|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра прикладной математики (ПМ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Технологии и инструментарий анализа больших данных»

**Практическое занятие № 1**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИМБО-02-22, Ким Кирилл Сергеевич* | (подпись) | |
| Преподаватель | *Тетерин Николай Николаевич, преподаватель* | (подпись) | |
| Отчет представлен | «\_\_\_»\_\_\_\_\_\_\_\_202\_\_г. | |  | |

Москва 2025 г.

СОДЕРЖАНИЕ

[ПРАКТИЧЕСКАЯ РАБОТА 1 3](#_Toc208920369)

[Введение 3](#_Toc208920370)

[Шаги выполнения 3](#_Toc208920371)

[Результат работы: 9](#_Toc208920372)

[Вывод: 9](#_Toc208920373)

[Список использованных источников и литературы: 10](#_Toc208920374)

ПРАКТИЧЕСКАЯ РАБОТА 1

Введение

**Цель**

Познакомиться с языком программирования python, установить Anaconda и выполнить задачи.

Шаги выполнения

1. Написать программу, которая вычисляет площадь фигуры, параметры которой подаются на вход. Фигуры, которые подаются на вход: треугольник, прямоугольник, круг. Результатом работы является словарь, где ключ – это название фигуры, а значение – это площадь.

Листинг 1 – 2 задача

|  |
| --- |
| s = input()  d = {"треугольник": 0,  "прямоугольник": 0,  "круг": 0}  if s == "треугольник":  a = float(input())  h = float(input())  S = 0.5 \* a \* h  d["треугольник"] = S    elif s == "прямоугольник":  a = float(input())  b = float(input())  S = a \* b  d["прямоугольник"] = S    elif s == "круг":  r = float(input())  pi = 3.14  S = r \* r \* pi  d["круг"] = S  print(d) |

1. Написать программу, которая на вход получает два числа и операцию, которую к ним нужно применить. Должны быть реализованы следующие операции: +, -, /, //, abs – модуль, pow или \*\* – возведение в степень.

Листинг 2 – 3 задача

|  |
| --- |
| a = int(input())  b = int(input())  sign = input()  if (sign == "+"):  print(a + b)  elif (sign == "-"):  print(a - b)  elif (sign == "/"):  if b == 0:  print("На нуль не делится!")  else:  print(a / b)  elif (sign == "//"):  if b == 0:  print("На нуль не делится!")  else:  print(a // b)  elif (sign == "abs"):  print(abs(a), abs(b))  elif (sign == "pow" or sign == "\*\*"):  print(a \*\* b) |

1. Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и после этого выводит сумму квадратов всех считанных чисел.

Листинг 3 – 4 задача

|  |
| --- |
| total = 0  sums = 0  while (True):  num = float(input())  total += num  sums += num \*\* 2  if total == 0:  break    print(sums)  sum\_of\_squares = sum(x \*\* 2 for x in numbers)  print(sum\_of\_squares) |

1. Напишите программу, которая выводит последовательность чисел, длинною N, где каждое число повторяется столько раз, чему оно равно. На вход программе передаётся неотрицательное целое число N. Например, если N = 7, то программа должна вывести 1 2 2 3 3 3 4. Вывод элементов списка через пробел – print(\*list).

Листинг 4 – 5 задача

|  |
| --- |
| N = int(input())  lst = []  for i in range(N):  lst.append(int(input()))  print(\*lst) |

1. Даны два списка: А = [1, 2, 3, 4, 2, 1, 3, 4, 5, 6, 5, 4, 3, 2] В = [‘a’, ’b’, ’c’, ’c’, ’c’, ’b’, ’a’, ’c’, ’a’, ’a’, ’b’, ’c’, ’b’, ’a’] Создать словарь, в котором ключи – это содержимое списка В, а значения для ключей словаря – это сумма всех элементов списка А в соответствии с буквой, содержащийся на той же позиции в списке В. Пример результата программы: {‘a’ : 10, ‘b’ : 15, ‘c’ : 6}.

Листинг 5 – 6 задача

|  |
| --- |
| A = [1, 2, 3, 4, 2, 1, 3, 4, 5, 6, 5, 4, 3, 2]  B = ['a', 'b', 'c', 'c', 'c', 'b', 'a', 'c', 'a', 'a', 'b', 'c', 'b', 'a']    my\_dict = {}  for i in range(len(A)):  key = B[i]  value = A[i]  if (key in my\_dict):  my\_dict[key] += value  else:  my\_dict[key] = value  print(my\_dict) |

1. Скачать и загрузить данные о стоимости домов в калифорнии,

используя библиотеку sklearn.

Листинг 6 – 7 задача

|  |
| --- |
| from sklearn.datasets import fetch\_california\_housing  import pandas as pd  data = fetch\_california\_housing(as\_frame=True)  df = data.frame  df |

1. Использовать метод info().

