**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Програмування**

**Лабораторна робота №2**

**«Типи даних, змінні та оператори мови програмування Python»**

Виконав:

студент групи ІО-63

Братун А.Ю.

Номер у списку групи: 5

Перевірив Новотарський М. А.

Київ 2016 р.

**Тема:** «Типи даних, змінні та оператори мови програмування Python».

**Мета:** вивчити типи даних, які використовуються в мові програмування Python. Змінні та правила їх іменування, операції над змінними. Оператори та їх застосування.

**Завдання:**

1. Вивчити матеріал лекцій 3, 4, 5 та 6.

2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

**Теоретичні основи що були використані при написанні лабораторної роботи:**

**Основні типи даних**

int – цілі числа.

float – дійсні числа.

**Змінні:**

x= 23 – присвоєння значення

x= int(input("x = "))– ввід значення

**Математичні оператори:**

+ – додавання.

- – віднімання.

\* – множення.

/ – ділення.

\*\* – піднесення до степеня.

**Оператори присвоювання:**

= - присвоює змінній значення.

+ = - збільшує значення.

**Оператори порівняння:**

== - дорівнює.

!= - не дорівнює.

< - менше.

> - більше.

<= - менше або дорівнює.

>= - більше або дорівнює.

**Оператори розгалуження й цикли:**

**Оператор розгалуження if ... else**

If <Логічний вираз>:

<Блок, виконуваний, якщо умова дійсна>

[elif <Логічний вираз>:

< Блок, виконуваний, якщо умова дійсна >

]

[else:

< Блок, виконуваний, якщо всі умови неправильні >

]

**Оператор циклу** while

< Початкове значення >

While < Умова >:

< Інструкції> >

< Збільшення >

[else:

< Блок, виконуваний, якщо не використовувався оператор break >

]

**Числа:**

int – цілі числа.

float – дійсні числа;

**Функції для роботи з числами:**

float( [<число або рядок >] )

**Модуль math. Математичні функції:**

Import math as m

е – повертає значення константи e:

sin(), cos(), tan( ) – стандартні тригонометричні функції

log( [<число >,< База > ] ) – логарифм по заданій базі.

fmod()– залишок від ділення.

**Поставлені завдання:**

**Завдання 1**

C:\Users\Андрей\Pictures\Screenshots\Снимок экрана (143).png

**Алгоритм розв’язання до завдання 1**:

* Імпортуємо модуль math.
* Вітаємось та забезпечуємо ввід даних
* Типізуємо дані та робимо виключення помилок
* Виконуємо обчислення, в потрібних місцях використовуємо модуль math.
* Вираховуємо ОДЗ до логарифма
* Перевіряємо чи введені значення задовольняють ОДЗ
* Якщо не задовольняють, відповідаємо користувачеві що число логарифма має бути більше від нуля
* Якщо введені значення задовольняють ОДЗ логарифма, обраховуємо значення знаменника
* Якщо знаменник дорівнює нулю виводимо надпис що на нуль ділити не можна
* Якщо знаменник відмінний від нуля і ОДЗ справидливе, форматуємо та виводимо відповідь
* Просимо користувача використати програму ще раз, запустивши всю програму в цикл

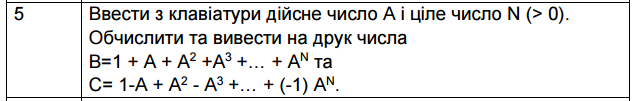
**Завдання 2**

**C:\Users\Андрей\Pictures\Screenshots\Снимок экрана (144).png**

**Алгоритм розв’язання до завдання 2**:

* Вітаємось та просимо ввести дані
* Типізуємо дані та робимо виключення помилок
* Перевіряємо чи **перше** число більше рівне нуля
* Якщо більше то підносимо до квадрату і виводимо відповідь
* Якщо менше від нуля, виводимо значення цього числа
* Перевіряємо чи **друге** число більше рівне нуля
* Якщо більше то підносимо до квадрату і виводимо відповідь
* Якщо менше від нуля, виводимо значення цього числа
* Перевіряємо чи **третє** число більше рівне нуля
* Якщо більше то підносимо до квадрату і виводимо відповідь
* Якщо менше від нуля, виводимо значення цього числа
* Форматуємо та виводимо відповідь
* Просимо користувача використати програму ще раз
* Запускаємо всю програму в цикл

