них новые парты. За каждой партой может сидеть не больше двух учеников. Известно количество учащихся в каждом из трёх классов. Сколько всего нужно закупить парт чтобы их хватило на всех учеников? Программа получает на вход три целых десятичных числа: количество учащихся в каждом из трех классов.

6. Доработайте код задачи № 3 таким образом, чтобы он запрашивал время начала занятий (минуты и часы отдельно) и номер урока, а далее также рассчитывал время окончания уроков.

7. Пользователь вводит число и систему счисления этого числа. Программа переводит число в десятичную, двоичную, восьмеричную и шестнадцатеричную системы счисления с использованием стандартных функций.

Тема 2. Условные операторы и циклы.

Лабораторные работы.

Лабораторная работа №2

Определение условных операторов, их виды. Зачем нужны? Как работают? Оператор if: описание, синтаксис, как работает, пример. Оператор else: описание, синтаксис, как работает, пример. Оператор elif: описание, синтаксис, как работает, пример. Отступ. Определение. Зачем нужны? Примеры их использования. Описание цикла while, его синтаксис, примеры. Оператор break: описание, пример. Оператор continue: описание, пример. Определение. Вложенные циклы for и while: синтаксис, примеры. Описание цикла for, его синтаксис, основные части конструкции, примеры. Оператор break: описание, пример. Оператор continue: описание, пример. Функция range: описание, синтаксис, пример. Примеры использования цикла for и их описание.

Задачи

1. [for] Пользователь вводит числа **a** и **b**. Вывести все целые числа, расположенные между ними.
2. [for] Доработать предыдущую задачу так, чтобы выводились только числа, делящиеся на 5 без остатка.
3. [for] Пользователь вводит число. Вывести таблицу умножения на это число
4. [for] Известны оценки абитуриента на четырех экзаменах. Определить сумму набранных им баллов. Оценки вводятся в цикле.
5. [while] В бесконечном цикле приложение запрашивает у пользователя числа. Ввод завершается, как только пользователь вводит слово 'end'. После завершения ввода приложение выводит сумму чисел.
6. [while] Дано натуральное число. Определить сумму цифр в нем.
7. Пользователь вводит число **n** и цифру **a**. Определить, сколько раз цифра встречается в числе. (не использовать метод count)

Задания для самостоятельной работы.

1. Напишите программу, которая получает с клавиатуры два целых числа и вычисляет их произведение, используя только операции сложения.
2. Напишите программу, которая получает с клавиатуры натуральное число и вычисляет целый квадратный корень из него – наибольшее число, квадрат которого не больше данного числа.
3. Натуральное число называется числом Армстронга, если сумма цифр числа, возведенных в N-ную степень (где N – количество цифр в числе) равна самому числу.
4. Вывести на экран циклом пять строк из нулей, причем каждая строка должна быть пронумерована;
5. Найти сумму ряда чисел от 1 до 100. Полученный результат вывести на экран;
6. Даны три числа. Вывести на экран «yes», если среди них есть одинаковые, иначе вывести "ERROR";
7. Даны три числа. Вывести на экран «yes», если можно взять какие-то два из них и в сумме получить третье;
8. Дано три числа. Найти количество положительных чисел среди них;
9. Вывести на экран все чётные значения в диапазоне от 1 до 497;
10. Написать программу, которая будет складывать, вычитать, умножать или делить два числа. Числа и знак операции вводятся пользователем. После выполнения вычисления

Тема 3. Строки и двоичные данные.

Лабораторные работы.

Лабораторная работа №3.

Тип данных bytes. Байтовые строки. Синтаксис байтовых литералов. Встроенные классы bytes. Методы bytes. Примеры использования. Тип данных bytearray. Встроенные классы bytearray. Объекты bytearray. Методы bytearray. Литеральный формат. Использование simple crypt. Использование библиотеки cryptocode. Использование пакета cryptography. Использование алгоритма RSA.

Задачи

1. Пользователь вводит фамилию, имя и отчество. Приложение должно вывести фамилию и инициалы. Пример:

Фамилия: Ершов

Имя: Андрей

Отчество: Петрович

Ершов А. П.

2. Доработать приложение из предыдущей задачи так, чтобы программа исправляла регистр символов. Пример:

Фамилия: ерШоВ

Имя: андрей

Отчество: петрович

Ершов А. П.

3. Пользователь вводит слово. Подсчитать количество символов 'а' в нем. Пример:

word: abracadabra

5

4. Пользователь вводит строку. Нужно удалить из нее первое слово. Разделителем слов считать пробел. Пример:

> Hello, World!

World

Пользователь вводит строку. Нужно удалить из нее последнее слово.

5. Пользователь вводит строку, содержащую два слова. Приложение выводит их в обратном порядке.

Пример:

Harry Potter

Potter Harry

6. Пользователь вводит строку и два слова. Приложение заменяет все вхождения первого слова на второе. Пример:

> To be or not to be.

Find: be

Replace: eat

To eat or not to eat.

7. Приложение принимает на вход строку и заменяет все вхождения буквы «ё» на букву «е».

8. Написать приложение, выполняющее транслитерацию введенного слова. Пример:

> Иван

Ivan

9. Пользователь вводит адрес электронной почты, нужно вывести отдельными строками почтовый логин и домен:

> vasya@mail.ru

Login: vasya

Domain: mail.ru

Задания для самостоятельной работы.

1. Строки в python обозначаются кавычками. Приведите все способы.
2. Какие типы данных можно преобразовать в строку?
3. Опишите синтаксис срезов строк при помощи квадратных скобок.
4. Как применяют операции сложения и умножения к строкам?
5. Что такое двоичные данные?

Тема 4. Функции и методы для работы со строками.

Лабораторные работы.

Лабораторная работа №4

Определение что такое строка. Литералы строк. Кодировка строк. Максимальная длина строки. Перенос строк. Сравнение строк. Пустая строка. Удаление строки. Сообщение по индексу. Функции для работы со строками. Оператор %. Метод str. format (). F-строки. Библиотека Template Strings. Функции для работы со строками. Методы для работы со строками. Примеры использования методов для работы со строками. Преобразование строки в другой тип. String to Int. String to Bytes. String to Float. Примеры преобразования строки в другой тип.

