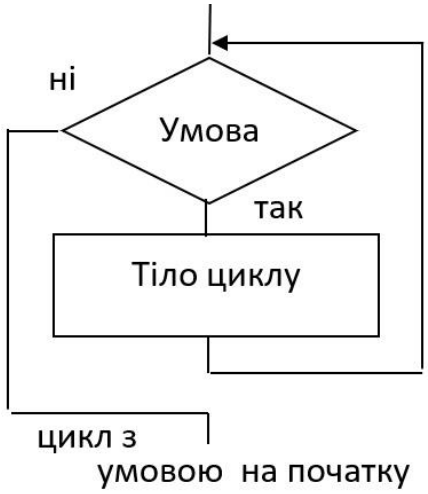
Лабораторна робота № 4

Програмування алгоритмів циклічної структури. Ітераційні циклічні алгоритми. Обчислення значення функції як суми нескінченного ряду.

***Мета роботи**: знайомство з циклічними алгоритмічними конструкціями мови Python.*

Короткі теоретичні відомості

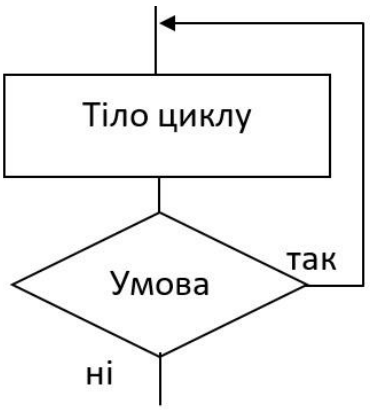
Цикл це повторення деякої послідовності інструкцій алгоритму. Кількість повторів може задаватися явно, або регламентуватися виконанням деякої умови. Послідовність команд, які повторюються в циклі прийнято називати ***тілом циклу***, а одне виконання ***тіла циклу***, зазвичай, називають ***ітерацією***. Цикл разом з розгалуженням та лінійною алгоритмічною конструкцією, розглянутими у попередніх розділах, складають базовий набір конструкцій для побудови алгоритмів будь-якої складності.



Залежно від способу організації ітерацій виділяють три базові конструкції циклу:

- ✓ *цикл з лічильником (параметром) виконує послідовність команд для кожного значення які пробігає деяка змінна-лічильник;*
- ✓ *цикл з умовою на початку перевіряє потребу виконання тіла циклу перед кожною ітерацією, якщо умова виконання хибна вже на початку циклу, то тіло не виконується жодного разу;*

цикл з умовою після тіла виконує перевірку умови після кожної ітерації і вирішує чи потрібно повернутися і повторити інструкції ще раз. Тут команди тіла циклу виконаються обов'язково хоча б один раз.



Такий поділ суто теоретичним, реалізації зазначених концепцій в різних технологіях програмування, як можна бачити далі, доволі розмаїті. Окремі синтаксичні конструкції, як, наприклад, оператор **`for`** в С-подібних мовах мають засоби для реалізації одразу двох концепцій (цикл з лічильником та цикл з умовою на початку). З іншого боку, цикли з умовами за рахунок допоміжних змінних та контролю їх значень дозволяють реалізовувати цикли з лічильником. І, взагалі, до кожної практичної задачі, що розв'язується за допомогою циклу, можна застосувати будь-яку із зазначених форм циклу.

В ***Python*** оператор циклу з умовою на початку має вигляд:

```
while <умова> :

    <блок команд>
```

а оператор з умовою після ітерації, як синтаксична конструкція, відсутній взагалі. Правила виконання стандартні:

- ✓ *виконання блоку команд повторюється до тих пір, поки умова не набуде хибного значення;*
- ✓ *після того, як умова перестає виконуватися, програма переходить до виконання наступного за* **`while`** *оператора;*
- ✓ *вкладений блок команд може містити довільну кількість команд відділену від основного блоку однаковим горизонтальним відступом зліва.*

Цикл з післяумовою, в ***Python***, взагалі кажучи не реалізовано, але його за потреби можна організувати з того циклу ***`while`*** за допомогою оператора дострокового завершення циклу ***`break`*** та оператора розгалуження:

```
while True :

    <блок команд>

    if <умова> : break
```

Python має доволі просту і лаконічну конструкцію циклу for:

```
for <змінна> in <набір значень> :

    <блок команд>
```

Він виконує блок команд для кожного значення змінної у вказаному наборі. Наборами значень можуть бути різного роду колекції та послідовності. Послідовність цілих чисел ***`0, 1, 2, … n-1`***, наприклад, можна утворити функцією ***`range(n)`***.