**Завдання 3**

****

* Імпортуємо модуль math.
* Вітаємося та забезпечуємо ввід даних.
* Типізуємо введені дані та робимо виключення помилок
* Заповнюємо пусті комірки з відповідями
* Перевіряємо чи введене число N більше від нуля
* Створюємо цикл з кроком 1 для визначення значень функцій за введенними даними.
* Забезпечуємо перевірку значення аргументу циклу на парність. Відповідно до результату виконуєму відповідну дію.
* Після закінчення циклу форматуємо та виводимо відповідь

**Завдання 1**

**import** math **as** m  
**while True**:  
 print("Вітаю вас в програмі для обрахування виразу")  
 print("Увага!!! Всі невірно введені символи будуть прирівнюватись 0")  
 print("Введіть будь ласка число x")  
 **try**:  
 x = float(input())  
 **except** (ValueError, TypeError):  
 x = float(0)  
 print("Введіть будь ласка число y")  
 **try**:  
 y = float(input())  
 **except** (ValueError, TypeError):  
 y = float(0)  
 print("Введіть будь ласка число z")  
 **try**:  
 z = float(input())  
 **except** (ValueError, TypeError):  
 z = float(0)  
 doo1 = 7.8\*x\*\*2+3.52\*y  
 odz = x+2\*z  
  
 **try**:  
  
 **if** odz <= 0:  
 print("Число логарифма має бути більшим від нуля")  
 **elif** m.log(x+2\*z, m.e)+m.e\*\*y == 0:  
 print("В школі казали на нуль ділити не можна")  
 **else**:  
 a = doo1/(m.log(x+2\*z , m.e)+m.e\*\*y)  
 print("Відповідь: %.2f" % a)  
  
 **except** (OverflowError):  
 print ("Занадто великі числа")  
  
 print("Давай рахувати ще!!!!!!!")  
 print("Якщо хочете продовжити рахувати нажмыть Enter , якщо хочете закінчить введіть q")  
 a = input()  
 **if** a == "q":  
 print("З вами було приємно працювати")  
 **break**

**Завдання 2**

print("Вітаю в програмі для обчислення квадратів додатних чисел")  
**while True**:  
 print("Увага!!! Всі невірно введені символи будуть прирівнюватись 0")  
 print("Введіть перше число")  
 **try**:  
 a = float(input())  
 **except** (ValueError, TypeError):  
 a = float(0)  
 print("Введіть друге число")  
 **try**:  
 b = float(input())  
 **except** (ValueError, TypeError):  
 b = float(0)  
 print("Введіть третє число")  
 **try**:  
 c = float(input())  
 **except** (ValueError, TypeError):  
 c = float(0)  
 **if** a >= 0:  
 aq = a\*a  
 print("Квадрат першого числа: %.2f" % aq)  
 **else**:  
 print("Число %2.f" % a , "від'ємне")  
 **if** b >= 0:  
 bq = b\*b  
 print("Квадрат другого числа: %.2f" % bq)  
 **else**:  
 print("Число %2.f" % b , "від'ємне")  
 **if** c >= 0:  
 cq = c\*c  
 print("Квадррат третього числа: %.2f" % cq)  
 **else**:  
 print("Число %2.f" % c , "від'ємне")  
 print("Нумооо, я хочу більше рахувати!!!!!!")  
 print("Якщо хочете продовжити рахувати нажмыть Enter , якщо хочете закінчить введіть q")  
 a = input()  
 **if** a == "q":  
 print("З вами було приємно працювати")  
 **break**

**Завдання 3\_1**

**while True**:  
 print("Вітаю тебе в програмі для обчислення суми n членів геометричної прогресії перший член якої = 1")  
 print("Увага!!! Всі невірно введені символи будуть прирівнюватись 0")  
 print("Введіть знаменник геометричної прогресії")  
 **try**:  
 znam = float(input())  
 **except** (ValueError, TypeError):  
 znam = float(0)  
 print("Введіть кількість члевнів прогресії")  
 **try**:  
 kilkist = int(input())  
 **except** (ValueError, TypeError):  
 kilkist = int(0)  
 **if** kilkist < 0:  
 print("Кількість членів може бути від\*ємною?, я такого не знав")  
 **elif** kilkist == 0:  
 print("При знаменику рівному нулю відповідь буде рівна першому члену тобто 1")  
 **else**:  
 **try**:  
 doo1 = ((1-znam\*\*kilkist)/(1-znam))\*znam+1  
 **except** (OverflowError):  
 print("Занадто велика послідовність, дуже тяжко порахувати")  
 **break** print("Відповідь: %.2f" % doo1)  
 print("Ну може ще одненьку прогресію?")  
 print("Якщо хочете продовжити рахувати нажмыть Enter , якщо хочете закінчить введіть q")  
 a = input()  
 **if** a == "q":  
 print("З вами було приємно працювати")  
 **break**

