­­­Министерство образования Республики Беларусь

Белорусский Национальный Технический Университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение вычислительной техники

и автоматизированных систем»

**Отчёт**

по лабораторной работе № 4

по дисциплине ***«Языки программирования»***

тема:«Базовый синтаксис языка Python: условные конструкции (ветвления)»

**Вариант 9**

|  |  |  |
| --- | --- | --- |
| Исполнитель: |  | студент группы 10701215  Воробей Иван Александрович |
| Преподаватель: |  | Белова С.В. |

2018-2019 учебный год

**ЛАБОРАТОРНАЯ РАБОТА № 4**

**БАЗОВЫЙ СИНТАКСИС ЯЗЫКА PYTHON: УСЛОВНЫЕ КОНСТРУКЦИИ (ВЕТВЛЕНИЯ)**

Цель работы:

Изучить синтаксис условных конструкций языка Python для программирования разветвляющихся алгоритмов и закрепить их на примере разработке интерактивных приложений.

Основное задание:

Разработать интерактивную программу «Quadric Equation» («Квадратное уравнение») для решения квадратных уравнений вида: . Программа должна запрашивать у пользователя соответствующие параметры a, b и с и, в зависимости от вычисленного дискриминанта *D*, выдавать соответствующий результат. В случае отрицательного дискриминанта программа должна выводить сообщение о том, что действительных корней нет.

Индивидуальное задание:

Задание 1. В соответствии с заданием своего варианта составить программу для вычисления значений составной функции G(x). Параметр a вводится с клавиатуры. Функция f(x) может принимать одно из двух значений (ex или x2) по выбору пользователя. Выполнить расчет для всех вариантов ветвления.

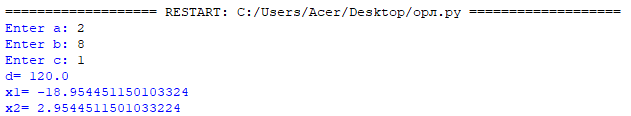
Задание 2. В соответствии с заданием своего варианта составить программу, которая распределяет целые числа, вводимые пользователем с клавиатуры по четырем группам. Результат представить в виде: число 49 относится к группам А и B.

Дополнительное задание\*

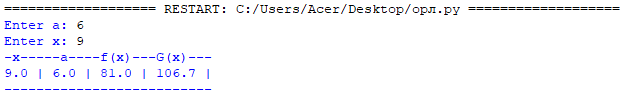
1. Напишите программу, которая бы эмулировала игру «*Dice*» (игра в кости). Суть игры заключается в броске двух шестигранных кубиков (костей) и подсчёта общей суммы очков, которые выпали на первой и второй костей. Для генерирования случайного значения на костях воспользуйтесь функциями из стандартного модуля *random*.
2. Напишите программу – симулятор пирожков с «сюрпризом». Программа должна выводить пирожок и один из пяти (можно больше) различных «сюрпризов», который бы выбирался случайным образом.
3. Напишите программу «*MoodSensor*» (эмулировать датчика настроения), которая «залазит» в душу пользователя и определяет его настроение в текущий момент времени. Приложение будет генерировать случайное число, в зависимости от значения которого на экран выводится одно из псевдографических «лиц», которое и будет отображать настроение пользователя.

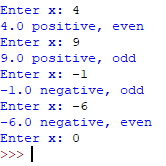
Требования к выполнению:

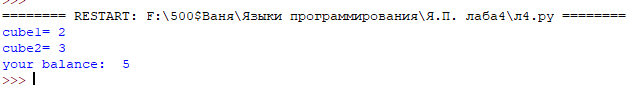
1. Программа должна обязательно быть снабжена комментариями на английском языке, в которых необходимо указать краткое предназначение програм-мы, номер лабораторной работы и её название, версию программы, ФИО разработчика, номер группы и дату разработки.
2. Каждая программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом (да-да, пусть даже пока в консольном варианте).

Результаты выполнения основного задания:

Результаты выполнения индивидуального задания:



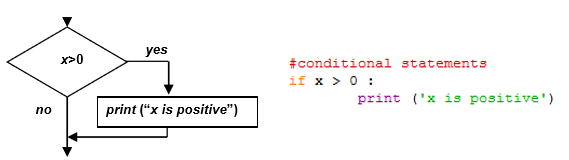
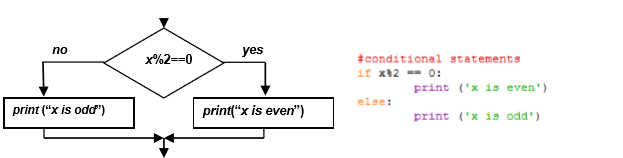
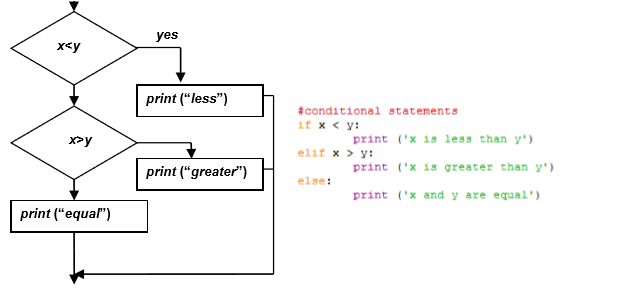


