



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 *ИКБО-01-22 Прокопчук Роман Олегович*

(подпись)

Ассистент *Тетерин Николай Николаевич*

(подпись)

Отчёт представлен «__» сентября 2025 г.

Москва, 2025 г.

ЦЕЛЬ

Выполнить все задания.

ХОД РАБОТЫ

Задание 2

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.

```
import math

Windsurf: Refactor | Explain | Docstring | ✕
def calculate_triangle_area(a, b, c): 1 usage new *
    s = (a + b + c) / 2
    area = math.sqrt(s * (s - a) * (s - b) * (s - c))
    return area

Windsurf: Refactor | Explain | Docstring | ✕
def calculate_rectangle_area(length, width): 1 usage new *
    return length * width

Windsurf: Refactor | Explain | Docstring | ✕
def calculate_circle_area(radius): 1 usage new *
    return math.pi * (radius ** 2)

Windsurf: Refactor | Explain | Docstring | ✕
def main(): 1 usage new *
    shape = input("Введите название фигуры (треугольник, прямоугольник, круг): ").lower()

    result = {}

    if shape == "треугольник":
        a = float(input("Введите длину стороны a: "))
        b = float(input("Введите длину стороны b: "))
        c = float(input("Введите длину стороны c: "))
        area = calculate_triangle_area(a, b, c)
        result[shape] = area

    elif shape == "прямоугольник":
        length = float(input("Введите длину: "))
        width = float(input("Введите ширину: "))
        area = calculate_rectangle_area(length, width)
        result[shape] = area

    elif shape == "круг":
        radius = float(input("Введите радиус: "))
        area = calculate_circle_area(radius)
        result[shape] = area

    else:
        print("Неизвестная фигура. Пожалуйста, выберите из списка: треугольник, прямоугольник, круг.")
        return

    if result:
        print(result)

if __name__ == "__main__":
    main()
```

Рисунок 1 – Код программы

```
2 x
C:\Users\Remsely\dev\mirea\python\big-data\.venv\Scripts\python.exe C:\Users\Remsely\dev\mirea\python\big-data\2\2.py
Введите название фигуры (треугольник, прямоугольник, круг): прямоугольник
Введите длину: 4
Введите ширину: 2
{'прямоугольник': 8.0}
```

Рисунок 2 – Пример вывода программы

Задание 3

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.

```
import math

Windsurf: Refactor | Explain | Docstring | X
def calculator(): 1 usage new *
    num1_str = input("Введите первое число: ")
    num1 = float(num1_str)

    op = input("Введите операцию (+, -, /, //, abs, pow, **): ").strip()

    if op != 'abs':
        num2_str = input("Введите второе число: ")
        num2 = float(num2_str)
    else:
        result = abs(num1)
        print(f"Модуль числа {num1} равен {result}")
        return

    if op == '+':
        result = num1 + num2
    elif op == '-':
        result = num1 - num2
    elif op == '/':
        if num2 == 0:
            print("Ошибка: Деление на ноль невозможно.")
            return
        result = num1 / num2
    elif op == '//':
        if num2 == 0:
            print("Ошибка: Деление на ноль невозможно.")
            return
        result = num1 // num2
    elif op in ('pow', '**'):
        result = math.pow(num1, num2)
    else:
        print(f"Неизвестная операция: '{op}'.")
        return

    print(f"Результат: {num1} {op} {num2} = {result}")

if __name__ == "__main__":
    calculator()
```

Рисунок 3 – Код программы



```
C:\Users\Remsely\dev\mirea\python\big-data\.venv\Scripts\python.exe C:\Users\Remsely\dev\mirea\python\big-data\2\3.py
Введите первое число: 2
Введите операцию (+, -, /, //, abs, pow, **): /
Введите второе число: 3
Результат: 2.0 / 3.0 = 0.6666666666666666
```