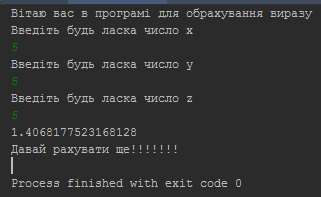
**Завдання 3\_2**

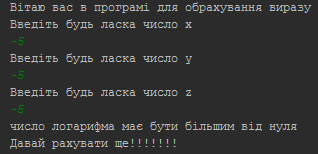
**import** math **as** m  
#Спочатку задаємо змінні на ввід  
**while True**:  
 print("Вітаю тебе в програмі для обчислення суми n членів деякої послідовності")  
 print("Увага!!! Всі невірно введені символи будуть прирівнюватись 0")  
 print("Введіть число яке буде показувати в скільки разів наступний член більший за попередній")  
 **try**:  
 pokaznik = float(input())  
 **except** (ValueError, TypeError):  
 pokaznik = float(0)  
 print("Введіть кількість члевнів постідовності")  
 **try**:  
 kilkist = int(input())  
 **except** (ValueError, TypeError):  
 kilkist = int(0)  
 #заповнимо далі комірки з дійсними числами  
 resultB = 1  
 resultC = 1  
 i = int(1)  
 #Використовуємо цикл  
 **if** kilkist > 0:  
 **while** i <= kilkist:  
 **try**:  
 resultB = resultB+pokaznik\*\*i  
 **except** (OverflowError):  
 print('Занадто велика послідовність, дуже тяжко порахувати')  
 **break** CheckPar = m.fmod(i,2)  
 **if** CheckPar != 0:  
 **try**:  
 resultC = resultC-pokaznik\*\*i  
 **except** (OverflowError):  
 print('Занадто велика послідовність, дуже тяжко порахувати')  
 **else**:  
 **try**:  
 resultC = resultC+pokaznik\*\*i  
 **except** (OverflowError):  
 print('Занадто велика послідовність, дуже тяжко порахувати')  
 i += 1  
 **else**:  
 print('Ви ввели неправильне число N')  
 print("B = %.2f" % resultB)  
 print("C = %.2f" % resultC)  
 print("Я хочу більше послідовнеостей!!!!!")  
 print("Якщо хочете продовжити рахувати нажмыть Enter , якщо хочете закінчить введіть q")  
 a = input()  
 **if** a == "q":  
 print("З вами було приємно працювати")  
 **break**

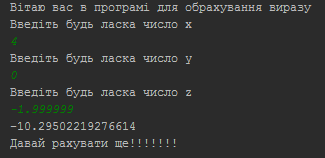
**Скомпонована програма**

**import** math **as** m  
**while True**:  
 print("Вітаю в програмі для здачі лабараторнгої роботи №2")  
 print("""Яке завдання будемо вирішувати?  
 1 - Обчислення виразу  
 2 - Піднесення до квадрату додатніх чисел  
 3.1 - Обчислення 1 послідовності через формулу суми n членів геометричної прогресії  
 3.2 - Обчислення послідовностей""")  
 quest = input("Введіть відповідне число: ")  
########################################################################################  
# Початок завдання 1  
########################################################################################  
 **if** quest == "1":  
 print("Вітаю вас в програмі для обрахування виразу")  
 print("Увага!!! Всі невірно введені символи будуть прирівнюватись 0")  
 print("Введіть будь ласка число x")  
 **try**:  
 x = float(input())  
 **except** (ValueError, TypeError):  
 x = float(0)  
 print("Введіть будь ласка число y")  
 **try**:  
 y = float(input())  
 **except** (ValueError, TypeError):  
 y = float(0)  
 print("Введіть будь ласка число z")  
 **try**:  
 z = float(input())  
 **except** (ValueError, TypeError):  
 z = float(0)  
 doo1 = 7.8\*x\*\*2+3.52\*y  
 odz = x+2\*z  
  