Задачи

1. Реализовать функцию is_sorted. Функция принимает на вход список и возвращает True, если элементы в нем упорядочены по возрастанию.

```
assert is_sorted([]) == True
assert is_sorted([42]) == True
assert is_sorted([3, 14, 15, 92]) == True
assert is_sorted([1, 1, 1]) == True
assert is_sorted([1, 1, 2, 2, 3, 3, 3]) == True
assert is_sorted([2, 1]) == False
assert is_sorted([1, 2, 1]) == False
```

2. Реализовать функцию find_longest. Функция принимает на вход список строк и возвращает строку с максимальной длиной. Если таких строк несколько, то возвращает первую из них.

```
assert find_longest([]) == None
assert find_longest([""]) == ""
assert find_longest(["42"]) == "42"
assert find_longest(["uno", "dos", "tre"]) == "uno"
assert find_longest(["a", "bb", "ccc"]) == "ccc"
assert find_longest(["a", "ccc", "bb"]) == "ccc"
```

3. Реализовать функцию min_max. Функция принимает на вход список чисел и возвращает пару: минимальное и максимальное число. Алгоритм должен выполнять не более одного прохода по списку.

```
assert min_max([]) == (None, None)
assert min_max([1]) == (1, 1)
assert min_max([1, 1]) == (1, 1)
assert min_max([1, 2]) == (1, 2)
assert min_max([2, 1]) == (1, 2)
assert min_max([14, 3, 14, 92, 15]) == (3, 92)
```

4. Реализовать функцию zip_longest_sum. Функция принимает на вход два списка и возвращает новый список, полученный сложением соответствующих элементов входных списков.

```
assert zip_sum_longest([], []) == []
assert zip_sum_longest([1], []) == [1]
assert zip_sum_longest([], [1]) == [1]
assert zip_sum_longest([1], [2]) == [3]
assert zip_sum_longest([1], [2, 1]) == [3, 1]
```

```
5. assert zip_sum_longest([2, 1], [1]) == [3, 1]
```

6. Реализовать функцию is_prime. Функция принимает на вход целое число и возвращает True, если число простое. Простым называется число, которое делится без остатка только на 1 и на себя.

```
assert is_prime(5) == True
assert is_prime(113) == True
assert is_prime(12) == False
```

7. Реализовать функцию join. Функция принимает разделитель и список строк, возвращает строку, в которой элементы списка чередуются с разделителем.

```
assert join(' ', []) == ""
assert join(' ', ['one']) == 'one'
```

`assert join(' ', ['one', 'two', 'three']) == 'one, two, three'`

Задания для самостоятельной работы.

1. Даны четыре действительных числа: x_1, y_1, x_2, y_2 . Напишите функцию `distance(x1, y1, x2, y2)`, вычисляющая расстояние между точкой (x_1, y_1) и (x_2, y_2) . Считайте четыре действительных числа и выведите результат работы этой функции.
2. Дано действительное положительное число a и целое число n . Вычислите a^n .
3. Напишите функцию `search_substr(subst, st)`, которая принимает 2 строки и определяет, имеется ли подстрока `subst` в строке `st`. В случае нахождения подстроки, возвращается фраза «Есть контакт!», а иначе «Мимо!». Должно быть найдено совпадение независимо от регистра обеих строк.

Тема 5. Списки. Операции над списками

Лабораторные работы.

Лабораторная работа №5

Определение что такое список. Объявление списка. Обращение к элементу списка. Добавление в список. Изменение элемента списка. Удаление элемента из списка. Проверка наличия элемента. Объединение списков. Копирование списка. Цикл по списку. Примеры использования. Методы работы со списками. Копирование списков. Сложение списков. Сортировка списков. Вложенные списки. Срезы. Генераторы списков. Примеры использования.

Задачи

1. Пользователь вводит числа a и b . Создать список, содержащий квадраты целых чисел расположенных между ними.
2. Пользователь вводит числа до тех пор, пока не введет слово `end`. Поместите введенные числа в список. Выведите на экран только нечетные элементы списка.
3. Пользователь вводит числа до тех пор, пока не введет слово `'end'`. Поместите введенные числа в список. Подсчитайте количество четных и нечетных элементов в списке.
4. Дан список чисел. Выведите все элементы списка, которые больше предыдущего элемента.
5. Дан список чисел. Поменяйте местами минимальный и максимальный элемент этого списка.
6. Дан список чисел. Определите, сколько в нем различных элементов.
7. Петя перешёл в другую школу. На уроке физкультуры ему понадобилось определить своё место в строю. Помогите ему это сделать. Программа получает на вход последовательность чисел, означающих рост каждого человека в строю. После этого вводится число X – рост Пети. Все числа во входных данных натуральные и не превышают 200. Выведите номер, под которым Петя должен встать в строй. Если в строю есть люди с одинаковым ростом, таким же, как у Пети, то он должен встать после них.
8. Дан список чисел. Циклически сдвиньте элементы списка вправо ($A[0]$ переходит на место $A[1]$, $A[1]$ на место $A[2]$, ..., последний элемент переходит на место $A[0]$).

Задания для самостоятельной работы.

1) Как создать список?

1) Все варианты верны

2) `L = list(1, 2, 3)`

3) `l = [1, 2, 3]`

4) `l = list[1, 2, 3]`

Ответ:

2) Что выведет этот код:

```
a = [ 1, 342, 223, 'Африка', 'Очки']
```

```
print(a[-3])
```

1) 223

2) 'Африка'

3) 342

4) Ошибку

Ответ:

3) Что выведет этот код:

```
sample = [10, 20, 30]
sample.append(60)
sample.insert(3, 40)
print(sample)
1)[10, 20, 30, 40]
2)[10, 20, 30, 40, 60]
3)[10, 20, 30, 60, 40]
4)[60, 10, 20, 30, 40]
```

Ответ:

4) Что из перечисленного правда?

