

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО 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»
Вариант № 18

Выполнил: Шорин В.Д.

Шифр: 171406

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверили: Захарова О.В., Раков В.И.

Отметка о зачете:

Дата: «____» _____ 2019 г.

Орел, 2019 г.

Задание:

1. Написать программу, вычисляющую выражение: $\frac{x^n + p \cdot x + c}{b \cdot a \cdot \sin 5^\circ}$. 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 элементов в начало списка (добавлять элементы разных типов);
- 3) удаление первого элемента списка;
- 4) сформировать кортеж, состоящий из вещественных элементов списка больших 100; вывести содержимое кортежа на экран;
- 5) найти сумму всех целочисленных отрицательных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество цифр в строке;
- 7) задать с клавиатуры множество M_1 , сформировать множество M_2 из списка; вывести на экран разницу множеств M_2 и M_1 ;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с четными 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: «Е» (площадь эллипса), «Т» (площадь треугольника), «Р» (площадь прямоугольника), «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:

«main.py»

```
import task1import task2import task3
while True:
    print("Choose task:")
    print("1 - 1")  print("2 - 2")  print("3 - 3")  print("0 - exit")
    task_number = input("Your choose: ")  print()
    if task_number == '1':      task1.TaskOneMain()
    elif task_number == '2':    task2.TaskTwoMain()
    elif task_number == '3':    task3.TaskThreeMain()
    elif task_number == '0':    break
    else:      print(task_number + "_Undefined command")
```

«task1.py»

```
import math

def calculateExpression(a, b, c, n, x):
    res = 0.0

    try:
        res = ((5 * math.pow(a, n * x))
                / math.fabs(b + c)) \
                - math.sqrt(math.fabs(math.sin(math.pow(x, n * x))))
    except ZeroDivisionError:
        print("Incorrect values! Division by zero!")
    return res

def TaskOneMain():
    while True:
        while True:
            a = input("a = ")
            try:
                a = float(a)
            except ValueError:
                print("Its not float")
                continue
            break

        while True:
            b = input("b = ")
            try:
                b = float(b)
            except ValueError:
                print("Its not float")
```

```

        continue
    break

while True:
    c = input("c = ")
    try:
        c = float(c)
    except ValueError:
        print("Its not float")
        continue
    break

while True:
    n = input("n = ")
    try:
        n = float(n)
    except ValueError:
        print("Its not float")
        continue
    break

while True:
    x = input("x = ")
    try:
        x = float(x)
    except ValueError:
        print("Its not float")
        continue
    break
break
print("Result: " + str(calculateExpression(a, b, c, n, x)))
input()

```

«task2.py»

```

def PrintList(l: list):
    for v in l:
        print(v)

def AddElement(l: list):
    newValue = input("New value: ")
    try:
        l.insert(0, int(newValue))
        return True
    except ValueError:
        pass

```

```

try:
    l.insert(0, float(newValue))
    return True
except ValueError:
    pass

l.insert(0, newValue)
return True

def DeleteFirst(l: list):
    l.pop(0)

def Task4(l: list):
    t = []
    for v in l:
        if type(v) == float and v > 100.0:
            t.append(v)

    myTuple = tuple(t)
    print(myTuple)

def SumIntNegative(l: list):
    sum = 0
    for v in l:
        if type(v) == int and v < 0:
            sum += v
    print("Sum: " + str(sum))

def NumbersCount(l: list):
    s = str(l)
    count = [int(i) for i in s if i.isdigit()]

    print("Count: " + str(len(count)))

def CreateM(l: list):

    print("Enter set elements:")

    custom_set = set(map(str, input().split()))
    print(str(custom_set))
    list_set = set(map(str, l))
    print(list_set.difference(custom_set))

def CreateDictionary(l: list):

```

```

myDict = { }

for i, v in enumerate(l):
    myDict[i] = v
for i, v in myDict.items():
    print(v) if i % 2 == 0 else 0

def TaskTwoMain():
    custom_list = [-1, 153.5, "Python", -14, 17, -5, 147.4, "Hello"]

    while True:
        print("1 - Show list")
        print("2 - Add new element")
        print("3 - Delete first element")
        print("4 - Show element > 100")
        print("5 - Sum all int negative element")
        print("6 - Count of numbers in string of list")
        print("7 - Create M1")
        print("8 - Show dictionary")
        print("0 - Exit")
        command = input("Command:")

        if command == '1':
            PrintList(custom_list)
        elif command == '2':
            AddElement(custom_list)
        elif command == '3':
            DeleteFirst(custom_list)
        elif command == '4':
            Task4(custom_list)
        elif command == '5':
            SumIntNegative(custom_list)
        elif command == '6':
            NumbersCount(custom_list)
        elif command == '7':
            CreateM(custom_list)
        elif command == "8":
            CreateDictionary(custom_list)
        elif command == '0':
            return
        else:
            print("Incorrect command")

    «task3.py»

import math

```

```

def Ellipse():
    res = 0.0

    while True:
        a = input("Big semiaxis = ")
        try:
            a = float(a)
        except ValueError:
            print("Its not float")
            continue
        break

    while True:
        b = input("Little semiaxis = ")
        try:
            b = float(b)
        except ValueError:
            print("Its not float")
            continue
        break

    res = math.pi * a * b
    print("Res: " + str(res))
    input()
    pass

def Triangle():
    res = 0.0

    while True:
        a = input("Basis of triangle = ")
        try:
            a = float(a)
        except ValueError:
            print("Its not float")
            continue
        break

    while True:
        h = input("Height = ")
        try:
            h = float(h)
        except ValueError:
            print("Its not float")
            continue

```

```
break
```

```
res = 1/2 * a * h  
print("Res: " + str(res))  
input()
```

```
def Rectangle():  
    res = 0.0
```

```
    while True:  
        a = input("a = ")  
        try:  
            a = float(a)  
        except ValueError:  
            print("Its not float")  
            continue  
        break
```

```
    while True:  
        b = input("b = ")  
        try:  
            b = float(b)  
        except ValueError:  
            print("Its not float")  
            continue  
        break
```

```
    res = a * b  
    print("Area: " + str(res))  
    input()
```

```
def TaskThreeMain():  
    while True:  
        print("Ellipse - e")  
        print("Triangle - t")  
        print("Rectangle - r")  
        print("Quit - q")  
        command = input("Command:")  
  
        if command == 'e':  
            Ellipse()  
        elif command == 't':  
            Triangle()  
        elif command == 'r':  
            Rectangle()
```



```
elif command == 'q':  
    break  
else:  
    print("Undefined command...")
```