 **try**:  
  
 **if** odz <= 0:  
 print("число логарифма має бути більшим від нуля")  
 **elif** m.log(x+2\*z, m.e)+m.e\*\*y == 0:  
 print("В школі казали на нуль ділити не можна")  
 **else**:  
 a = doo1/(m.log(x+2\*z , m.e)+m.e\*\*y)  
 print("Відповідь: %.2f" % a)  
  
 **except** (OverflowError):  
 print ("Занадто великі числа")  
  
 print("Давай рахувати ще!!!!!!!")  
########################################################################################  
# Початок завдання 2  
########################################################################################  
 **elif** quest == "2":  
 print("Вітаю в програмі для обчислення квадратів додатних чисел")  
 print("Увага!!! Всі невірно введені символи будуть прирівнюватись 0")  
 print("Введіть перше число")  
 **try**:  
 a = float(input())  
 **except** (ValueError, TypeError):  
 a = float(0)  
 print("Введіть друге число")  
 **try**:  
 b = float(input())  
 **except** (ValueError, TypeError):  
 b = float(0)  
 print("Введіть третє число")  
 **try**:  
 c = float(input())  
 **except** (ValueError, TypeError):  
 c = float(0)  
 **if** a >= 0:  
 aq = a\*a  
 print("Квадрат першого числа: %.2f" % aq)  
 **else**:  
 print("Число %2.f" % a , "від'ємне")  
 **if** b >= 0:  
 bq = b\*b  
 print("Квадрат другого числа: %.2f" % bq)  
 **else**:  
 print("Число %2.f" % b , "від'ємне")  
 **if** c >= 0:  
 cq = c\*c  
 print("Квадррат третього числа: %.2f" % cq)  
 **else**:  
 print("Число %2.f" % c , "від'ємне")  
 print("Нумооо, я хочу більше рахувати!!!!!!")  
########################################################################################  
# Початок завдання 3.1  
########################################################################################  
 **elif** quest == "3.1":  
 print("Вітаю тебе в програмі для обчислення суми n членів геометричної прогресії перший член якої = 1")  
 print("Увага!!! Всі невірно введені символи будуть прирівнюватись 0")  
 print("Введіть знаменник геометричної прогресії")  
 **try**:  
 znam = float(input())  
 **except** (ValueError, TypeError):  
 znam = float(0)  
 print("Введіть кількість члевнів прогресії")  
 **try**:  
 kilkist = int(input())  
 **except** (ValueError, TypeError):  
 kilkist = int(0)  
 **if** kilkist < 0:  
 print("Кількість членів може бути від\*ємною?, я такого не знав")  
 **elif** kilkist == 0:  
 print("При знаменику рівному нулю відповідь буде рівна першому члену тобто 1")  
 **else**:  
 **try**:  
 doo1 = ((1-znam\*\*kilkist)/(1-znam))\*znam+1  
 **except** (OverflowError):  
 print("Занадто велика послідовність, дуже тяжко порахувати")  
 **break** print("Відповідь: %.2f" % doo1)  
 print("Ну може ще одненьку прогресію?")  
########################################################################################  
# Початок завдання 3.2  
########################################################################################  
 **elif** quest == "3.2":  
 #Спочатку задаємо змінні на ввід  
 print("Вітаю тебе в програмі для обчислення суми n членів деякої послідовності")  
 print("Увага!!! Всі невірно введені символи будуть прирівнюватись 0")  
 print("Введіть число яке буде показувати в скільки разів наступний член більший за попередній")  
 **try**:  
 pokaznik = float(input())  
 **except** (ValueError, TypeError):  
 pokaznik = float(0)  
 print("Введіть кількість члевнів постідовності")  
 **try**:  
 kilkist = int(input())  
 **except** (ValueError, TypeError):  
 kilkist = int(0)  
#заповнимо далі комірки з дійсними числами  
 resultB = 1  
 resultC = 1  
 i = int(1)  
 #Перевіряємо на парність  
 CheckPar=float(0)  
 #Використовуємо цикл  
 **if** kilkist > 0:  
 **while** i <= kilkist:  
 **try**:  
 resultB = resultB+pokaznik\*\*i  
 **except** (OverflowError):  
 print('Занадто велика послідовність, дуже тяжко порахувати')  
 **break** CheckPar = m.fmod(i,2)  
 **if** CheckPar != 0:  
 **try**:  
 resultC = resultC-pokaznik\*\*i  
 **except** (OverflowError):  
 print('Занадто велика послідовність, дуже тяжко порахувати')  
 **else**:  
 **try**:  
 resultC = resultC+pokaznik\*\*i  
 **except** (OverflowError):  
 print('Занадто велика послідовність, дуже тяжко порахувати')  
 i += 1  
 **else**:  
 print('Ви ввели неправильне число N')  
 print("B = %.2f" % resultB)  
 print("C = %.2f" % resultC)  
 print("Я хочу більше послідовнеостей!!!!!")  
 **else**:  
 print("Ви не ввели число")  
 print("Якщо бажаєте порахувати щось ще, нажміть Enter , якщо хочете закінчить введіть q")  
 a = input()  
 **if** a == "q":  
 print("З вами було приємно мати справу")  
 **break**