Завдання 1.

Скласти блок-схему та програму розв’язання задачі відповідно до варіанта. У звіт подати:

- умову задачі варіанта;
- блок-схему алгоритму розв’язування задачі;
- складений Python-код;
- результати обчислень для деякого набору вхідних даних.

Варіанти завдань:

1. Дано два цілі числа A та B (A < B). Знайти суму всіх парних цілих чисел від 2A до 4B включно.

2. Для заданого натурального числа N знайти суму

$$\frac{1}{1!}+\frac{2}{2!}+\frac{3}{3!}+\ldots+\frac{N}{N!},$$

за допомогою одного циклу.

3. Для заданого натурального числа N і дійсного числа X знайти суму

$$\frac{1}{X}+\frac{1}{X^2}+\frac{1}{X^3}+\ldots+\frac{1}{X^N},$$

використовуючи лише один цикл.

4. Дано два цілі числа A і B (A < B). Знайти добуток всіх цілих чисел від A до B включно.

5. Вивести таблицю квадратів усіх натуральних чисел, починаючи від заданого натурального K, до заданого натурального N (N>K)

6. Дано ціле число N > 0. Послідовність натуральних чиселA_k, задана за допомогою формул $A_1 = 3,\ A_2 = 7,\ A_k = 2A_{k-1} + 3A_{k-2},$ k = 3, 4, … . Вивести елементи A_1,A_2,…,A_N.

7. Дано два цілі числа A и B (A < B). Знайти суму квадратів усіх цілих чисел від A до B включно.

8. Для заданого додатного дійсного числа X знайти суму

$$\frac{X}{1}-\frac{X}{2}+\frac{X}{3}-\frac{X}{4}+\ldots+\frac{X}{N},$$

де N найближче до X натуральне число, менше за X .

9. Задано ціле число N > 0. Знайти суму

$$N^2+(N+1)^2+(N+2)^2+\ldots+(2N)^2$$

10. Задано ціле число N (N > 0). Знайти добуток

$$1.1\cdot 1.2\cdot 1.3\cdot \ldots\ (1+0.1*N).$$

11. Задане ціле число N (N > 0). Знайти квадрат даного числа, використовуючи для його обчислення формулу:

$$N^2 = 1+3+5+\ldots+(2N+1).$$

Після додавання до суми наступного доданка виводити поточне значення.

12. Задане натуральне число N. Послідовність дійсних чисел A_k, задана за допомогою формул $A_0 = 2,\ A_k = 2+\frac{3}{A_{k-1}},$ k = 1, 2, … . Вивести елементи A_1,A_2,…,A_N.

13. Задане натуральне число N. Послідовність дійсних чиселA_k, задана за допомогою формул $A_0 = 1,\ A_k = \frac{A_{k-1}+1}{k},$ k = 1, 2, … . Вивести елементи A_1,A_2,…,A_N.

14. Задане дійсне число A і натуральне число N. Використовуючи один цикл, знайти суму

$$1+A+A^2+A^3+\ldots+A^N.$$

15. Для заданого натурального числа N і дійсного числа X знайти суму

$$\frac{1}{X} - \frac{1}{X^2} + \frac{1}{X^3} - \ldots + \frac{(-1)^{N-1}}{X^N},$$

використовуючи лише один цикл.