Листинг 7 – 8 задача

|  |
| --- |
| df.info() |

1. Узнать, есть ли пропущенные значения, используя isna().sum().

Листинг 8 – 9 задача

|  |
| --- |
| df.isna().sum() |

1. Вывести записи, где средний возраст домов в районе более 50 лет и

население более 2500 человек, используя метод loc().

Листинг 9 – 10 задача

|  |
| --- |
| df.loc[(df['HouseAge'] > 50) & (df['Population'] > 2500)] |

1. Узнать максимальное и минимальное значения медианной стоимости дома.

Листинг 10 – 11 задача

|  |
| --- |
| print(df['MedHouseVal'].max())  print(df['MedHouseVal'].min()) |

1. Используя метод apply(), вывести на экран название признака и его среднее значение.

Листинг 11 – 12 задача

|  |
| --- |
| def func(x):  mean = x.mean()  return mean    a = df.apply(func, axis=0)  print(a) |

1. Дан текст на английском языке. Необходимо закодировать его с помощью азбуки Морзе, где каждой букве соответствует последовательность точек и тире. Например, буква «g» превратится в строку «--.». В переменной morze для удобства хранится словарь соответствия латинских букв коду Морзе.

Листинг 12 – 1 задача повышенной сложности

|  |
| --- |
| morze = {'a': '.-', 'b': '-...', 'c': '-.-.', 'd': '-..',  'e': '.', 'f': '..-.', 'g': '--.', 'h': '....',  'i': '..', 'j': '.---', 'k': '-.-', 'l': '.-..',  'm': '--', 'n': '-.', 'o': '---', 'p': '.--.',  'q': '--.-', 'r': '.-.', 's': '...', 't': '-',  'u': '..-', 'v': '...-', 'w': '.--', 'x': '-..-',  'y': '-.--', 'z': '--..'}  s = input().lower()  words = s.split()  for word in words:  abc = []  for char in word:  if char in morze:  abc.append(morze[char])  print(' '.join(abc)) |

1. В некотором городе открывается новая служба по доставке электронных писем. Необходимо наладить систему регистрации новых пользователей. Регистрация должна работать следующим образом: если новый пользователь хочет зарегистрироваться на сайте, то он должен послать системе запрос name со своим именем. Система должна определить, существует ли уже такое имя в базе данных. Если такого имени не существует, то оно заносится в базу данных системы и пользователю возвращается ответ "ОК", подтверждающий успешную регистрацию. А если пользователь с таким именем уже существует, то система должна сформировать новое имя и выдать его пользователю в качестве подсказки, при этом сама подсказка также добавляется в базу данных. Новое имя формируется следующим образом: к name последовательно приписываются числа, начиная с 1 (name1, name2 и так далее), и среди них находят такое наименьшее i, что namei еще не содержится в системе.

Листинг 13 – 2 задача повышенной сложности

|  |
| --- |
| n = int(input())  data = {}  for i in range(n):  name = input()    if name not in data:  print("OK")  data[name] = 0  else:  data[name] += 1  new\_name = f"{name}{data[name]}"  while new\_name in data:  data[name] += 1  new\_name = f"{name}{data[name]}"  data[new\_name] = 0  print(new\_name) |

1. Необходимо создать программу обработки запросов пользователей к файловой системе компьютера. Над каждым файлом можно производить следующие действия: запись – w ("write"), чтение – r ("read"), запуск – x ("execute").

Листинг 14 – 3 задача повышенной сложности

|  |
| --- |
| n = int(input())  files = {}  for i in range(n):  data = input().split()  filename = data[0]  permissions = data[1:]  files[filename] = set(permissions)  # Обработка запросов  m = int(input())  results = []  for i in range(m):  operation, filename = input().split()  if operation == "write":  op\_code = "w"  elif operation == "read":  op\_code = "r"  elif operation == "execute":  op\_code = "x"  else:  op\_code = ""  if filename in files and op\_code in files[filename]:  results.append("OK")  else:  results.append("Access denied")  for result in results:  print(result) |

Результат работы:

Данную работу можете увидеть в блокноте Jupyter Notebook.

<https://drive.google.com/file/d/1xDe45gMO5Up1vN2eQxu5MQELXuSKP6kN/view?usp=sharing>

Вывод:

Можно сделать вывод, что установили Anaconda, вспомнили Python, написали первые задачи, поработали со словарями и с pandas.

Список использованных источников и литературы:

1. Ростовцев В.С. Искусственные нейронные сети,   
   Издательство "Лань", 2019. — 216 с. — URL: https://e.lanbook.com/book/122180
2. Араки М. Манга: Машинное обучение,   
   Издательство "ДМК Пресс", 2020. — 214 с. — URL: <https://e.lanbook.com/book/179473>
3. Платонов, А. В. Машинное обучение : учебное пособие для вузов / А. В. Платонов. — Москва : Издательство Юрайт, 2022. — 85 с. — (Высшее образование). — ISBN 978-5-534-15561-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/508804