- 1) Элементы списка не могут повторяться
- 2) Все элементы списка должны быть одного типа
- 3) Мы можем вставить элемент на любую позицию в списке
- 4) Список не может содержать вложенных списков

Ответ:

5) Как получить ['bar', 'baz'] из списка

```
a = ['foo', 'bar', 'baz', 'qux', 'quux']
```

?

- 1) print(a[2:4])
- 2) print(a[1], a[2])
- 3) print(a[1:-2])
- 4) print(a[-4:-3])
- 5) print(a[2:3])

Ответ:

Тема 6. Кортежи, множества и диапазоны

Лабораторные работы.

Лабораторная работа №6

Определение кортежей. Использование кортежей вместо списков. Работа с кортежами. Методы и особые операции. Преобразование кортежей. Именованые кортежи. Определение множеств. Назначение множеств. Особенности set. Работа с set-ами. Операции на множествах. Свойства методов и операторов. Преобразование множеств. Определение диапазона. Range start stop step. Функции range. Использование параметров. Использование отрицательных значений. Генерация диапазона. Переборка списка с помощью range. Использование индекса. Объединение выводов.

Задачи

1. Реализовать игру «Города».

Шаг 1. Версия для двух игроков за одним компьютером. Названные города не должны повторяться.

Шаг 2. Добавить проверку на существование города.

Шаг 3. Реализовать режим игры против компьютера.

Модули с подготовленными списками городов:

- cities_cpu_win.py
- cities_loop.py
- cities_player_win.py
- cities.py

2. Реализовать игру «Анаграммы». Приложение выводит слово, буквы в котором перемешаны.

Игрок должен назвать исходное слово.

3. Пользователь вводит три числа: n, min_value, max_value. Приложение создает список из n элементов и заполняет его случайными числами из [min_value, max_value].

4. По аналогии с задачей 1 создать матрицу n на m и заполнить ее случайными числами из $[\text{min_value}, \text{max_value})$.

5. Цвета кодируются тройкой (red, green, blue). Каждый компонент может принимать значения от 0 до 255. Напишите приложение, генерирующее случайный цвет. Для удобства его значение можно вывести в шестнадцатеричном формате (например, 345EBF). Для просмотра полученного цвета можно использовать сервис <https://colorscheme.ru/>.

6. Пользователь вводит число n . Приложение создает и выводит матрицу, заполненную «змейкой».

Пример:

```
n: 5
1  2  3  4  5
10 9  8  7  6
11 12 13 14 15
20 19 18 17 16
21 22 23 24 25
```

7. Реализовать приложение-лотерею. Пусть лотерейный билет хранится как список списков:

```
ticket = [
    [ 1, 2, 3, 4, 5],
    [ 6, 7, 8, 9, 10],
    [11, 12, 13, 14, 15],
    [16, 17, 18, 19, 20],
    [21, 22, 23, 24, 25],
]
```

Пользователь должен выбрать пять чисел: по одному числу из каждой строки. Числа могут быть выбраны в любом порядке. Приложение делает аналогичный выбор случайным образом, а затем выводит статистику. Пример:

```
row 0 3
row 1 7
row 2 14
row 3 19
row 4 21
User Random
3    1
7    9
14   13
19   19 +
21   23
```

Задания для самостоятельной работы.

1) Что такое множество в Python?

1) Это любая коллекция элементов

2) Это список, содержащий в себе только функции

3) Это контейнер, значения в котором не повторяются

4) Это список, содержащий вложенные списки в себе

Ответ:

2) Каким образом правильно объявляется множество?

1) $a = \{\}$

2) $a = []$

3) $a = \text{set}()$

4) $a = \text{set}$

Ответ:

3) Чем отличаются методы `remove()` и `discard()`, применяемые к множеству?

1) Ничем

2) `remove()` удаляет элемент если он есть, но бросает ошибку если элемента нет. `discard()` просто удаляет элемент если он есть

3) `discard()` удаляет элемент если он есть, но бросает ошибку если элемента нет. `remove()` просто удаляет элемент если он есть

4) Метода `discard()` для множеств не существует

Ответ:

4) Что такое `frozenset`?

1) Множество, которое нельзя изменять

2) Множество, которое хранит в себе только неизменяемые объекты

3) Множество, которое используется для хранения констант

4) Выдумка нашей редакции

Ответ:

5) Что это за метод такой, `set.difference(another_set)`

1) Возвращает `True`, если два множества одинаковые, `False` если хотя бы один элемент не совпадает

2) Возвращает `True`, если два множества разные, `False` если хотя бы один элемент совпадает

3) Возвращает множество из элементов, которые встречаются только в множестве `set`

4) Возвращает множество из элементов, которые встречаются только в множестве `another_set`

Тема 7. Словари. Операции и методы для работы со словарями.

Лабораторные работы.

Лабораторная работа №7

Определение. Создание словаря. Доступ к элементам словаря. Добавление элементов в словарь. Обновление элементов. Удаление элементов. Примеры их использования. Итерационный словарь. Встроенные методы словаря: описание, синтаксис, примеры

Задачи

1. Написать программу, которая на вход получает файл и выводит на экран уникальные слова. Использовать тип `set`.

2. Написать программу, которая на вход получает файл и выводит на экран какое слово сколько раз встречалось. Использовать тип `dict`.

3. Создайте функцию `compute_bill`, считающую итоговую сумму товаров в чеке. Функция должна принимать 1 параметр - словарь, в котором указано количество единиц товара. Цены хранятся в словаре:

```
prices = {
    "banana": 4,
    "apple": 2,
    "orange": 1.5,
    "pear": 3
}
```

4. Создайте программу, которая будет выводить все возможные комбинации при броске 2 игральных костей и сумму их значений. Пример вывода:

2: [(1, 1)]

3: [(1, 2), (2, 1)]

...

Задания для самостоятельной работы.

1) Что означает ошибка `TypeError: unhashable type`?

1) Неверно задано значение

2) Тип данных нельзя использовать в роли ключа

3) Слишком большое значение

4) Ошибка синтаксиса

Ответ:

2) Какие типы данных нельзя использовать в роли ключа?

