

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный лесотехнический университет
имени Г.Ф. Морозова»

Отчет

По лабораторным работам
По дисциплине «Командная работа в IT-проектах»

Выполнил студент
группы ИС2-212-ОБ
Тюнина А.М.
Преподаватель
Акименко А.В.

Воронеж 2023

ЛАБОРАТОРНАЯ(ПРАКТИЧЕСКАЯ) РАБОТА № 1

Тема: Основы синтаксиса. Математические операции в **Python**.

Цель: освоить основные правила синтаксиса языка **Python**, математические операции.

Задание №1

16	$T = \frac{\sin^2 2u}{\ln(2y+u)} + 0,24$
----	--

main.py +

```
1 import math
2 u=float(input("Введите переменную u:"))
3 y=float(input("Введите переменную y:"))
4 print((math.pow(math.sin(2*u),2)/(math.log10(2*y+u)))+0.24)
```

Ln: 4, Col: 22

Run

Share

Command Line Arguments

Введите переменную u:

3

Введите переменную y:

4

0.3149698185386157

** Process exited - Return Code: 0 **
Press Enter to exit terminal

Контрольные вопросы

1. Какая операция с целыми числами никогда не дает в виде результата целое число?
-ПРОСТОЕ ДЕЛЕНИЕ
2. Отличаются ли чем-то варианты объявления целого числа: 1230 или int(1230)?
- В ПЕРВОМ СЛУЧАЕ СТРОКОВЫЙ ТИП ДАННЫЙ, А ВО ВТОРОМ ЦЕЛОЧИСЛЕННЫЙ
3. Какой результат дает применение операции int(a) к любому числу «a» с плавающей точкой?
- ПРЕОБРАЗОВАНИЕ СТРОКИ К ЦЕЛОМУ ЧИСЛУ
4. Каким методом можно проверить, является ли заданное вещественное число целым?
- ОСТАТКОМ ОТ ДЕЛЕНИЯ (print(a//b)) – должно быть равно 0
5. Какой тип данных получится в результате вычисления выражения: 3 + 3.0 + 4?
- ЧИСЛО С ПЛАВАЮЩЕЙ ТОЧКОЙ

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА №2

Тема: условная инструкция if-elif-else.

Цель работы: познакомиться со структурой ветвление (if, if-else, if-elif-else), научиться работать с числами и строками используя данную структуру.

16	Дано натуральное число. Определить, является ли оно четным или оканчивающимся цифрой 7. Ответ вывести в виде сообщения.
----	---

main.py +

```
1 n=int(input("Введите n: "))
2 if n%2==0:
3     print("Число четное")
4 else:
5     print("Число нечетное")
6
7 if n%10==7:
8     print("Число оканчивается на 7")
9 else:
10    print("Число не оканчивается на 7")
```

Введите n:

7

Число нечетное

Число оканчивается на 7

** Process exited - Return Code: 0 **
Press Enter to exit terminal

Контрольные вопросы

1. Приведите пример минимально возможной конструкции условного выражения в Python. `if a==b: print('Числа равны')`
 2. Назовите все возможные типовые варианты написания условных инструкций.
- Для реализации выбора из нескольких альтернатив можно использовать конструкцию `if – elif – else`.

Конструкция `if – else`.

Конструкция `if`

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА № 3

Тема: работа с циклами.

Цель работы: познакомиться с циклическими конструкциями.

16 Вывести квадраты чисел от 11 до 99.

```
main.py def squared(): for i in range(11,100): print(i*i,end="\t") squared()
```

Ln: 4, Col: 10

Run Share Command Line Arguments

121	144	169	196	225	256	289	324	361	400	441	484	529	576	625	676	729	784
841	900	961	1024	1089	1156	1225	1296	1369	1444	1521	1600	1681	1764	1849	1936	2025	2116
2209	2304	2401	2500	2601	2704	2809	2916	3025	3136	3249	3364	3481	3600	3721	3844	3969	4096
4225	4356	4489	4624	4761	4900	5041	5184	5329	5476	5625	5776	5929	6084	6241	6400	6561	6724
6889	7056	7225	7396	7569	7744	7921	8100	8281	8464	8649	8836	9025	9216	9409	9604	9801	

** Process exited - Return Code: 0 **
Press Enter to exit terminal

Контрольные вопросы

В чем основное отличие циклов `for` и `while`?

`while` в **Python** повторяет указанный блок кода до тех пор, пока указанное в цикле логическое выражение будет оставаться истинным.

`for` в **Python** обладает способностью перебирать элементы любого комплексного типа данных (например, строки или списка).

Приведите пример бесконечного цикла. Применяется ли он на практике?

`a = 1`

`while a==1:`

`b = input('Как тебя зовут?')`

`print('Привет', b, ', Добро пожаловать')`

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА № 4

Тема: работа со строками.

Цель работы: познакомиться с методами работы со строками.

16 Ввести массив, состоящий из 11 элементов. Определить количество нечетных элементов массива, больших по значению 8.

```
main.py 1 mass=[] 2 i=0 3 y=0 4 n=11 5 for i in range(i,n): 6     mass.append(int(input("Введите элемент массива: "))) 7     if mass[i]>8: 8         if mass[i]%2==0: 9             y+=1 10 print("Количество элементов удовлетворяющих условию: ", y)
```

input

Введите элемент массива: 12
Введите элемент массива: 3
Введите элемент массива: 12
Введите элемент массива: 45
Введите элемент массива: 23
Введите элемент массива: 34
Введите элемент массива: 21
Введите элемент массива: 22
Введите элемент массива: 34
Введите элемент массива: 2
Введите элемент массива: 3
Количество элементов удовлетворяющих условию: 5

Контрольные вопросы

1. Строки в python обозначаются кавычками. Приведите все способы. (`'...'`), (`"..."`), (`"""..."""`)
2. Как применяют операции сложения и умножения к строкам?

