

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ПО ДИСЦИПЛИНЕ**
Основы программирования на Python

Код модуля
1162555(1)

Модуль
Современные методы и модели в экономике

Екатеринбург

Оценочные материалы составлены автором(ами):

№ п/п	Фамилия, имя, отчество	Ученая степень, ученое звание	Должность	Подразделение
1	Зверев Владимир Сергеевич	кандидат физико-математических наук, нет	доцент	кафедра теоретической и математической физики

Согласовано:

Управление образовательных программ

И.Ю. Русакова

Авторы:

- Зверев Владимир Сергеевич, доцент, кафедра теоретической и математической физики

1. СТРУКТУРА И ОБЪЕМ ДИСЦИПЛИНЫ Основы программирования на Python

1.	Объем дисциплины в зачетных единицах	3	
2.	Виды аудиторных занятий	Лекции Лабораторные занятия	
3.	Промежуточная аттестация	Зачет	
4.	Текущая аттестация	Домашняя работа	1

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ (ИНДИКАТОРЫ) ПО ДИСЦИПЛИНЕ МОДУЛЯ Основы программирования на Python

Индикатор – это признак / сигнал/ маркер, который показывает, на каком уровне обучающийся должен освоить результаты обучения и их предъявление должно подтвердить факт освоения предметного содержания данной дисциплины, указанного в табл. 1.3 РПМ-РПД.

Таблица 1

Код и наименование компетенции	Планируемые результаты обучения (индикаторы)	Контрольно-оценочные средства для оценивания достижения результата обучения по дисциплине
1	2	3
ПК-6 -Способен работать с различными информационными системами и базами данных о социально-экономических системах, в том числе на иностранном языке	З-2 - Знать различия основных методов анализа статистических данных, контрфактуального анализа и визуализации данных, а также классификации деревьев решений экономико-статистических проблем, связанных с анализом данных П-2 - Владеть навыками работы в пакетах прикладных программ для анализа данных У-2 - Уметь использовать пакеты прикладных программ для решения задач прогнозирования и анализа данных	Домашняя работа Зачет Лабораторные занятия Лекции
ПК-7 -Способен производить статистические	З-2 - Знать современные способы анализа данных	Домашняя работа Зачет Лабораторные занятия

расчеты с применением методов математической статистики и современных технологий для анализа данных, анализировать полученные данные и представлять результаты в виде доклада	методами математической статистики	Лекции
---	------------------------------------	--------

3. ПРОЦЕДУРЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ В РАМКАХ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ МОДУЛЯ В БАЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ (ТЕХНОЛОГИЧЕСКАЯ КАРТА БРС)

3.1. Процедуры текущей и промежуточной аттестации по дисциплине

1. Лекции: коэффициент значимости совокупных результатов лекционных занятий – 0.3		
Текущая аттестация на лекциях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>Работа на лекциях</i>	7	100
Весовой коэффициент значимости результатов текущей аттестации по лекциям – не предусмотрено		
Промежуточная аттестация по лекциям – нет		
Весовой коэффициент значимости результатов промежуточной аттестации по лекциям – не предусмотрено		
2. Практические/семинарские занятия: коэффициент значимости совокупных результатов практических/семинарских занятий – не предусмотрено		
Текущая аттестация на практических/семинарских занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по практическим/семинарским занятиям – не предусмотрено		
Промежуточная аттестация по практическим/семинарским занятиям – нет		
Весовой коэффициент значимости результатов промежуточной аттестации по практическим/семинарским занятиям – не предусмотрено		
3. Лабораторные занятия: коэффициент значимости совокупных результатов лабораторных занятий – 0.7		
Текущая аттестация на лабораторных занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>домашняя работа</i>	8	100

Весовой коэффициент значимости результатов текущей аттестации по лабораторным занятиям -0.6		
Промежуточная аттестация по лабораторным занятиям –зачет		
Весовой коэффициент значимости результатов промежуточной аттестации по лабораторным занятиям – 0.4		
4. Онлайн-занятия: коэффициент значимости совокупных результатов онлайн-занятий –не предусмотрено		
Текущая аттестация на онлайн-занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по онлайн-занятиям -не предусмотрено		
Промежуточная аттестация по онлайн-занятиям –нет		
Весовой коэффициент значимости результатов промежуточной аттестации по онлайн-занятиям – не предусмотрено		

3.2. Процедуры текущей и промежуточной аттестации курсовой работы/проекта

Текущая аттестация выполнения курсовой работы/проекта	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент текущей аттестации выполнения курсовой работы/проекта– не предусмотрено		
Весовой коэффициент промежуточной аттестации выполнения курсовой работы/проекта– защиты – не предусмотрено		

4. КРИТЕРИИ И УРОВНИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ МОДУЛЯ

4.1. В рамках БРС применяются утвержденные на кафедре/институте критерии (признаки) оценивания достижений студентов по дисциплине модуля (табл. 4) в рамках контрольно-оценочных мероприятий на соответствие указанным в табл.1 результатам обучения (индикаторам).