1) Список, словарь

2) Словарь, кортеж

3) Кортеж, число

4) Число, булево значение

Ответ:

3) Что выдаст этот код?

```
Another_dict = {'a': {'a': ['a']}}
```

```
Print(another_dict.pop('a') == another_dict.clear())
```

1) True

2) False

3) Ошибка

Ответ:

4) Каков будет результат кода?

```
Dict_1 = {'a': 10, 'b': 20}
```

```
Dict_2 = {'b': 20, 'c': 30}
```

```
Dict_1.update(dict_2)
```

```
Print(dict_1)
```

1) {'a': 10}

2) {'a': 10, 'b': 20}

3) {'a': 10, 'b': 20, 'c': 30}

4) {'b': 20, 'c': 30}

Ответ:

5) Что выведет этот код?

```
Old_dict = {'a': 10, 'b': 10}
```

```
New_dict = {}
```

```
For i, j in old_dict.items():
```

```
New_dict[j] = i
```

```
Print(new_dict)
```

1) {'a': 10, 'b': 10}

2) {10: 'a', 10: 'b'}

3) {'a': 10}

4) {10: 'b'}

Ответ:

Тема 8. Работа с датой и временем.

Лабораторные работы.

Лабораторная работа №1

Описание. Применение. Классы, представленные модулем datetime: описание, синтаксис, примеры.

Методы класса datetime: описание, синтаксис, примеры. Datetime.datetime и Datetime.timedelta:

Описание. Синтаксис. Параметры. Атрибуты экземпляра класса. Создание объекта: описание,

синтаксис, примеры. Поддерживаемые операции. Описание. Функция модуля времени time:

описание, синтаксис, пример.

Задачи

1) Определите что будет выведено после выполнения данного кода:

```
from datetime import datetime, timedelta
```

```
given_date = datetime(2020, 2, 25)
```

```
print("Given date")
print(given_date)
days_to_subtract = 7
res_date = given_date - timedelta(days=days_to_subtract)
print("New Date")
print(res_date)
```

а) Все даты прошлой недели

б) Все дни прошлой недели

в) Дата неделю назад

г) День недели неделю назад

2) Определите что будет выведено после выполнения данного кода:

```
from datetime import datetime
given_date = datetime(2020, 7, 26)
print(given_date.today().weekday())
print(given_date.strftime('%A'))
```

а) Проверка заданной даты на четность

б) День недели заданного числа

в) Проверка является ли заданная дата выходным

г) Является ли четной неделя, в которой находится заданная дата

3) Определите что будет выведено после выполнения данного кода:

```
from datetime import datetime
given_date = datetime(2020, 2, 25)
string_date = given_date.strftime("%Y-%m-%d %H:%M:%S")
print(string_date)
```

а) Заданная дата в измененном формате (другой внешний вид)

б) Заданная дата в типе str

в) Заданная дата через неделю

г) Данная дата в типе int для проведения вычислений

4) Определите что будет выведено в результате выполнения данного кода:

```
from datetime import datetime
date_1 = datetime(2020, 2, 25).date()
date_2 = datetime(2020, 9, 17).date()
delta = None
```

```
if date_1 > date_2:
```

```
    print("date_1 is greater")
```

```
    delta = date_1 - date_2
```

```
else:
```

```
    print("date_2 is greater")
```

```
    delta = date_2 - date_1
```

```
print("Difference is", delta.days, "days")
```

а) Количество дней между сегодняшним днем и указанными датами

б) Количесво дней между указанными днями

в) Какая дата "старше" (какое число дальше от начала календаря)

г) Какая дата "младше" (какое число ближе к началу календаря)

****За начало календаря принимать Рождество Христово**

5) Определите что будет выведено в результате выполнения данного кода:

```
from datetime import datetime, timedelta
given_date = datetime(2020, 3, 22, 10, 00, 00)
print("Given date")
```

```
print(given_date)
days_to_add = 7
res_date = given_date + timedelta(days=days_to_add, hours=12)
print("New Date")
print(res_date)
```

- а) Указанная дата с добавлением 1 недели и 12 часов
- б) Указанная дата с добавлением 12 часов
- в) Указанная дата с вычетом 1 недели и 12 часов
- г) Указанная дата с вычетом 12 часов и добавлением одной недели

Задания для самостоятельной работы.

1. Как получить текущую дату и время в Python?
2. Как получить текущую дату?
3. Что внутри datetime?
4. Как вывести час, минуту, секунду и микросекунду?

Тема 9. Пользовательские функции.

Лабораторные работы.

Лабораторная работа №9

Определение функций. Синтаксис. Важность функций. Модульность. Абстракция. Повторное использование. Пространство имен. Объявление функций. Вызов функции. Локальная видимость функций. Область объемлющих функций. Позиционные аргументы. Именованные аргументы. Необязательные параметры. Что можно возвращать. Распаковка возвращаемых значений. Пустая функция. Чистые функции. Lambda функции. Аннотация типов. Типы данных. Использование функций. Перевод из одного типа в другой.

Задачи

1. Реализуйте функцию `encode_symbol`. Функция принимает символ и число, на которое его нужно сдвинуть. Функция шифрует только символы латинского алфавита, все прочие оставляет без изменений.
2. Реализуйте функцию `encode_str`. Функция принимает строку и ключ шифрования и возвращает зашифрованную строку.
3. Реализуйте функцию `encode_file(input_file_path, output_file_path, key)`.
4. Реализуйте для приложения графический интерфейс с помощью `easygui`. Для открытия шифруемого файла используйте функцию `easygui.fileopenbox(msg=None, title=None)`. Для сохранения зашифрованного файла используйте `easygui.filesavebox(msg=None, title=None)`. Для ввода ключа — `easygui.integerbox(msg="", title=" ")`.

Задания для самостоятельной работы.