Рисунок 4 – Пример вывода

Задание 4

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.

```
def sum_squares_until_sum_is_zero(): 1 usage new *
    numbers = []
    current_sum = 0

    print("Введите числа (по одному в строке).")
    print("Ввод закончится, когда сумма всех введённых чисел станет равна 0.")

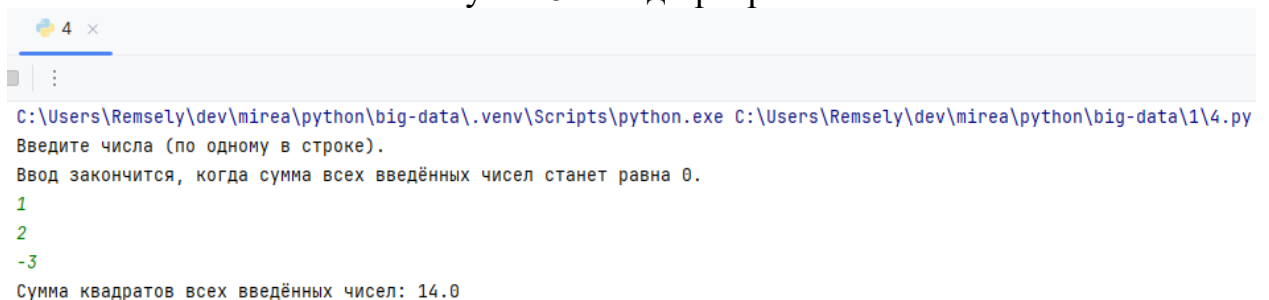
    while True:
        input_str = input()
        number = float(input_str)
        numbers.append(number)
        current_sum += number

        if current_sum == 0:
            break

    sum_of_squares = sum(x ** 2 for x in numbers)
    print(f"Сумма квадратов всех введённых чисел: {sum_of_squares}")

if __name__ == "__main__":
    sum_squares_until_sum_is_zero()
```

Рисунок 5 – Код программы



```
C:\Users\Remsely\dev\mirea\python\big-data\.venv\Scripts\python.exe C:\Users\Remsely\dev\mirea\python\big-data\1\4.py
Введите числа (по одному в строке).
Ввод закончится, когда сумма всех введённых чисел станет равна 0.
1
2
-3
Сумма квадратов всех введённых чисел: 14.0
```

Рисунок 6 – Пример вывода

Задание 5

Напишите программу, которая выводит последовательность чисел, длиною 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)`.

```
def main(): 1 usage new *
    n = int(input("Введите неотрицательное целое число N: "))

    if n == 0:
        print()
        return

    sequence = []
    current_number = 1

    while len(sequence) < n:
        for _ in range(current_number):
            sequence.append(current_number)
            if len(sequence) == n:
                break
        current_number += 1

    print(*sequence)

if __name__ == "__main__":
    main()
```

Рисунок 7 – Код программы

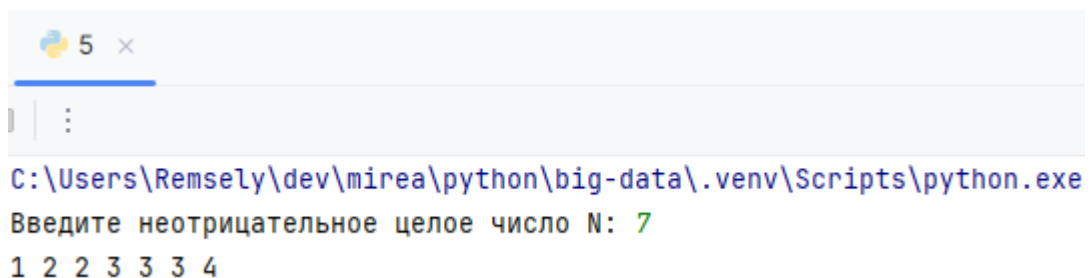


Рисунок 8 – Пример вывода

Задание 6

Даны два списка:

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']

Создать словарь, в котором ключи – это содержимое списка B, а значения для ключей словаря – это сумма всех элементов списка A в соответствии с буквой,


содержащийся на той же позиции в списке B. Пример результата программы: {‘a’: 10, ‘b’: 15, ‘c’: 6}.

```
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']

result_dict_zip = {}
for value, key in zip(A, B):
    result_dict_zip[key] = result_dict_zip.get(key, 0) + value

print(result_dict_zip)
```

Рисунок 9 – Код программы



```
C:\Users\Remsely\dev\mirea\python\big-data\.venv\Scripts\python.exe
{'a': 17, 'b': 11, 'c': 17}
```

