# Лабораторна робота 2 Процеси розгалуження в Python

**Мета роботи:** Ознайомитись з операторами розгалуження обчислювальних процесів в Python. Навчитися працювати з числами, використовуючи дану структуру. Побудувати блок-схему алгоритму, розробити програму, реалізувати її та оцінити результат її роботи.

## Рейтинг лабораторної роботи №2

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 0,5 | Вересень |
| 2 | Захист роботи | 0,5 |
| 3 | Звіт з роботи | 0,5 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 2,5 |  |

## Теоретичний матеріал

### Умовний оператор розгалуження if, if-else, if-elif-else

Оператор розгалуження **if** дозволяє виконати певний набір інструкцій залежно від деякої умови. Можливі наступні варіанти використання.

#### 1. Конструкція if

if логічний вираз**:**

    команда\_1

    команда\_2

    ...

    команда\_n

Після оператора **if** записується логічне вираз, результатом обчислення якої є «істина» або «хибність». Якщо цей вислів істинний, то виконуються інструкції, які визначаються даним оператором. Вираз є істинним, якщо його результатом є **число не рівне нулю, непорожній об'єкт, або логічне True**.

Після виразу потрібно поставити двокрапку ":".

ВАЖЛИВО: блок коду, який необхідно виконати, в разі вислову, відділяється чотирма пробілами зліва!

#### 2. Конструкція if - else

Бувають випадки, коли при істинній умові потрібно виконати один набір інструкцій, при помилковій - інший. Для цього використовується конструкція **if - else.**

if логічний вираз:

     команда\_1

     команда\_2

     ...

     команда\_n

else:

     команда\_1

     команда\_2

     ...

     команда\_n

#### 3. Конструкція if - elif - else

Для реалізації вибору з декількох альтернатив використовують конструкцію **if - elif - else.**

if логічне вираз\_1:

    команда\_1

    команда\_2

    ...

    команда\_n

elif логічне вираженіе\_2:

    команда\_1

    команда\_2

    ...

    команда\_n

elif логічне вираженіе\_3:

    команда\_1

    команда\_2

    ...

    команда\_n

else:

    команда\_1

    команда\_2

    ...

    команда\_n

*Приклад*

Програма запитує число у користувача і порівнює його з нулем a <0. Якщо воно менше нуля, то виводиться повідомлення про це. Якщо перше логічне вираження не істинно, то програма переходить до другого - a == 0. Якщо воно істинне, то програма виведе повідомлення, що число дорівнює нулю, в іншому випадку, якщо обидва вищевказаних логічних вирази виявилися хибними, то програма виведе повідомлення, що введене число більше нуля.

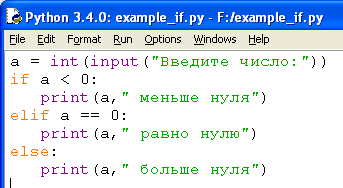


Рис.1. Приклад програми на Python

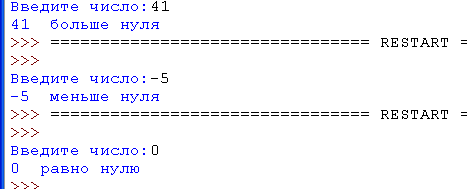


Рис.2 Результат виконання програми з використанням умовного оператора if-elif-else

## Математичні функції та операції в Python

### Цілі числа (int) та дійсні (float)

Числа в Python 3 підтримують набір звичайних математичних операцій:

|  |  |
| --- | --- |
| x + y | Додавання |
| x - y | Віднімання |
| x \* y | Множення |
| x / y | Ділення |
| x // y | Отримати цілу частину від ділення |
| x % y | Остача від ділення |
| -x | Зміна знака числа |
| abs(x) | Модуль числа |
| x \*\* y | Піднесення у степінь |
| pow(x, y) | x в степені y |

### Бібліотека (модуль) math

У стандартну поставку Python входить бібліотека math, в якій міститься велика кількість математичних функцій. Для роботи з даним модулем його попередньо потрібно імпортувати командою import math.