Таблица 4

Критерии оценивания учебных достижений обучающихся

Результаты обучения	Критерии оценивания учебных достижений, обучающихся на соответствие результатам обучения/индикаторам
Знания	Студент демонстрирует знания и понимание в области изучения на уровне указанных индикаторов и необходимые для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Умения	Студент может применять свои знания и понимание в контекстах, представленных в оценочных заданиях, демонстрирует освоение умений на уровне указанных индикаторов и необходимых для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Опыт /владение	Студент демонстрирует опыт в области изучения на уровне указанных индикаторов.

Другие результаты	<p>Студент демонстрирует ответственность в освоении результатов обучения на уровне запланированных индикаторов.</p> <p>Студент способен выносить суждения, делать оценки и формулировать выводы в области изучения.</p> <p>Студент может сообщать преподавателю и коллегам своего уровня собственное понимание и умения в области изучения.</p>
-------------------	---

4.2 Для оценивания уровня выполнения критериев (уровня достижений обучающихся при проведении контрольно-оценочных мероприятий по дисциплине модуля) используется универсальная шкала (табл. 5).

Таблица 5

Шкала оценивания достижения результатов обучения (индикаторов) по уровням

Характеристика уровней достижения результатов обучения (индикаторов)				
№ п/п	Содержание уровня выполнения критерия оценивания результатов обучения (выполненное оценочное задание)	Шкала оценивания		
		Традиционная характеристика уровня		Качественная характеристика уровня
1.	Результаты обучения (индикаторы) достигнуты в полном объеме, замечаний нет	Отлично (80-100 баллов)	Зачтено	Высокий (В)
2.	Результаты обучения (индикаторы) в целом достигнуты, имеются замечания, которые не требуют обязательного устранения	Хорошо (60-79 баллов)		Средний (С)
3.	Результаты обучения (индикаторы) достигнуты не в полной мере, есть замечания	Удовлетворительно (40-59 баллов)		Пороговый (П)
4.	Освоение результатов обучения не соответствует индикаторам, имеются существенные ошибки и замечания, требуется доработка	Неудовлетворительно (менее 40 баллов)	Не зачтено	Недостаточный (Н)
5.	Результат обучения не достигнут, задание не выполнено	Недостаточно свидетельств для оценивания		Нет результата

5. СОДЕРЖАНИЕ КОНТРОЛЬНО-ОЦЕНОЧНЫХ МЕРОПРИЯТИЙ ПО ДИСЦИПЛИНЕ МОДУЛЯ

5.1. Описание аудиторных контрольно-оценочных мероприятий по дисциплине модуля

5.1.1. Лекции

Самостоятельное изучение теоретического материала по темам/разделам лекций в соответствии с содержанием дисциплины (п. 1.2. РПД)

5.1.2. Лабораторные занятия

Примерный перечень тем

1. Понятие типов данных
 2. Базовые конструкции языка
 3. Контейнерные типы данных в Python
 4. Модули для обработки и визуализации данных
- LMS-платформа – не предусмотрена

5.2. Описание внеаудиторных контрольно-оценочных мероприятий и средств текущего контроля по дисциплине модуля

Разноуровневое (дифференцированное) обучение.

Базовый

5.2.1. Домашняя работа

Примерный перечень тем

1. Базовые конструкции языка
2. Контейнерные типы данных в Python

Примерные задания

1. В григорианском календаре обычный год состоит из 365 дней. Поскольку фактическая длительность звёздного года (время, необходимое Земле для одного оборота вокруг Солнца) 365.25635 дней, «високосный год» 366 дней используется один раз в четыре года для устранения данной ошибки, вызванной тремя обычными (но короткими) годами. Любой год, число которого делится на 4 —високосный год: например, 1988, 1992 и 1996 года являются високосными. Однако по-прежнему есть небольшая ошибка, которая должна быть учтена. Чтобы устранить эту ошибку, григорианский календарь определяет, что год, делится на 100 (например, 1900) —является високосным, только если он также делится на 400. Пусть x –текущий год, определяется вводом пользователя, преобразованный к целому числу, то есть $x = \text{int}(\text{input}())$. Выведите на экран "Високосный" в случае, если считанный год является високосным и "Обычный" в обратном случае (не забывайте проверять регистр выводимых программой символов). Запишите программу в файл `big_hw_1_1.py`. В ответе покажите корректность работы программы на 10 примерах. Программа должна корректно работать на числах $1900 \leq n \leq 3000$.