Операция сложения или конкатенирования складывает две или более строк в одну. В качестве аргументов могут выступать только строки.

Пример - Интерактивный режим

```
>>> 'Hello ' + 'Py'
```

```
'Hello Py'
```

```
>>> 'Hello ' + 4
```

```
TypeError: can only concatenate str (not "int") to str
```

Операция умножение, применяемая к строкам, подразумевает ее дублирование определенное количество раз (в соответствии со значением числа, которое передано в качестве параметра. Если передать отрицательное число или ноль, то вернется пустая строка).

Пример – Интерактивный режим

```
>>> 'Hello ' * 4
```

```
'Hello Hello Hello Hello '
```

```
>>> 'Hello ' * -4
```

3. Какие типы данных можно преобразовать в строку?

Типы int, float и другие числовые типы можно преобразовать в строку, а наоборот нельзя (если строка состоит только из символов алфавита).

4. Перечислите характеристики типа данных «список», которые вы знаете.

Свойств много. Вот некоторые: 1) Списки изменяются динамически (вы можете создать пустой список, потом добавить в него ряд элементов, затем удалить часть из них – и все это будет осуществляться в одном контейнере, относиться к одной и той же переменной); 2) К элементам списка можно получить доступ по индексу (начиная с нулевого). Индексация бывает и обратной, отрицательной (индекс -1 обозначает последний элемент списка, -2 – предпоследний и т.д.); 3) В списках может содержаться ряд элементов любых типов (числа, строки, другие списки ...); 4) У списков имеется большое разнообразие методов, позволяющих осуществлять операции с ними (расширять, удалять элементы, очищать, сортировать); 5) Списки можно «резать», т.е. формировать более мелкие последовательности на основании имеющейся; 6) Значения элементов могут повторяться

5. Как проверить наличие элемента в списке?

Есть 2 очевидных способа:

При помощи конструкции in:

Пример – IDE `lst = [1, 2, 3, 14, 33, 1, 1]`

```
if 1 in lst: ____print('Есть')
```

Результат: Есть

При помощи метода count:

Пример – IDE

```
lst = [1, 2, 3, 14, 33, 1, 1]
```

```
if lst.count(5): ____print('Есть')
```

```
else: ____print('Нет')
```

Результат: Нет

6. Чем отличаются методы append() и extend()?

Метод append() добавляет в конец текущего списка новый элемент. Метод extend() добавляет в конец текущего списка новые элементы в распакованном виде.

7. Какие параметры можно передавать при срезах списков?

Для срезов можно пользоваться функцией slice() или специальным сокращением, куда входит 3 параметра: начало среза (по умолчанию – первый элемент), конец среза (конечный член списка, не включая его), шаг (по умолчанию – 1, т.е. выбираем все элементы без пропусков).

Пример – IDE `h = [1, 2, 3, 14, 33, 1, 9]`

```
print(h[slice(2, 6, 2)])
```

Результат: [3, 33]

Т.е. создаем новый список начиная с элемента с индексом 2 (в нашем случае это цифра 3) вплоть до 6 элемента (не включая его) с шагом 2 (пропускаем каждое второе значение). Часть или все параметры можно опускать в специальных сокращениях.

Пример – IDE `h = [1, 2, 3, 14, 33, 1, 9]`

`print(h[2:6:2]) print(h[::2])`

Результат: [3, 33] [1, 3, 33, 9]

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ) РАБОТА № 5

Тема: функции и процедуры в Python.

Цель работы: изучение процедур и функций в Python.

16	Даны два правильных треугольника со сторонами a_1, a_2 . Найти отношение $x = S_1/S_2$, где S_1, S_2 – площади треугольников, определяемые по формуле $S = \frac{a}{4}\sqrt{3}$.
----	--

```
main.py +
1 import math
2 def triangle(a):
3     S=(a/4)*math.sqrt(3)
4     return S
5 a1=int(input("Введите сторону 1 треугольника: "))
6 S1=triangle(a1)
7 print("Площадь 1 треугольника: ",S1 )
8 a2=int(input("Введите сторону 2 треугольника: "))
9 S2=triangle(a2)
10 print("Площадь 1 треугольника: ",S2)
11 x=S1/S2
12 print("Отношение S1 k S2= ",x)

Ln: 12, Col: 1
Run Share Command Line Arguments

Введите сторону 1 треугольника:
4
Площадь 1 треугольника:  1.7320508075688772
Введите сторону 2 треугольника:
6
Площадь 1 треугольника:  2.598076211353316
Отношение S1 k S2=  0.6666666666666666

** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

Контрольные вопросы

1. В чем основное различие между глобальными и локальными переменными?
- с глобальными переменными можно взаимодействовать из любого положения (места) программы, а локальные переменные работают только в пределах функции
2. Какая инструкция позволяет возвращать значения?
`return b`
3. Назовите ключевое слово, которое определяет функцию
`def`
4. Какое количество аргументов может принимать функция?
- Функции в Python могут иметь максимально 256 аргументов.