|  |  |
| --- | --- |
| Назва функції | Опис |
| **Функції в бібліотеці math** | |
| ceil(x) | Повертає округлене x як найближче ціле значення типу int, яке дорівнює або перевищує x (округлення "вгору"). |
| fabs(x) | Повертає абсолютне значення (модуль) числа x. В Python є вбудована функція abs, але вона повертає модуль числа з тим же типом, що число, тут же завжди float abs (fabs). |
| factorial(x) | Повертає факторіал цілого числа x, якщо x не ціле виникає виключення ValueError. |
| floor(x) | На противагу ceil (x) повертає округлене x як найближче ціле значення типу int, менше або рівне x (округлення "вниз"). |
| fsum(iterable) | Повертає float суму від числових елементів об'єкта iterable. |
| isinf(x) | Перевіряє, чи є float об'єкт x плюс або мінус нескінченністю, результат відповідно True або False. |
| isnan(x) | Перевіряє, чи є float об'єкт x об'єктом NaN (not a number). |
| modf(x) | Повертає дробову і цілу частину float числа. Обидва результати зберігають знак вихідного числа x і представлені типом float. |
| trunc(x) | Повертає цілу частину числа x у вигляді int об'єкта. |
| **Степеневі та логарифмічні функції** | |
| exp(x) | Повертає e\*\*x. |
| log(x[, base]) | При передачі функції одного аргументу x, повертає натуральний логарифм x (логарифм за основою e = 2.7182 ...). При передачі двох аргументів, другий береться як основа логарифма. |
| log10(x) | Повертає десятковий логарифм x. |
| pow(x, y) | Повертає x в ступені y. На відміну від операції \*\* перетворює обидва аргументи в тип float. |
| sqrt(x) | Квадратний корінь (square root) з x. |
|  | Основа натуральних логарифмів e = 2.71828. |
| **Тригонометричні функції** | |
| acos(x) | Повертає арккосинус x, в радіанах. |
| asin(x) | Повертає арксинус x, в радіанах. |
| atan(x) | Повертає арктангенс x, в радіанах. |
| cos(x) | Повертає косинус x, де x виражений в радіанах. |
| hypot(x, y) | Повертає sqrt (x \*\* 2 + y \*\* 2). Зручно для обчислення гіпотенузи і довжини вектора. |
| sin(x) | Повертає синус x, де x виражений в радіанах. |
| tan(x) | Повертає тангенс x, де x виражений в радіанах. |
| pi | Константа π = 3.1415 ... |
| **Перетворення кутів** | |
| degrees(x) | Конвертує значення кута x з радіан в градуси. |
| radians(x) | Конвертує значення кута x з градусів в радіани. |

### Приклад завдання

Знайти розв’язки квадратного рівняння I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_1.gif, коефіцієнти якого є дійсними числами, що їх вводить користувач. Залежно від значень коефіцієнтів *a, b, c* та дискримінанта I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_2.gif можливі такі результати: всі дійсні числа є коренями (a = 0, b = 0, c=0), коренів немає (a = 0, b = 0, cI:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_3.gif0), є один корінь (a = 0, bI:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_3.gif0), є два різних дійсних корені (aI:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_3.gif0, d>0), два дійсних корені збігаються (aI:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_3.gif0, d=0) або існує два комплексно-спряжених корені (aI:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_02\pictures\pr2_3.gif0, d < 0).

Блок-схема алгоритму:



