

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ И. С. ТУРГЕНЕВА»

Кафедра программной инженерии

**ОТЧЕТ**

по лабораторной работе № 1  
на тему: «Основные типы данных. Управляющие конструкции»  
по дисциплине: «Программирование на языке Python»

Выполнил: Евдокимов Н.А.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверила: Захарова О.В.

Отметка о зачете:

Дата: « \_\_\_\_ » \_\_\_\_\_ 2019 г.

Орёл, 2019

## Задание

### Вариант 4

1. Написать программу, вычисляющую выражение:  $\sqrt[n]{y}(\sin y^2 + \cos^2 y) + \frac{x^n}{m}$ . Значения всех переменных задавать с клавиатуры. При задании неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элемента в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление последнего элемента списка;
- 4) сформировать кортеж, состоящий из строковых элементов списка; вывести содержимое кортежа на экран;
- 5) найти сумму всех целочисленных отрицательных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество букв в строке;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M2 и M1;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с нечетными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «R» (площадь прямоугольника), «T» (площадь треугольника), «E» (площадь эллипса), «Q» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать `lambda`.

## Код

1.

```
import math as m
```

```
x = None
while x is None:
    try:
        x = float(input('Введите x: '))
    except ValueError:
        x = None
```

```
y = None
while y is None:
    try:
        y = float(input('Введите y: '))
    except ValueError:
        y = None
```

```

n = None
while n is None:
    try:
        n = float(input('Введите n: '))
    except ValueError:
        n = None

m = None
while m is None or m == 0:
    try:
        m = float(input('Введите m: '))
    except ValueError:
        m = None

result = m.pow(y, 1 / n) * (m.sin(m.pow(y, 2)) + m.pow(m.cos(y), 2)) + m.pow(x, m.pow(n, m)) / m
print('Результат вычислений:', result)

```

2.

```

import textwrap as tw
import re

```

```

def print_menu():
    menu = """
    1) Вывести список на экран
    2) Добавить элемент в список
    3) Удалить элемент из списка
    4) Составить кортеж из четных элементов списка
    5) Найти сумму всех целочисленных элементов
    6) Сформировать строку и посчитать слова
    7) Задать 2 множества и вывести их объединение
    8) Сделать словарь и вывести на экран пары с ключом > 5
    0) Выйти
    """
    print(tw.dedent(menu))

```

```

def add_item(target_list):
    if type(target_list) != list:
        raise ValueError

    index = None
    while index is None:
        try:
            index = int(input('Введите индекс: '))
        except ValueError:
            index = None

    value = input('Введите значение: ')

    if '.' in value:
        # float
        try:
            typed_value = float(value)
        except ValueError:
            typed_value = value
    elif 'j' in value:
        # complex
        try:
            typed_value = complex(value)
        except ValueError:

```

```

        typed_value = value
    else:
        # int or string
        try:
            typed_value = int(value)
        except ValueError:
            typed_value = value

    target_list.insert(index, typed_value)

def remove_item(target_list):
    if type(target_list) != list:
        raise ValueError

    index = None
    while index is None:
        try:
            index = int(input('Введите индекс: '))
        except ValueError:
            index = None

    target_list.pop(index)

def make_tuple_of_even(src_list):
    if type(src_list) != list:
        raise ValueError

    list_of_even = src_list[::2]

    return tuple(list_of_even)

def sum_of_ints(src_list):
    if type(src_list) != list:
        raise ValueError

    ints_sum = 0
    for item in src_list:
        if type(item) == int:
            ints_sum += item

    return ints_sum

def word_count(src_list):
    if type(src_list) != list:
        raise ValueError

    result_str = ""
    for item in src_list:
        result_str += str(item)

    words = re.findall('[a-zA-Za-яА-Я]+', result_str)

    return len(words)

def make_set_union():
    set_str1 = input('Введите первое множество через запятую: ')
    set_str2 = input('Введите второе множество через запятую: ')

```

```
set1 = set(set_str1.split(','))
set2 = set(set_str2.split(','))
union = sorted(set1.union(set2))

print(union)
```

```
def make_dict(src_list):
    if type(src_list) != list:
        raise ValueError

    result_dict = { }
    for index, item in enumerate(src_list):
        if index <= 5:
            continue
        result_dict[index] = item

    print(result_dict)
```

```
def run():
    should_run = True
    item_list = [1, None, 2.3, "sda"]

    while should_run:
        print_menu()
        try:
            user_choice = int(input())
        except ValueError:
            user_choice = None

        if user_choice is None:
            print("Некорректный ввод")
            continue

        if user_choice == 0:
            should_run = False

        elif user_choice == 1:
            print(item_list)

        elif user_choice == 2:
            add_item(item_list)

        elif user_choice == 3:
            remove_item(item_list)

        elif user_choice == 4:
            print(make_tuple_of_even(item_list))

        elif user_choice == 5:
            print(sum_of_ints(item_list))

        elif user_choice == 6:
            print(word_count(item_list))

        elif user_choice == 7:
            make_set_union()

        elif user_choice == 8:
            make_dict(item_list)
```

```
if __name__ == '__main__':  
    run()
```

3.

```
import textwrap as tw  
import math as m
```

```
def print_menu():  
    menu = """  
        R) Площадь прямоугольника  
        T) Площадь прямоугольного треугольника  
        M) Площадь правильного многоугольника  
        E) Выйти  
    """  
    print(tw.dedent(menu))
```

```
def rectangle_area():  
    a, b = None, None  
  
    while a is None:  
        try:  
            a = float(input('Введите первую сторону: '))  
        except ValueError:  
            a = None  
  
    while b is None:  
        try:  
            b = float(input('Введите вторую сторону: '))  
        except ValueError:  
            b = None  
  
    area = a * b  
    print(area)
```

```
def right_triangle_area():  
    a, b = None, None  
  
    while a is None:  
        try:  
            a = float(input('Введите первую сторону: '))  
        except ValueError:  
            a = None  
  
    while b is None:  
        try:  
            b = float(input('Введите вторую сторону: '))  
        except ValueError:  
            b = None  
  
    area = a * b / 2  
    print(area)
```

```
def right_polygon_area():  
    n, a = None, None  
  
    while n is None:  
        try:  
            n = float(input('Введите количество сторон: '))
```

```

except ValueError:
    n = None

while a is None:
    try:
        a = float(input('Введите длину стороны: '))
    except ValueError:
        a = None

area = n / 4 * a**2 * (1 / m.tan(m.pi / n))
print(area)

def run():
    should_run = True
    while should_run:
        print_menu()

        user_choice = input().lower()
        if user_choice == 'r':
            rectangle_area()

        elif user_choice == 't':
            right_triangle_area()

        elif user_choice == 'm':
            right_polygon_area()

        elif user_choice == 'e':
            should_run = False
        else:
            print("Некорректный ввод")
            continue

if __name__ == '__main__':
    run()

```