- 1) Как можно вызвать метод `func` у следующего класса (выберите все подходящие варианты):

```
Class myclass:
```

```
Def func(self):
```

```
Print('hello')
```

- 1) `myClass.func()`
- 2) `obj = myClass() obj.func()`
- 3) `obj = myClass() myClass.func(obj)`
- 4) `obj = myClass() obj.func`
- 5) ни один из перечисленных

Ответ:

- 2) Что напечатает следующий код:

```
def func(n):
```

```
    n = n + 1
```

```
print(func(0))
```

- 1)0
- 2)1
- 3)func(0)
- 4)None
- 5)возникнет ошибка

Ответ:

3) Что выведет следующий код:

```
a = 3
a = "foo" if a / 2 == 1 else 2
a = a + a
print (a)
```

- 1)6
- 2)Возникнет ошибка
- 3)2
- 4)4
- 5)foofoo

Ответ:

4) Что необходимо добавить на место пропущенной строки?

```
def find_max(nums):
    max_num = float("-inf")
    for num in nums:
        if num > max_num:
            # пропущенная строка
    return max_num
```

- 1)max_num = num
- 2)num = max_num
- 3)max_num += 1
- 4)max_num += num

Ответ:

Тема 10. Работа с файлами и каталогами.

Лабораторные работы.

Лабораторная работа №10

В Python существует два типа файлов: текстовые, бинарные. Существуют несколько способов составления списка каталогов в Python, позволяющих получить имена всех файлов и папок. Обработка файлов в Python с помощью модуля os включает создание, переименование, перемещение, удаление файлов и папок, а также получение списка всех файлов и каталогов и многое другое.

Задачи

1. Напишите функцию `read_last(lines, file)`, которая будет открывать определенный файл `file` и выводить на печать построчно последние строки в количестве `lines` (на всякий случай проверим, что задано положительное целое число). Протестируем функцию на файле «`article.txt`» со следующим содержимым:

```
Вечерело
Жужжали мухи
Светил фонарик
Кипела вода в чайнике
Венера зажглась на небе
```

Деревья шумели
Тучи разошлись
Листва зеленела

2. Выберите любую папку на своем компьютере, имеющую вложенные директории. Выведите на печать в терминал ее содержимое, как и всех подкаталогов при помощи функции `print_docs(directory)`.

3. Документ «article.txt» содержит следующий текст:

Вечерело
Жужжали мухи
Светил фонарик
Кипела вода в чайнике
Венера зажглась на небе
Деревья шумели
Тучи разошлись
Листва зеленела

Требуется реализовать функцию `longest_words(file)`, которая выводит слово, имеющее максимальную длину (или список слов, если таковых несколько).

4. Требуется создать csv-файл «rows_300.csv» со следующими столбцами:– № - номер по порядку (от 1 до 300);– Секунда – текущая секунда на вашем ПК;– Микросекунда – текущая миллисекунда на часах. На каждой итерации цикла искусственно приостанавливайте скрипт на 0,01 секунды.

5. При помощи библиотеки Pillow в директории circles (создайте ее во время выполнения функции) нарисуйте и сохраните 100 кругов радиусом 300 пикселей случайных цветов в формате jpg на белом фоне (каждый круг - отдельный файл). Для этого напишите функцию `circles_generator(num_of_circles=100)`.

Задания для самостоятельной работы.

1. Разработать приложение, которое записывает в файл все строки, введенные пользователем. Признак конца ввода — пустая строка. После выполнения программы должен появиться файл data.txt, содержащий три строки:

2. Разработать приложение для нумерации строк в файле. Приложение принимает на вход имя файла и выводит его построчно, добавляя к каждой строке ее номер. Если использовать файл, созданный в предыдущей задаче. Введите имя файла: data.txt

3. Разработать приложение для разделения файла на части. Приложение принимает на вход имя файла для разделения и целое число N. На выходе у приложения множество файлов, каждый из которых содержит не более, чем N строк из исходного файла.

4. Разработать приложение для объединения файлов. Приложение принимает на вход имена файлов для объединения (можно использовать файлы, полученные из предыдущего задания) и имя выходного файла.

4. Контроль знаний обучающихся и типовые оценочные средства

4.1. Распределение баллов:

4 семестр

- посещаемость – 10 баллов
- текущий контроль – 60 баллов
- контрольные срезы – 2 среза по 15 баллов каждый

Распределение баллов по заданиям:

№ те мы	Название темы / вид учебной работы	Формы текущего контроля / срезы	Мах. кол-во баллов	Методика проведения занятия и оценки

1.	Переменные. Типы данных. Преобразования типов данных.	Отчет по лаборатор ному занятию	5	баллов – студент выполнил лабораторную работу правильно, расчеты по работе произведены верно, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 3 балла – студент выполнил лабораторную работу с ошибками, расчеты по работе содержат неточности, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 1 балл – студент выполнил лабораторную работу с грубыми ошибками, не смог правильно провести расчеты и ответить на контрольные вопросы. 0 баллов – студент не выполнил лабораторную работу.
2.	Условные операторы и циклы.	Отчет по лаборато рному занятию(контроль ный срез)	15	15 баллов – студент выполнил лабораторную работу правильно, расчеты по работе произведены верно, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 8 баллов – студент выполнил лабораторную работу с ошибками, расчеты по работе содержат неточности, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 4 балла – студент выполнил лабораторную работу с грубыми ошибками, не смог правильно провести расчеты и ответить на контрольные вопросы. 0 баллов – студент не выполнил лабораторную работу.
3.	Строки и двоичные данные.	Отчет по лаборатор ному занятию	10	10 баллов – студент выполнил лабораторную работу правильно, расчеты по работе произведены верно, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 6 баллов – студент выполнил лабораторную работу с ошибками, расчеты по работе содержат неточности, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 2 балла – студент выполнил лабораторную работу с грубыми ошибками, не смог правильно провести расчеты и ответить на контрольные вопросы. 0 баллов – студент не выполнил лабораторную работу.
4.	Функции и методы для работы со строками.	Отчет по лаборатор ному занятию	5	5 баллов – студент выполнил лабораторную работу правильно, расчеты по работе произведены верно, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 3 балла – студент выполнил лабораторную работу с ошибками, расчеты по работе содержат неточности, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 1 балл – студент выполнил лабораторную работу с грубыми ошибками, не смог правильно провести расчеты и ответить на контрольные вопросы. 0 баллов – студент не выполнил лабораторную работу.
5.	Списки. Операции над списками	Отчет по лаборатор ному занятию	10	10 баллов – студент выполнил лабораторную работу правильно, расчеты по работе произведены верно, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 6 баллов – студент выполнил лабораторную работу с ошибками, расчеты по работе содержат неточности, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 2 балла – студент выполнил лабораторную работу с грубыми ошибками, не смог правильно провести расчеты и ответить на контрольные вопросы. 0 баллов – студент не выполнил лабораторную работу.