****Результаты выполнения дополнительного задания:



****

Ответы на контрольные вопросы:

1. Перечислите основные фундаментальные конструкции, которые используются при построении любой компьютерной программы.(ifelifelse)
2. Зачем нужно в начале разрабатывать алгоритм решения задачи и проектировать будущее приложение, а затем лишь преступать к непосредственному написанию кода программы?(чтобы повысить свой кпд)
3. Опишите основные элементы блок-схемы для графическо-го предоставления алгоритма решения задачи?(начало, блок данных,блок действия,блок вызова процедур или функций, блок условия, блок вывода, блок соеденитель на странице, блок соеденитель меж страницами)
4. Как в языке Python реализуется механизм истинности-ложности? Может ли само значение выступать в качестве условия? Какое при этом используется правило?(тип bool, 1 это правда, 0 это ложь)
5. Опишите синтаксис простой условной конструкции *if*, кото-рая используется для исполнения кода при определённом условии. Представьте примерную блок-схему конструкции.
6. С помощью каких операторов можно комбинировать в одной условной конструкции *if* несколько условий чтобы получить один результат? Кратенько опишите принцип их действия. Какой механизм оптимизации применяет интерпретатор Python для эффективного вычисления результата комбинированных условных выражений?(elifelse)
7. Опишите синтаксис условной конструкции *if-else*, которая используется для выбора вариантов продолжения выполнения кода при определённом условии. Представьте примерную блок-схему конструкции.
8. Опишите синтаксис условной конструкции *elif*, которая используется для множественного выбора исполнения кода на основе нескольких условий. Представьте примерную блок-схему конструкции.
9. Чем использование условной конструкции множественного выбора *elif* будет отличаться от использования вложенных условных конструкций *if-else* в случае использования их при решении одного и того же алгоритма?(if…if…if…if…else) .(if-elif-elif…elif-else)
10. Как сгенерировать случайную последовательность чисел с использованием функций *randint(…)* и *randrange(…)*? Чем они отличаются? Какие ещё есть полезные для вас функции в модуле *random*?(randint(a,b) генерирует яисло с промежутка[a,b]. Randrange(начало, конец, шаг)

Что я освоил в процессе выполнения лабораторной работы (выводы):

1. Научился использывать оператор if.
2. Научился генерировать случайные числа.
3. Научился програмировать разветвляющиеся алгоритмы

Листинг исходных кодов программ

**Основное задание**

import math #programm solve quadratic equation

a=float(input('Enter a: '))

b=float(input('Enter b: '))

c=float(input('Enter c: '))

d=2\*b\*b-4\*a\*c

print("d=",d)

if d<0 :

print("no solution")

elif d==0:

x1=(-b)/2\*a

print("x=",x1)

else:

x1=(-b-math.sqrt(d))/2\*a

x2=(-b+math.sqrt(d))/2\*a

print("x1=",x1,"\nx2=",x2)

**Дополнительныезадания**

**1)**

import random

x1=random.randint(1, 6) #In x1 write random value from 1 to 6

x2=random.randint(1, 6)

print("cube1=",x1)

print("cube2=",x2)

print("your balance: ",x1+x2)

**2)**

import random

x=random.randint(1, 5)

if x==1:

print("(000) have a good day!")

if x==2:

print("(000) Be happy!")

if x==3:

print("(000) Love is...!")

if x==4:

print("(000) Good boy")

if x==5:

print("(000) Good girl")

**3)**

import random #module conection

x=random.randint(1, 3)

if x==1:

print("=)")

if x==2:

print("=(")

if x==3:

print("=|")

**Индивидуальноезадание**

**1)**

import math

a=float(input('Enter a: '))

x=float(input('Enter x: '))

g=float

f=float

f=x\*x

if x>math.fabs(a):

g=2\*math.sin(x\*x)+3\*a\*a

g = float('{:.1f}'.format(g))

print("-x-----a----f(x)---G(x)---")

print(x,"|",a,"|",f,"|",g,"|")

print("--------------------------")

elif x<math.fabs(a):

g=math.fabs(x\*x-a)

g = float('{:.1f}'.format(g))

print("-x-----a----f(x)---G(x)---")

print(x,"|",a,"|",f,"|",g,"|")

print("--------------------------")

elif x==math.fabs(a):

g=math.sqrt(math.fabs(a\*a+x\*x\*x\*x))

g = float('{:.1f}'.format(g))

print("-x-----a----f(x)---G(x)---")

print(x,"|",a,"|",f,"|",g,"|")

print("--------------------------")

**2)**

import math

while True:

x=float(input('Enter x: '))

if x>0:

if x//2==x/2:

print(x, "positive, even" )

else:

print(x, "positive, odd")

if x<0:

if x//2!=x/2:

print(x, "negative, odd")

else:

print(x, "negative, even" )

if x==0:

break