2. У коллег Тимофея Бондарева и Павла Федотова в институте биоинформатики по офису передвигается робот. Они решили предложить студентам из группы программистов написать для него программу, по которой робот, когда заходит в комнату, считает количество программистов в ней и произносит его вслух: "n программистов". Для того, чтобы это звучало правильно, для каждого n нужно использовать верное окончание слова. Напишите программу, считывающую с пользовательского ввода (с помощью команды `input`) целое число n (неотрицательное), выводящее это число на экран вместе с правильным образом изменённым словом «программист» для того, чтобы робот мог нормально общаться с людьми, например: 1 программист, 2 программиста, 5 программистов. В комнате может быть очень много программистов. Проверьте, что ваша

программа правильно обработает все случаи, как минимум до 1000 человек.Результат будет проверен автоматически, поэтому запишите программу в big_hw_1_2.py

3. У Тимофея Бондарева и Павла Федотова из института биоинформатики, много занятых задачек и вот еще одна:Напишите программу, которая считывает со стандартного ввода целые числа (с помощью команды `input`), по одному числу в строке, и после первого введенного нуля выводит на экран сумму полученных на вход чисел.Запишите программу в big_hw_1_3.py

4. Оформите вычисление квадрата площади треугольника по формуле Герона и самой площади в виде функции.

5.Специалисты из института биоинформатики также предложили идею, продолжающую задачу подсчета символов, которую мы решали на паре.

Вот в чем суть. Все данные хранятся в сжатом виде. Один из простых алгоритмов архивации без потери информации записывает количество повторов. Например, строка `s = 'aaaabbbcaa'` преобразуется в `'a4b2c1a2'`, то есть группы одинаковых символов исходной строки заменяются на этот символ и количество его повторений в этой позиции строки. Более оптимальный вариант не дописывает ничего для символов, которые встречаются только 1 раз, то есть `s = 'aaaabbbcaa'` преобразуется в `'a4b2ca2'` и итог занимает еще меньше места. Напишите программу, которая считывает строку, кодирует её предложенным оптимизированным алгоритмом и выводит закодированную последовательность на стандартный вывод(экран). Кодирование должно учитывать регистр символов. Результат будет проверен автоматически, поэтому запишите программу в big_hw_1_5.py

6.Для логического выражения составить таблицу истинности для логической функции от трех аргументов в виде:

a	b	c	ваша_функция
True	False	True	результат_функции_в_этом_случае
False	True	False	результат_функции_в_этом_случае....

Логическая функция у каждого своя, написана в таблице ниже. Должны быть результаты для всех комбинаций значений переменных.

Функция для вычисления. Каждый выбирает только одну в соответствии с номером в списке группы

- 1.a or not (b or not c)
- 2.not (a and not b) or not c
- 3.(a and not b) and c
- 4.a or (b and c)
- 5.not a and not (b and not c)
- 6.not a or not b and c
- 7.a or b and not c
- 8.not a or not (b or not c)
- 9.not a or b or c
- 10.not (a and b) and not c
- 11.a and b or not c
- 12.not a and b and c

13. not (a and b) and c
14. a or not b and c
15. not (a or b) and c
16. not a or not b and not c
17. a or b and c
18. not a or b or not c
19. not a and (b or not c)
20. (a or b) or c
21. a and (b or not c)
22. a and (b and not c)
23. (a or not b) and c

7. Создайте функцию, которая принимает в качестве параметров строку и положительное число `split_number`, которая меньше длины строки.

Функция должна:

- проверить, что `split_number` меньше длины строки, иначе выдать пользователю сообщение;
- разделить строку на две части, одна из них должна иметь длину `split_number`;
- вернуть строку, в которой части исходной строки поменяны местами, одна из подстрок должна иметь длину `split_number`;
- создайте примеры, демонстрирующие работу функции.

Запишите программу в `big_hw_1_7.py`

8. Создайте функцию, которая из переданной строки убирает все двойные пробелы. Создайте примеры, демонстрирующие работу функции. Запишите программу в `big_hw_1_8.py`

9. Создайте функцию, которая принимает как параметры строку и букву `letter`. Удалите из этой строки первое и последнее вхождение буквы `letter`, а также все символы, находящиеся между ними. Создайте примеры, демонстрирующие работу функции. Запишите программу в `big_hw_1_9.py`

10. Создайте функцию для создания словаря (напомню, что имеется ввиду <https://pythontutor.ru/lessons/dicts/>) слов с указанием, частоты их вхождения. На вход функция принимает строку. Результат – словарь вида {«слово»: 0.75}. Создайте примеры, демонстрирующие работу функции. Запишите программу в `big_hw_1_10.py`
LMS-платформа – не предусмотрена

5.3. Описание контрольно-оценочных мероприятий промежуточного контроля по дисциплине модуля

5.3.1. Зачет

Список примерных вопросов

1. Каков результат выполнения кода? `str = "pynative" print (str[1:3])` -py -yn -pyn -yna