6.	Кортежи, множества и диапазоны	Отчет по лабораторному занятию	5	5 баллов – студент выполнил лабораторную работу правильно, расчеты по работе произведены верно, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 3 балла – студент выполнил лабораторную работу с ошибками, расчеты по работе содержат неточности, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 1 балл – студент выполнил лабораторную работу с грубыми ошибками, не смог правильно провести расчеты и ответить на контрольные вопросы. 0 баллов – студент не выполнил лабораторную работу.
7.	Словари. Операции и методы для работы со словарями.	Отчет по лабораторному занятию(контрольный срез)	15	15 баллов – студент выполнил лабораторную работу правильно, расчеты по работе произведены верно, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 8 баллов – студент выполнил лабораторную работу с ошибками, расчеты по работе содержат неточности, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 4 балла – студент выполнил лабораторную работу с грубыми ошибками, не смог правильно провести расчеты и ответить на контрольные вопросы. 0 баллов – студент не выполнил лабораторную работу.
8.	Работа с датой и временем.	Отчет по лабораторному занятию	10	10 баллов – студент выполнил лабораторную работу правильно, расчеты по работе произведены верно, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 6 баллов – студент выполнил лабораторную работу с ошибками, расчеты по работе содержат неточности, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 2 балла – студент выполнил лабораторную работу с грубыми ошибками, не смог правильно провести расчеты и ответить на контрольные вопросы. 0 баллов – студент не выполнил лабораторную работу.
9.	Пользовательские функции.	Отчет по лабораторному занятию	5	5 баллов – студент выполнил лабораторную работу правильно, расчеты по работе произведены верно, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 3 балла – студент выполнил лабораторную работу с ошибками, расчеты по работе содержат неточности, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 1 балл – студент выполнил лабораторную работу с грубыми ошибками, не смог правильно провести расчеты и ответить на контрольные вопросы. 0 баллов – студент не выполнил лабораторную работу.
10.	Работа с файлами и каталогами.	Отчет по лабораторному занятию	10	10 баллов – студент выполнил лабораторную работу правильно, расчеты по работе произведены верно, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 6 баллов – студент выполнил лабораторную работу с ошибками, расчеты по работе содержат неточности, ответил на контрольные вопросы преподавателя после выполнения лабораторной работы. 2 балла – студент выполнил лабораторную работу с грубыми ошибками, не смог правильно провести расчеты и ответить на контрольные вопросы. 0 баллов – студент не выполнил лабораторную работу.
11.	Посещаемость		10	10 баллов – студент посетил все 100% занятий 7-9 баллов – студент посетил не менее 80% занятий 4-6 баллов – студент посетил не менее 50% занятий 1-3 балла – студент посетил не менее 25% занятий Если студент посетил менее 25% занятий, баллы не начисляются
12.	Итого за семестр		100	

Итоговая оценка по зачету выставляется в 100-балльной шкале и в традиционной четырехбалльной шкале. Перевод 100-балльной рейтинговой оценки по дисциплине в традиционную четырехбалльную осуществляется следующим образом:

100-балльная система	Традиционная система
50 - 100 баллов	Зачтено
0 - 49 баллов	Не зачтено

4.2 Типовые оценочные средства текущего контроля

Отчет по лабораторному занятию

Тема 1. Переменные. Типы данных. Преобразование типов данных.

Тема 1. Переменные. Типы данных. Преобразование типов данных

- 1) Что такое переменная в Python?
- 2) Как указать значение переменной в Python?
- 3) Какие типы данных вы знаете?
- 4) Какие виды числового типа данных вы знаете?
- 5) Как преобразовывать типы данных?

Отчет по лабораторному занятию

Тема 2. Условные операторы и циклы.

Тема 2. Условные операторы и циклы

1. Какие условные операторы и циклы вы знаете?
2. Что делает оператор цикла выполняет?
3. Для чего предназначены операторы break и continue?
4. Для чего используется конструкция if – elif – else?

Тема 3. Строки и двоичные данные.

Тема 3. Строки и двоичные данные

1. Строки в python обозначаются кавычками. Приведите все способы.
2. Какие типы данных можно преобразовать в строку?
3. Опишите синтаксис срезов строк при помощи квадратных скобок.
4. Как применяют операции сложения и умножения к строкам?
5. Что такое двоичные данные?

Тема 4. Функции и методы для работы со строками.

Тема 4. Функции и методы для работы со строками

- 4.1. Арифметические выражения.
- 4.2. Логические выражения.
- 4.3. Строковые выражения.

Тема 5. Списки. Операции над списками

Тема 5. Списки. Операции над списками

1. Перечислите характеристики типа данных «список», которые вы знаете.
2. Как объединить списки?
3. Как умножать списки?
4. Как перевернуть список?

5. В чём разница между `append` и `extend`?

Тема 6. Кортежи, множества и диапазоны

Тема 6. Кортежи, множества и диапазоны.

1. Чем список отличается от других структур?
2. Как объединить два списка в список кортежей?
3. Как работает функция `range`?
4. В каких ситуациях лучше использовать списки, а в каких кортежи, словари или множества?

Тема 7. Словари. Операции и методы для работы со словарями.

Тема 7. Словари. Операции и методы для работы со словарями

1. Какими способами можно получить доступ к значению ключа, не изменяя при этом словарь?
2. Что может служить ключом словаря? Перечислите структуры данных и общие требования к именованию.
3. Для чего нужен метод `pop()`? Какие параметры он может принимать? Приведите пример использования.
4. Охарактеризуйте методы `keys()`, `items()`, `values()`. Что они возвращают, какова специфика результирующих объектов?
5. В чем опасность копирования словаря? Для чего нужно глубокое копирование?

Тема 8. Работа с датой и временем.