Рис. 3. Блок-схема алгоритму обчислення евадрптних коренів

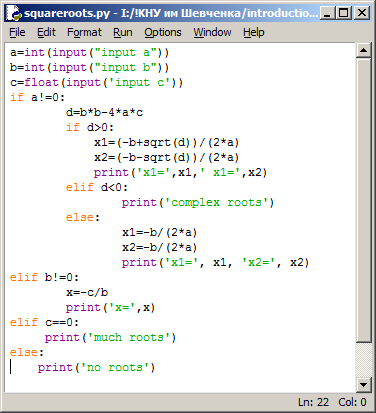


Рис. 4. Код програми розрахунку квадратних коренів

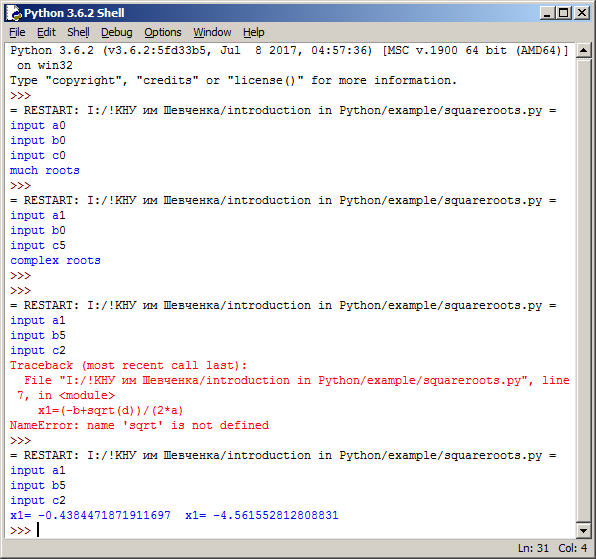


Рис.5. Результати роботи програми

## Завдання для самостійної роботи (за варіантами)

#### **Варіант 1**

Визначити тип трикутника за значеннями уведених його сторін (рівнобедрений, рівнобокій, прямокутний, різнобокій, не існує).

#### **Варіант 2**

Дано три цілих числа. Вибрати з них ті, які належать інтервалу [1,3].

#### Варіант 3

Складіть програму підрахунку площі рівнобедреного трикутника. Якщо площа трикутника парна, розділити її на 2, в іншому випадку вивести повідомлення «Не можу ділити на 2!»

#### Варіант 4

Скласти програму, яка по заданому числу (1-12) виводить назву відповідного йому місяця англійською мовою.

#### Варіант 5

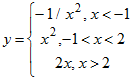
Дано три числа. Знайти кількість додатних чисел серед них;

#### Варіант 6

Якщо дійсні числа x і y мають однаковий знак, знайти їх середнє геометричне, в іншому випадку знайти їх середнє арифметичне.

#### Варіант 7

Обчислити значення функції y, якщо

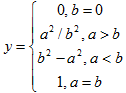


#### Варіант 8

Визначити, чи існує прямокутний трикутник зі сторонами x, y, z, значення яких введено з клавіатури. Якщо так, обчислити його площу.

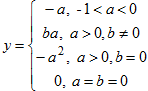
#### Варіант 9

Обчислити значення функції y, якщо



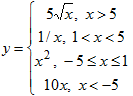
#### Варіант 10

Обчислити значення функції y, якщо



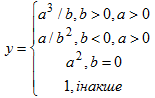
#### Варіант 11

Обчислити значення функції y, якщо



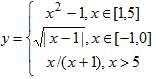
#### Варіант 12

Обчислити значення функції y, якщо



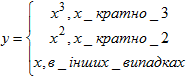
#### Варіант 13

Обчислити значення функції y, якщо



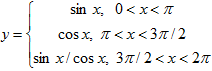
#### Варіант 14

Обчислити значення функції y, якщо



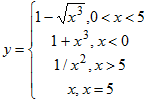
#### Варіант 15

Обчислити значення функції y, якщо



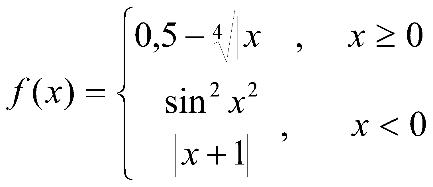
#### Варіант 16

Обчислити значення функції y, якщо



#### Варіант 17

Обчислити значення функції f(x), якщо

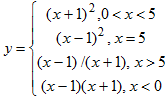


#### Варіант 18

Запросити у користувача два числа. Якщо перше більше другого, то обчислити їх різницю і вивести дані на друк. Якщо друге число більше першого, то обчислити їх суму і вивести на друк. Якщо обидва числа рівні, то вивести це значення на друк.

#### Варіант 19

Обчислити значення функції y, якщо



#### Варіант 20

Визначити умови взаємного розташування двох прямих, що задаються коефіцієнтами рівнянь вигляду *ax* + *by* + *c* = 0 (паралельні, перпендикулярні, перетинаються, збігаються).

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.