16. Задане дійсне число A та натуральне число N. Використовуючи один цикл і не використовуючи умовного оператора, знайти суму

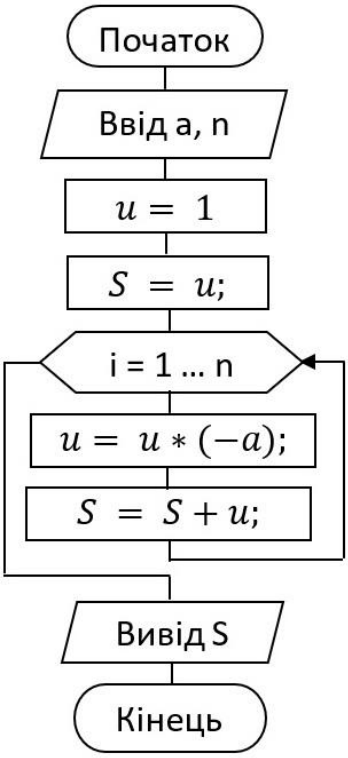
$$1 - A + A^2 - A^3 + \ldots + (-1)^N A^N.$$

Приклад виконання завдання 1

Варіант № 16. Умова задачі: Задане дійсне число A та натуральне число N. Використовуючи один цикл і не використовуючи умовного оператора, знайти суму

$$1 - A + A^2 - A^3 + \ldots + (-1)^N A^N.$$

Умова задачі табулювання функцій передбачає відому наперед кількість ітерацій циклу. Ця кількість або задана явно умовою задачі в якості вхідного параметра (як у випадку цієї задачі), або обчислюється через інші вхідні дані. Тобто, тут оптимальним є застосування конструкцій циклів з лічильником.



Зазвичай, в програмах для обчислення суми запроваджують допоміжну змінну для зберігання поточного доданка, і в тілі циклу додають її значення до суми. В нашій задачі шукаємо суму доданків з почерговою зміною знаку, в початківців така ситуація часто породжує ідеї на кшталт перевірки парності номера доданка і вибору відповідного знаку через умовний оператор. В цьому немає жодної потреби, про що і наголошується в умові задачі. Оскільки кожен наступний доданок відрізняється від попереднього, окрім протилежного знака, на одиницю більшим степенем числа A, то для обчислення чергового доданка

$$(-1)^k A^k = (-1) \cdot (-1)^{k-1} \cdot A \cdot A^{k-1} = -A \cdot (-1)^{k-1} A^{k-1}$$

достатньо домножити попередній доданок на $-A$. Початкове значення змінної (в алгоритмі позначено u) для поточного доданка за умовою задачі дорівнює 1. Подібний підхід застосовується також і в задачі про обчислення значень елементарних функцій за допомогою степеневих рядів.

Алгоритм розв'язування задачі буде таким:

- *спочатку програми отримує від користувача задає число a, для якого обчислюється сума, та кількість доданків n;*

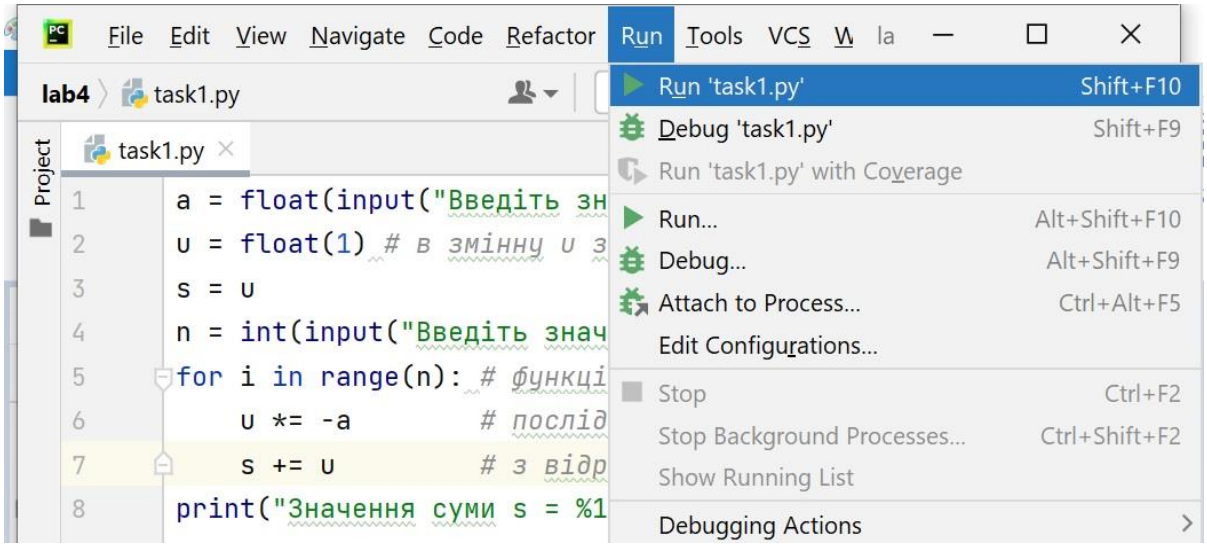
- *програма задає початкові значення суми, та поточного доданка: враховуючи, що сума в умові задачі має* $n+1$ *доданок, додамо до суми перший з них* $u = 1$ *перед циклом, а наступні знаходитимемо і додаватимемо в тілі циклу;*
- *n разів програма повторює виконання таких дій: знаходить значення наступного доданка, домноживши попередній на -a та додає його до суми;*
- *після завершення n ітерацій циклу обчислену суму програма виводить на екран і завершує свою роботу.*