Тема 8. Работа с датой и временем

1. Как получить текущую дату и время в Python?
2. Как получить текущую дату?
3. Что внутри `datetime`?
4. Как вывести час, минуту, секунду и микросекунду?

Тема 9. Пользовательские функции.

Тема 9. Пользовательские функции

1. Что такое пользовательская функция?
2. Чем служит ключевое слово `def`?
3. Для чего используется слово `return`?
4. Приведите пример функции Python, которая принимает два параметра, вычисляет сумму и возвращает вычисленное значение.

Тема 10. Работа с файлами и каталогами.

Тема 10. Работа с файлами и каталогами

1. Какими способами можно открыть текстовый файл (в формате `.txt`) в Python (без использования сторонних библиотек)?
2. Опишите основные режимы открытия документа, доступные в функции `open()`.
3. Как используют функцию `print()` для записи файлов?
4. Охарактеризуйте любые 3 объекта модуля `os`.

4.3 Промежуточная аттестация по дисциплине проводится в форме зачета

Типовые вопросы зачета (УК-1)

1. Основные характеристики и критерии оценки алгоритмов. Данные. Понятие типа данных.
2. Понятие типа данных. Константы. Переменные.

3. Основные характеристики и критерии оценки алгоритмов. Целочисленные типы данных.
4. Вещественные типы данных.
5. Основные характеристики и критерии оценки алгоритмов. Символьные типы данных.
6. Булевские типы данных.
7. Определение новых типов данных.
8. Основные характеристики и критерии оценки алгоритмов. Перечисляемые типы данных.
9. Интервальные типы данных.
10. Временной тип данных.
11. Операции. Выражения. Арифметические операции.
- 12 Операции. Операции отношения.
- 13 Операции. Булевские операции.
- 14 Основные характеристики и критерии оценки алгоритмов. Оператор присваивания.
- 15 Оператор ветвления if.
- 16 Оператор ветвления case.
- 17 Операторы повтора — циклы. Оператор повтора for.
- 18 Операторы повтора — циклы. Оператор повтора repeat
- 19 Операторы повтора — циклы. Оператор повтора while
20. Процедуры. Понятие. Свойства. Параметры.
21. Функции. Понятие. Свойства. Параметры.
22. Параметры процедур и функций.
23. Рекурсивные подпрограммы.
24. Строковые переменные.
25. Операции над строками.
26. Форматы кодирования символов.
27. Стандартные процедуры и функции для работы со строками.
28. Объявление массива.
29. Работа с массивами.
30. Понятие файла.
31. Работа с файлами.
32. Стандартные подпрограммы управления файлами

Типовые задания для зачета (УК-1)

Не предусмотрено.

4.4. Шкала оценивания промежуточной аттестации

Оценка	Компетенции	Дескрипторы (уровни) – основные признаки освоения (показатели достижения результата)
«зачтено» (50 - 100 баллов)	УК-1	
«не зачтено» (0 - 49 баллов)	УК-1	

5. Методические указания для обучающихся по освоению дисциплины (модуля)

Методические указания по организации самостоятельной работы обучающихся

Приступая к изучению дисциплины, в первую очередь обучающимся необходимо ознакомиться содержанием рабочей программы дисциплины (РПД), которая определяет содержание, объем, а также порядок изучения и преподавания учебной дисциплины, ее раздела, части.

Для самостоятельной работы важное значение имеют разделы «Объем и содержание дисциплины», «Учебно-методическое и информационное обеспечение дисциплины» и «Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы».

В разделе «Объем и содержание дисциплины» указываются все разделы и темы изучаемой дисциплины, а также виды занятий и планируемый объем в академических часах.

В разделе «Учебно-методическое и информационное обеспечение дисциплины» указана рекомендуемая основная и дополнительная литература.

В разделе «Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы» содержится перечень профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины.

Работая с РПД, необходимо обратить внимание на следующее:

- отдельные разделы или темы дисциплины не разбираются на лекциях, но отводятся на самостоятельное изучение по рекомендуемой учебной литературе и учебно-методическим разработкам;
- при самостоятельном изучении тем следует использовать источники из современных, в том числе международных профессиональных баз данных и информационных справочных систем через «Интернет», состав которых определяется в РПД и ежегодно обновляется;
- усвоение теоретических положений (методик, расчетных формул и др.), входящих в самостоятельно изучаемые темы дисциплины необходимо самостоятельно контролировать по вопросам для самоконтроля в учебных изданиях и в пункте 3.2 РПД;
- материалы тем, отведенных на самостоятельное изучение, в обязательном порядке входят составной частью в темы текущего и промежуточного контроля.

Рекомендации обучающимся по работе с теоретическими материалами по дисциплине

При изучении и проработке теоретического материала необходимо:

- повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;
- при самостоятельном изучении теоретической темы сделать конспект, используя рекомендованные в РПД литературные источники, профессиональные базы данных и информационные справочные системы;
- ответить на контрольные вопросы, по теме представленные в пункте 3.2 РПД.
- при подготовке к текущему контролю использовать материалы фонда оценочных средств (ФОС).

Рекомендации по работе с научной и учебной литературой

Работа с основной и дополнительной литературой является главной формой самостоятельной работы и необходима при подготовке к устному опросу на семинарских занятиях, к контрольным работам, тестированию, зачету/экзамену. Она включает проработку лекционного материала – изучение рекомендованных источников и литературы по тематике лекций.

Конспект лекции должен содержать реферативную запись основных вопросов лекции, предложенных преподавателем схем (при их демонстрации), основных источников и литературы по темам, выводы по каждому вопросу. Конспект должен быть выполнен в отдельной тетради по предмету. Он должен быть аккуратным, хорошо читаемым, не содержать не относящуюся к теме информацию или рисунки.

Конспекты научной литературы при самостоятельной подготовке к занятиям должны содержать ответы на каждый поставленный в теме вопрос, иметь ссылку на источник информации с обязательным указанием автора, названия и года издания используемой научной литературы. Конспект может быть опорным (содержать лишь основные ключевые позиции), но при этом позволяющим дать полный ответ по вопросу, может быть подробным. Объем конспекта определяется самим студентом.