Рисунок 10 – Результат вывода

Задание 7

Скачать и загрузить данные о стоимости домов в калифорнии, используя библиотеку sklearn.

```
from sklearn.datasets import fetch_california_housing

data = fetch_california_housing(as_frame=True)
```

Рисунок 11 – Код программы

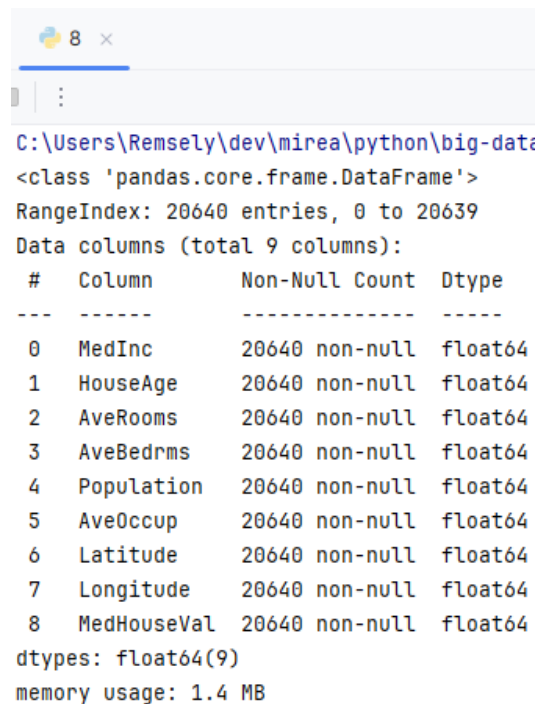
Задание 8

Использовать метод info().

```
from sklearn.datasets import fetch_california_housing

data = fetch_california_housing(as_frame=True)
data.frame.info()
```

Рисунок 12 – Код программы



```

C:\Users\Remsely\dev\mirea\python\big-data
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   MedInc          20640 non-null  float64
1   HouseAge        20640 non-null  float64
2   AveRooms        20640 non-null  float64
3   AveBedrms       20640 non-null  float64
4   Population      20640 non-null  float64
5   AveOccup        20640 non-null  float64
6   Latitude        20640 non-null  float64
7   Longitude       20640 non-null  float64
8   MedHouseVal     20640 non-null  float64
dtypes: float64(9)
memory usage: 1.4 MB

```

Рисунок 13 – Результат вывода

Задание 9

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

```
from sklearn.datasets import fetch_california_housing
```

```

Windsurf: Refactor | Explain | Docstring | X
def check_missing_values_california_housing(): 1 usage new *
    california_housing = fetch_california_housing(as_frame=True)
    df = california_housing.frame
    print("Проверка на пропущенные значения в каждом столбце:")
    missing_values_count = df.isna().sum()
    print(missing_values_count)

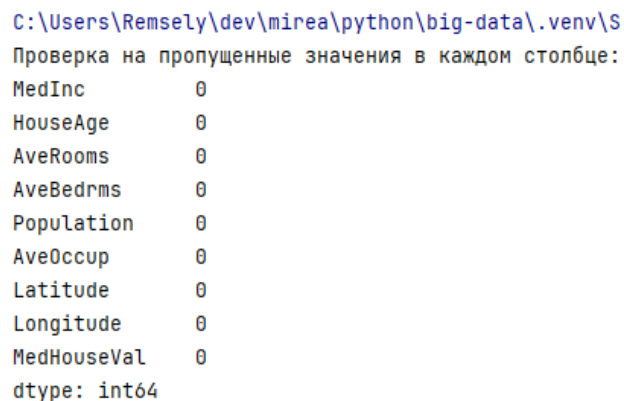
```

```

if __name__ == "__main__":
    check_missing_values_california_housing()

```

Рисунок 14 – Код программы



```

C:\Users\Remsely\dev\mirea\python\big-data\.venv\S
Проверка на пропущенные значения в каждом столбце:
MedInc          0
HouseAge        0
AveRooms        0
AveBedrms       0
Population      0
AveOccup        0
Latitude        0
Longitude       0
MedHouseVal     0
dtype: int64

```

Рисунок 15 – Результат вывода

Задание 10

Вывести записи, где средний возраст домов в районе более 50 лет и население более 2500 человек, используя метод `loc()`.

```
from sklearn.datasets import fetch_california_housing

housing = fetch_california_housing(as_frame=True)
df = housing.frame

filtered_data = df.loc[(df['HouseAge'] > 50) & (df['Population'] > 2500)]

print(filtered_data)
```