Графічне зображення сформульованого вище словесного опису алгоритму подано у вигляді блок-схеми на малюнку.

Код програми до завдання 1 на мові ***Python***:

```
a = float(input("Введіть значення A: "))
u = float(1) # в змінну u записуємо дійсне число 1.0
s = u
n = int(input("Введіть значення n: "))
for i in range(n): # функція range(n) генерує набір
    u *= -a        # послідовних натуральних чисел з
    s += u         # з відрізка [0; n]
print("Значення суми s = %10.3f" % s)
```

Для набору та виконання цього коду створюємо новий проєкт (наприклад, ***Lab4***) в PyCharm, в ньому новий Python-файл, наприклад, ***task1.py*** . Зверніть увагу, що запустити програмний код з активного вікна в PyCharm можна комбінацією клавіш ***Shift+F10***, або ***Alt+Shift+F10***.



Завдання 2.

Скласти програму для обчислення значення елементарної функції в заданій точці за допомогою степеневого ряду з точністю до члена, меншого за 0.00001. Визначити кількість доданків. Порівняти отримане значення суми зі значенням вказаної функції, обчисленим за допомогою бібліотечних функцій. У звіт подати:

- умову задачі варіанта;
- блок-схему алгоритму розв'язування задачі;
- код програми на Python;
- результат виконання програми для деяких вхідних даних.

Варіанти завдань:

1. $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$;

2. $sin^3 x = \frac{1}{4} \cdot \sum_{k=1}^{\infty} (-1)^{k+1} * \frac{3^{2k+1}-3}{(2k+1)!} * x^{2k+1}$;

3. $\ln x = 2 \sum_{k=1}^{\infty} \frac{1}{2k-1} \left(\frac{x-1}{x+1}\right)^{2k-1}$;

4. $e^{-x^2} = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{k!}$;

5. $\cos^2 x = 1 - \sum_{k=1}^{\infty} (-1)^{k+1} \frac{2^{2k-1} x^{2k}}{(2k)!}$

6. $ln \frac{1+x}{1-x} = 2 \sum_{k=0}^{\infty} \frac{x^{2k+1}}{2k+1}$;

7. $(1+x)^{-\frac{1}{2}} = 1 - \frac{1}{2}x + \frac{1*3}{2*4}x^2 - \frac{1*3*5}{2*4*6}x^3 + \cdots$;

8. $arcctg\, x = \frac{\pi}{2} - \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{2k+1}$;

9. $\cos^3 x = \frac{1}{4} \sum_{k=0}^{\infty} (-1)^k \frac{(3^{2k}+3)}{(2k)!} x^{2k}$;

10. $sh\, x = \sum_{k=0}^{\infty} \frac{x^{2k+1}}{(2k+1)!}$;

11. $\sin x = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}$;

12. $(1+x)^{-\frac{1}{3}} = 1 - \frac{1}{3}x + \frac{1*4}{3*6}x^2 - \frac{1*4*7}{3*6*9}x^3 + \cdots$;

13. $\sin^2 x = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{2^{2k-1} x^{2k}}{(2k)!}$;

14. $ch\, x = \sum_{k=0}^{\infty} \frac{x^{2k}}{(2k)!}$;