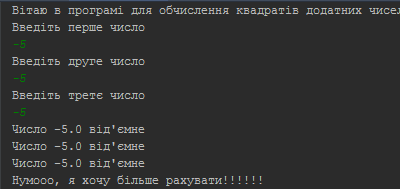
**Результат завдання 1**

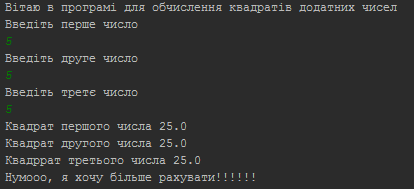
****

****

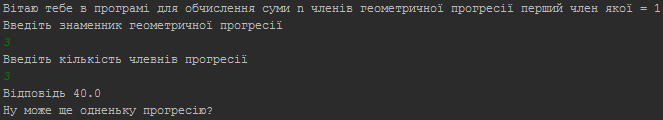
****

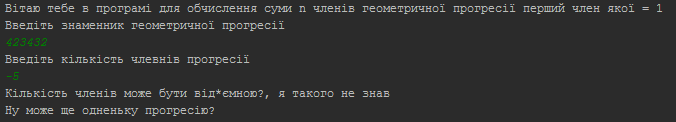
**Результат завдання 2**

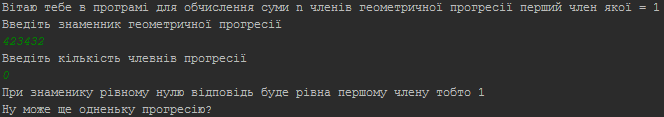
****

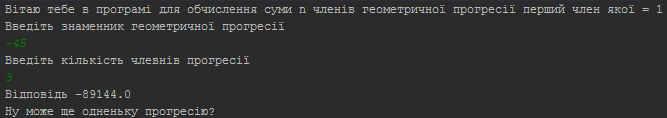
****

**Результат завдання 3\_1**

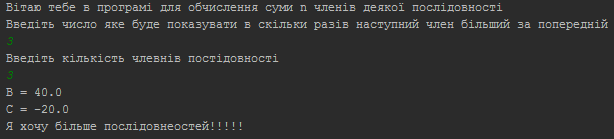
****

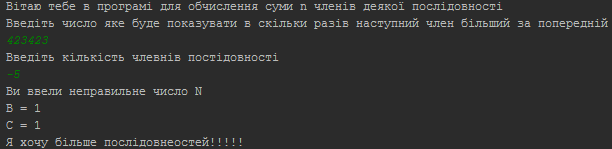
****

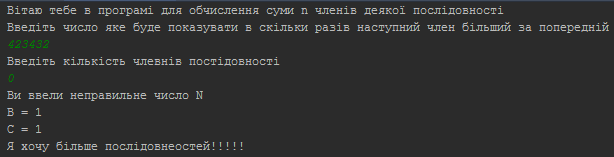
****

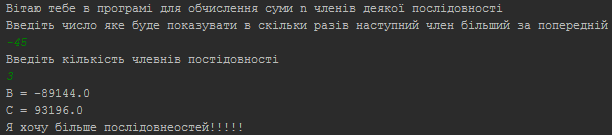
****

**Результат завдання 3\_2**

****

****

****

****

**Висновки**

Складнощів під час виконання роботи майже не було, головна ціль в завданні була створити правильний алгоритм дій та правильно їх передати машині за допомогою синтаксису, але найважчою частиною виконання роботи було виключити всі можливі неправильні ситуації введення символів