Рисунок 16 – Код программы

```
C:\Users\Remsely\dev\mirea\python\big-data\.venv\Scripts\python.exe C:\Us
      MedInc  HouseAge  AveRooms  ...  Latitude  Longitude  MedHouseVal
460      1.4012      52.0  3.105714  ...    37.87    -122.26      1.75000
4131     3.5349      52.0  4.646119  ...    34.13    -118.20      1.93600
4440     2.6806      52.0  4.806283  ...    34.08    -118.21      1.53000
5986     1.8750      52.0  4.500000  ...    34.10    -117.71      2.12500
7369     3.1901      52.0  4.730942  ...    33.97    -118.21      1.67600
8227     2.3305      52.0  3.488860  ...    33.78    -118.20      1.62500
13034    6.1359      52.0  8.275862  ...    38.69    -121.15      2.25000
15634    1.8295      52.0  2.628169  ...    37.80    -122.41      2.43800
15652    0.9000      52.0  2.237474  ...    37.80    -122.41      5.00001
15657    2.5166      52.0  2.839075  ...    37.79    -122.41      2.75000
15659    1.7240      52.0  2.278566  ...    37.79    -122.41      2.25000
15795    2.5755      52.0  3.402576  ...    37.77    -122.42      3.25000
15868    2.8135      52.0  4.584329  ...    37.76    -122.41      2.60300

[13 rows x 9 columns]
```

Рисунок 17 – Результат вывода

Задание 11

Узнать максимальное и минимальное значения медианной стоимости дома.

```
from sklearn.datasets import fetch_california_housing

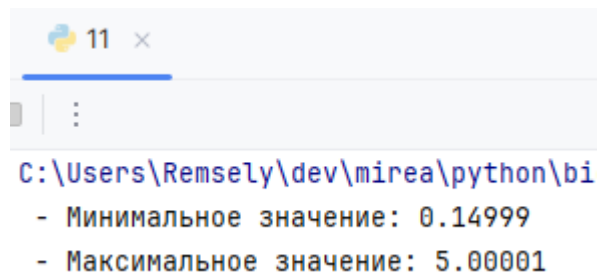
housing = fetch_california_housing(as_frame=True)
df = housing.frame

median_house_value_series = df['MedHouseVal']

min_value = median_house_value_series.min()
max_value = median_house_value_series.max()

print(f" - Минимальное значение: {min_value}")
print(f" - Максимальное значение: {max_value}")
```

Рисунок 18 – Код программы



```
C:\Users\Remsely\dev\mirea\python\bi
- Минимальное значение: 0.14999
- Максимальное значение: 5.00001
```

Рисунок 19 – Результат вывода

Задание 12

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

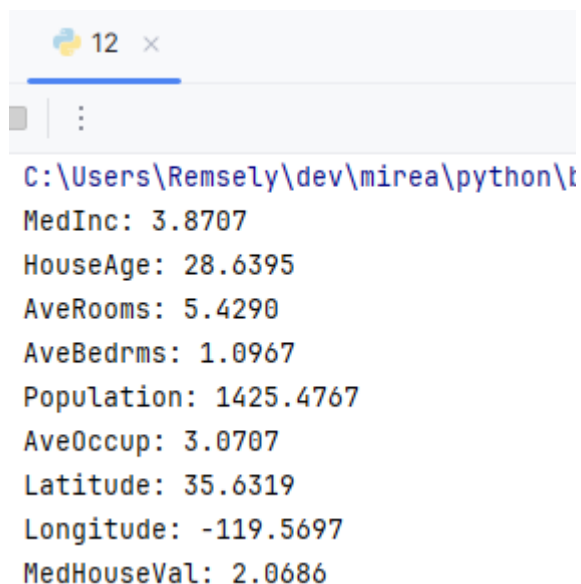
```
import pandas as pd
from sklearn.datasets import fetch_california_housing

housing = fetch_california_housing(as_frame=True)
df = housing.frame

mean_values = df.apply(pd.Series.mean)

for feature_name, mean_value in mean_values.items():
    print(f"{feature_name}: {mean_value:.4f}")
```

Рисунок 20 – Код программы



```
C:\Users\Remsely\dev\mirea\python\t
MedInc: 3.8707
HouseAge: 28.6395
AveRooms: 5.4290
AveBedrms: 1.0967
Population: 1425.4767
AveOccup: 3.0707
Latitude: 35.6319
Longitude: -119.5697
MedHouseVal: 2.0686
```

Рисунок 21 – Результат вывода

Задание 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. 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': '-...'}.