В процессе работы с основной и дополнительной литературой студент может:

- делать записи по ходу чтения в виде простого или развернутого плана (создавать перечень основных вопросов, рассмотренных в источнике);

- составлять тезисы (цитирование наиболее важных мест статьи или монографии, короткое изложение основных мыслей автора);
- готовить аннотации (краткое обобщение основных вопросов работы);
- создавать конспекты (развернутые тезисы).

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Основная литература:

1. Кудрина Е. В., Огнева М. В. Основы алгоритмизации и программирования на языке C# : учебник для вузов. - Москва: Юрайт, 2025. - 322 с. - Текст : электронный // ЭБС «ЮРАЙТ» [сайт]. - URL: <https://urait.ru/bcode/565466>
2. Лубашева, Т. В., Железко, Б. А. Основы алгоритмизации и программирования : учебное пособие. - 2022-08-04; Основы алгоритмизации и программирования. - Минск: Республиканский институт профессионального образования (РИПО), 2016. - 379 с. - Текст : электронный // IPR BOOKS [сайт]. - URL: <http://www.iprbookshop.ru/67689.html>
3. Прищепов М.А., Степанцов В.П., Севернева Е.В. Экзамен по информатике. Основы алгоритмизации и программирования : Справочное пособие. - Мн.: ТетраСистемс, 2001. - 192 с. - Текст : электронный // АБИС "Руслан" ТГУ им. Г.Р. Державина [сайт]
4. Николаев Е. И. Основы алгоритмизации и программирования: лабораторный практикум : практикум. - Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2015. - 211 с. - Текст : электронный // ЭБС «Университетская библиотека онлайн» [сайт]. - URL: <https://biblioclub.ru/index.php?page=book&id=457504>
5. Белоцерковская И. Е., Галина Н. В., Катаева Л. Ю. Алгоритмизация. Введение в язык программирования C++. - 2-е изд., испр.. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 197 с. - Текст : электронный // ЭБС «Университетская библиотека онлайн» [сайт]. - URL: <http://biblioclub.ru/index.php?page=book&id=428935>

6.2 Дополнительная литература:

1. Агафонов, Е. Д., Ващенко, Г. В. Прикладное программирование : учебное пособие. - Весь срок охраны авторского права; Прикладное программирование. - Красноярск: Сибирский федеральный университет, 2015. - 112 с. - Текст : электронный // IPR BOOKS [сайт]. - URL: <http://www.iprbookshop.ru/84092.html>
2. Колокольникова А. И. Информатика : учебное пособие. - 2-е изд., испр. и доп.. - Москва, Берлин: Директ-Медиа, 2020. - 290 с. - Текст : электронный // ЭБС «Университетская библиотека онлайн» [сайт]. - URL: <https://biblioclub.ru/index.php?page=book&id=596690>
3. Агапов, В. П. Основы программирования на языке C# : учебное пособие. - 2024-07-01; Основы программирования на языке C#. - Москва: Московский государственный строительный университет, ЭБС АСВ, 2012. - 128 с. - Текст : электронный // IPR BOOKS [сайт]. - URL: <http://www.iprbookshop.ru/16366.html>
4. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования C++. - М., [Харьков]: АСТ, Фолио, 2004. - 500 с.
5. Фридман А. Л. Язык программирования Си++ : курс лекций. - 2-е изд., исправ.. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 219 с. - Текст : электронный // ЭБС «Университетская библиотека онлайн» [сайт]. - URL: <https://biblioclub.ru/index.php?page=book&id=578114>
6. Шелудько В. М. Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули : учебное пособие. - Ростов-на-Дону|Таганрог: Южный федеральный университет, 2017. - 108 с. - Текст : электронный // ЭБС «Университетская библиотека онлайн» [сайт]. - URL: <http://biblioclub.ru/index.php?page=book&id=500060>

6.3 Методические разработки:

1. Тарланов А. Т., Магомедов Ш. Г. Основы языка программирования Python : учебно-методическое пособие. - Москва: РТУ МИРЭА, 2019. - 107 с. - URL: <https://e.lanbook.com/book/171465>

6.4 Иные источники:

1. elibrary.tsutmb.ru - <https://elibrary.tsutmb.ru/>
2. БД издательства SpringerNature: Home - Springer - URL: - <https://link.springer.com/>
3. БД издательства SpringerNature: SpringerMaterials - свойства материалов. - URL: - <https://materials.springer.com/>
4. Библиотека научной и учебной литературы - <http://sbiblio.com>
5. Библиотека РАН - <http://www.ras.ru/>

7. Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы

Для проведения занятий по дисциплине необходимо следующее материально-техническое обеспечение: учебные аудитории для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещения для самостоятельной работы.

Учебные аудитории и помещения для самостоятельной работы укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа используются наборы демонстрационного оборудования, обеспечивающие тематические иллюстрации (проектор, ноутбук, экран/ интерактивная доска).

Лицензионное и свободно распространяемое программное обеспечение:

Adobe Acrobat 8.0 Standart Russian Version Win Full Educ

Kaspersky Endpoint Security 10 для Windows "Лаборатория Касперского"

Microsoft Windows 10

Операционная система Microsoft Windows 10

Профессиональные базы данных и информационные справочные системы:

1. Электронная библиотека. Образовательная платформа «Юрайт». – URL: <https://biblio-online.ru/book/sud-prisyazhnyh-442275>
2. ЭБС "Лань" - консорциум Сетевых электронных библиотек вузов, коллекция учебников для СПО и учебников для 10-11 кл.. – URL: <https://e.lanbook.com/books>
3. Российская национальная библиотека: официальный сайт. – URL: <http://nlr.ru>
4. Российская государственная библиотека: официальный сайт. – URL: <https://www.rsl.ru>

Электронная информационно-образовательная среда

https://auth.tsutmb.ru/authorize?response_type=code&client_id=moodle&state=xyz

Взаимодействие преподавателя и студента в процессе обучения осуществляется посредством мультимедийных, гипертекстовых, сетевых, телекоммуникационных технологий, используемых в электронной информационно-образовательной среде университета.