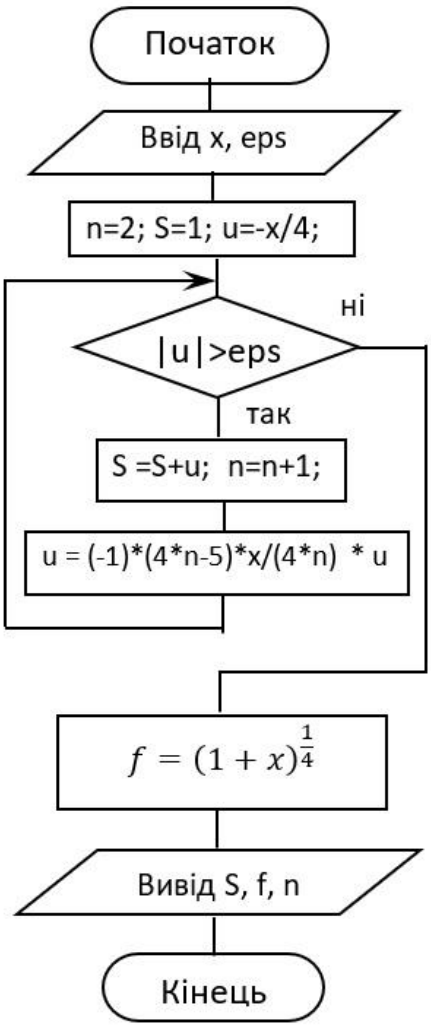
15. $(1+x)^{\frac{1}{4}} = 1 + \frac{1}{4}x - \frac{1*3}{4*8}x^2 + \frac{1*3*7}{4*8*12}x^3 - \frac{1*3*7*11}{4*8*12*16}x^4 + \cdots$.

Приклад виконання завдання 2

Варіант № 15. Умова задачі: Скласти програму для обчислення значення елементарної функції в заданій точці за допомогою степеневого ряду з точністю до члена, меншого за 0.00001. Визначити кількість доданків. Порівняти отримане значення суми зі значенням вказаної функції, обчисленим за допомогою бібліотечних функцій.

$$(1+x)^{\frac{1}{4}} = 1 + \frac{1}{4}x - \frac{1*3}{4*8}x^2 + \frac{1*3*7}{4*8*12}x^3 - \frac{1*3*7*11}{4*8*12*16}x^4 + \cdots .$$

За структурою алгоритму розв'язування задача подібна до задачі з завдання 1, але якщо в завданні 1 кількість доданків n, а отже і кількість ітерацій циклу, програма запитувала в користувача, то в цій задачі кількість ітерацій визначатиметься в ході виконання самого циклу. Тому в цьому алгоритмі слід використовувати не цикл з параметром, а цикл з умовою:



- *користувач вводить значення x, для якого обчислюється функція за допомогою суми ряду;*
- *програма обчислює початковий доданок суми (або кілька доданків);*
- *на кожній ітерації виконується перевірка того, чи знайдений черговий доданок достатньо малий за абсолютною величиною, щоб забезпечити відповідну точність обчислення значення функції;*
- *якщо абсолютна величина чергового доданка не менша від заданої точності, програма додає його значення до суми, обчислює наступний доданок та повторює перевірку;*
- *при досягненні потрібної точності циклічний процес завершується і програма виводить отриманий результат разом і значенням функції, отриманим за допомогою бібліотечних функцій.*

Обчислення загального члена ряду безпосередньо за формулою під знаком суми в умові не завжди можливе через певні обмеження математичного апарату обраної мови програмування. Наприклад, функція піднесення до степеня не працює з від'ємною основою степеня, а потрібно обчислити $(-1)^n$, при великих n значення n! виходить цілочисельного типу, і т.п. В таких випадках для знаходження значення наступного члена ряду A_n зручно використовувати рекурентне співвідношення вигляду $A_n = M_n(x)A_{n-1}$, що пов'язує його із значенням попереднього. Множник $M_n(x)$ завди можна

знайти, формально розділивши вираз для A_n на вираз для A_{n-1}: $M_n(x) = \frac{A_n}{A_{n-1}}$. Деколи, як у випадку цієї задачі, $M_n(x)$ можна побудувати просто проаналізувавши формулу загального члена ряду.