На входе: В одной строке вам дан текст, который состоит из латинских букв и пробелов.

На выходе: Выведите каждое слово исходного текста, закодированное азбукой Морзе. Количество строк в ответе должно совпадать с количеством слов в исходном тексте. Между закодированными буквами ставится ровно один пробел. Например, слово «Help» превратится в «.... . -.-. ---». Строчные и заглавные буквы кодируются одинаково. Например: Ignition sequence start

Перевод .. ---. -.-. ... -.-. ---. -.-.-. ---. -.-. -.-. -.-. -.-. -

```
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': '-...'
}

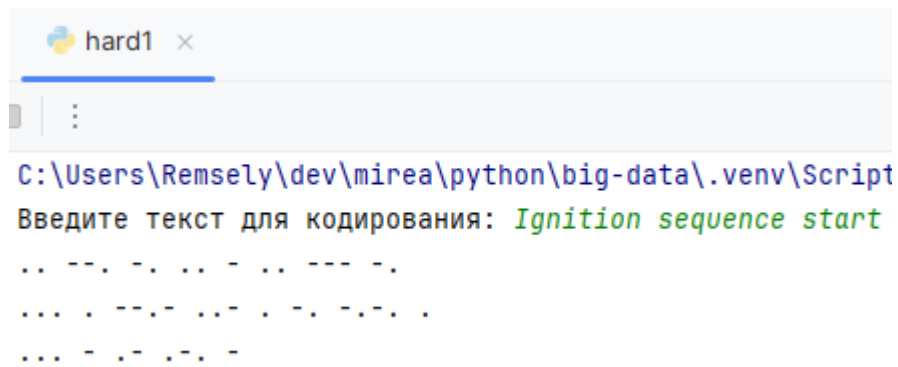
Windsurf: Refactor | Explain | Docstring | X
def encode_to_morse(): 1 usage new *
    input_text = input("Введите текст для кодирования: ")
    words = input_text.split()

    for word in words:
        encoded_word_parts = []
        for char in word:
            lower_char = char.lower()
            morse_code = morze.get(lower_char)
            if morse_code:
                encoded_word_parts.append(morse_code)

        encoded_word = " ".join(encoded_word_parts)
        print(encoded_word)

if __name__ == "__main__":
    encode_to_morse()
```

Рисунок 22 – Код программы



```
hard1 x
C:\Users\Remsely\dev\mirea\python\big-data\.venv\Script
Введите текст для кодирования: Ignition sequence start
... ..
... ..
... ..
```

Рисунок 23 – Пример вывода

Задание 2*

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.

Входные данные: в первой строке входных данных задано число n ($1 \leq n \leq 100000$). Следующие n строк содержат запросы к системе. Каждый запрос представляет собой непустую строку длиной не более 32 символов, состоящую только из строчных букв латинского алфавита.

Выходные данные: в выходных данных должно содержаться n строк – ответы системы на запросы: "ОК" в случае успешной регистрации, или подсказка с новым именем, если запрашиваемое уже занято.

```

def register_users(): 1 usage new *
    n = int(input())

    registered_users = set()

    for _ in range(n):
        try:
            name = input()
        except EOFError:
            break

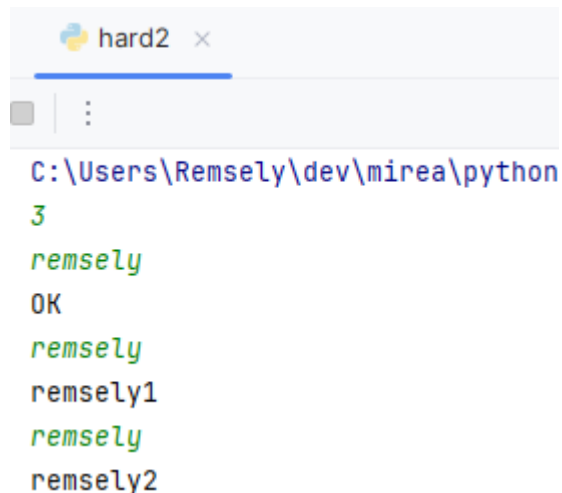
        if name not in registered_users:
            print("OK")
            registered_users.add(name)
        else:
            counter = 1
            while True:
                new_name = f"{name}{counter}"

                if new_name not in registered_users:
                    print(new_name)
                    registered_users.add(new_name)
                    break
                counter += 1

if __name__ == "__main__":
    register_users()