2. Каков будет результат выполнения кода? `sampleList = ["Jon", "Kelly", "Jessa"]
sampleList.append(2, "Scott") print(sampleList)` -Выполнение программы закончится ошибкой
-['Jon', 'Kelly', 'Scott', 'Jessa'] -['Jon', 'Kelly', 'Jessa', 'Scott'] -['Jon', 'Scott', 'Kelly', 'Jessa']
3. Что выведется в результате? `x = 36 / 4 * (3 + 2) * 4 + 2 print(x)` -182 -37 -117 -
Выполнение программы закончится ошибкой
4. Каков будет результат выполнения кода? `var = "James" * 2 * 3 print(var)` -
JamesJamesJamesJamesJamesJames -JamesJamesJamesJamesJames -Error: invalid syntax
5. Каков будет результат выполнения кода? `var1 = 1 var2 = 2 var3 = "3" print(var + var2 + var3)` -6 -33 -123 - Error. Mixing operators between numbers and strings are not supported
6. Каков будет результат выполнения кода? `listOne = [20, 40, 60, 80] listTwo = [20, 40, 60, 80] print(listOne == listTwo) print(listOne is listTwo)` - True True - True False - False True
7. Для чего используется оператор 'in'? -Возвращает True если последовательность присутствует в объекте - Возвращает False если последовательность присутствует в объекте - Для типа string `x in y = True` тогда и только тогда, когда `x` является подстрокой `y` - Для типа string `x in y = False` тогда и только тогда, когда `x` является подстрокой `y`
8. Для чего используется оператор 'not in'? - Возвращает True если последовательность присутствует в объекте - Возвращает False если последовательность присутствует в объекте - Для типа string `x in y = True` тогда и только тогда, когда `x` является подстрокой `y` -Для типа string `x not in y = False` тогда и только тогда, когда `x` является подстрокой `y`
9. Каков будет результат выполнения кода? `for x in range(0.5, 5.5, 0.5): print(x)` - [0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5] - [0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5] - Выполнение программы закончится ошибкой
10. Каков будет результат выполнения кода? `p, q, r = 10, 20 ,30 print(p, q, r)` - 10 20 - 10 20 30 - Error: invalid syntax
11. Каков будет результат выполнения кода? `salary = 8000 def printSalary(): salary = 12000 print("Salary:", salary) printSalary(); print("Salary:", salary)` - Salary: 12000 Salary: 8000 - Salary: 8000 Salary: 12000 - Выполнение завершится ошибкой
12. Каков будет результат выполнения кода? `valueOne = 5 ** 2 valueTwo = 5 ** 3 print(valueOne) print(valueTwo)?` - 10 15 - 25 125 - Error: invalid syntax
13. Как бы вы записали постоянное значение с плавающей запятой 3.2 умножить на 10 в степени -12 в Python? Напишите, пожалуйста, ваш вариант _____
14. Какой будет результат выполнения кода в python 3 — `print(3/5)?` - 0 - 0.6 - 1 - None
15. Какой будет результат выполнения этого кода? `abra = ['a', 'aa', 'bb', 'b', 'ra'] print(abra[int(1/2)])` - bbb - Ошибка - a - 'a', 'aa', 'aaa', 'b', 'bb', 'bbb'
16. Какой будет результат выполнения кода — `print('Monty' + 'Python')` ? - MontyPython - Monty Python - Python - Monty
17. Что выведет этот код? `x, y, z = True, False, False if not x or y: print(1) elif not x or not y and z: print(2) elif nox x or y or not y and x: print(3) else: print(4)` -3 -4 -1 -2
18. Что выведет этот код? `X, Y, Z = True, False, False if not x or y: print(1) elif not x or not y and z: print(2) elif nox x or y or not y and x: print(3) else: print(4)` -3 -4 -Сообщение об ошибке -Ничего из указанного
19. Каков будет результат выполнения программы? `import re string = "The rain in Spain" x = re.sub("\s", "9", string) print(x)` -Сообщение об ошибке -The9rain in Spain - The9rain9in9Spain -The rain in Spain

20. Каков будет результат выполнения программы? `import re string = "The rain in Spain"`
`x = re.sub("\s", "9", sttring) print(x)` -Сообщение об ошибке -The9rain in Spain -
The9rain9in9Spain -The rain in Spain

21. Что выведет этот код? `def test1(param): return param` `def test2(param): return param * 2`
`def test3(param): return param + 3` `result = test(test2(test3(1)))` -3 -6 -1 -8

LMS-платформа – не предусмотрена

5.4 Содержание контрольно-оценочных мероприятий по направлениям воспитательной деятельности

Направления воспитательной деятельности сопрягаются со всеми результатами обучения компетенций по образовательной программе, их освоение обеспечивается содержанием всех дисциплин модулей.