```

Рисунок 24 - Код программы



```

C:\Users\Remsely\dev\mirea\python
3
remse
OK
remse
remse1
remse
remse2

```

Рисунок 25 – Пример вывода

Задание 3*

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").

Входные данные: на вход программе подаются следующие параметры: число n – количество файлов в файловой системе. В следующих n строках содержится информация с именами файлов и допустимыми действиями (w, x, r), разделённых пробелами. Далее идёт число m – количество запросов к файлам вида «операция файл» (обозначение операции: "write", "read", "execute").

Выходные данные: для каждого допустимого запроса программа должна возвращать ОК, для недопустимого – Access denied.

```
def process_file_requests(): 1 usage new *
    operation_map = {
        "write": "w",
        "read": "r",
        "execute": "x"
    }

    file_permissions = {}

    num_files = int(input())

    for _ in range(num_files):
        parts = input().split()
        file_name = parts[0]
        allowed_operations = set(parts[1:])
        file_permissions[file_name] = allowed_operations

    num_requests = int(input())

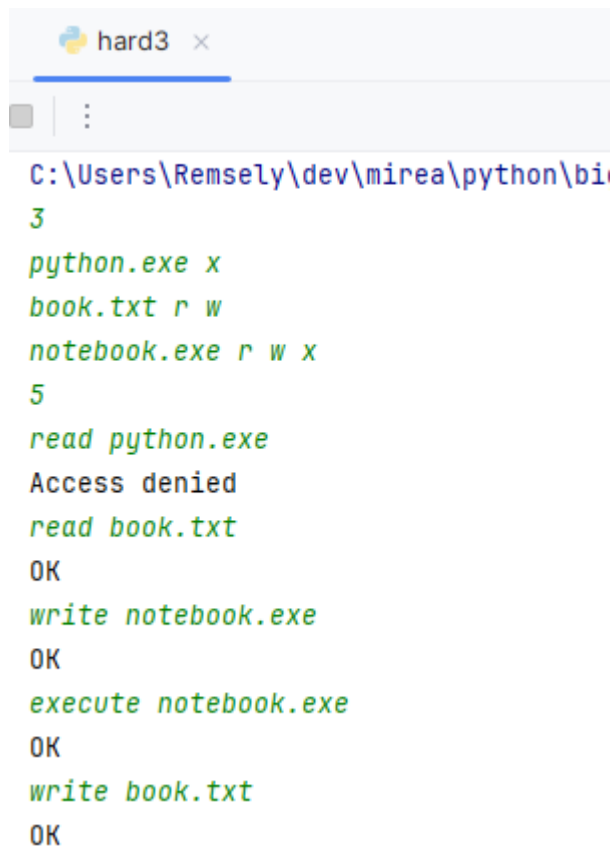
    for _ in range(num_requests):
        request_parts = input().split()
        operation_word = request_parts[0]
        file_name = request_parts[1]

        operation_code = operation_map.get(operation_word)

        if file_name in file_permissions and operation_code in file_permissions[file_name]:
            print("OK")
        else:
            print("Access denied")

if __name__ == "__main__":
    process_file_requests()
```

Рисунок 26 – Код программы



```
C:\Users\Remsely\dev\mirea\python\bi
3
python.exe x
book.txt r w
notebook.exe r w x
5
read python.exe
Access denied
read book.txt
OK
write notebook.exe
OK
execute notebook.exe
OK
write book.txt
OK
```

Рисунок 27 – Пример вывода

ВЫВОД

В ходе выполненной практической работы изучены основы языка Python, а также библиотеки numpy, упрощающей работу с данными.