Ряд в умові нашої задачі

$$1 + \frac{1}{4}x - \frac{1*3}{4*8}x^2 + \frac{1*3*7}{4*8*12}x^3 - \frac{1*3*7*11}{4*8*12*16}x^4 + \cdots$$

має певну специфіку, порівняно з більшістю варіантів цього завдання, – тут перші два доданки не зовсім відповідають загальній формулі члена ряду, тому перший доданок включаємо безпосередньо в суму, і обчислення суми починаємо з $S = 1$, а не з $S = 0$. У змінну поточного доданка записуватимемо члени ряду, починаючи з другого ($u = \frac{1}{4}x$). Наступні доданки вже вкладаються в загальну схему і дозволяють побудувати рекурентну формулу.

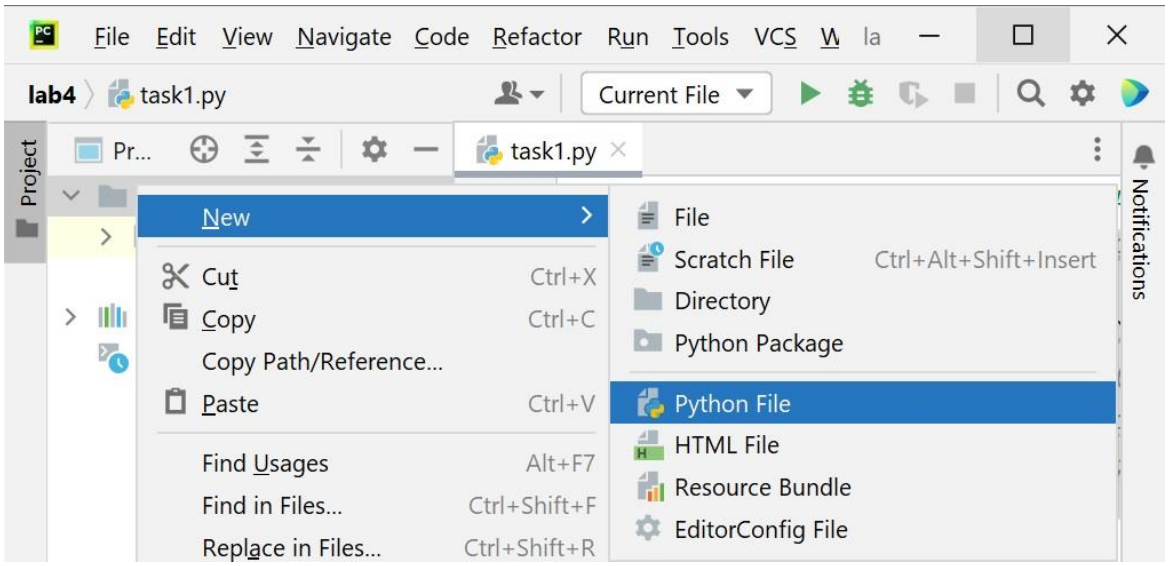
Для побудови згаданого вище коефіцієнта $M_n(x)$ зауважимо, що нашому знакопочерговому ряді кожен наступний член відрізняється від попереднього протилежним знаком, та на одиницю більшим степенем змінної x. У знаменнику доданків маємо добуток послідовних натуральних чисел, кратних 4, котрі можна записати як $4n$, а в чисельнику, починаючи з другого множника бачимо подібні множники: 3 = 4 – 1, 7=8 – 1 = 4*2 – 1, 11 = 12-1 =4*3 – 1, і т.д. Враховуючи порядок, множник в чисельнику можна задати відповідно виразом $4(n-1) - 1 = 4n - 5$. Тобто, $M_n(x) = -\frac{4n-5}{4n}x$ і, починаючи з третього, члени ряду можна шукати за рекурентною формулою

$$A_n = -\frac{4n-5}{4n}x \cdot A_{n-1}.$$

Зупинимося на не зовсім прозорому моменті щодо підрахунку кількості доданків. Нумеруючи доданки з 0 та записавши значення $A_0 = 1$ безпосередньо в початкове значення суми ми використовуємо для початку ітераційного процесу член ряду $A_1 = -\frac{1}{4}x$.

Значення $n = 2$, яким ініціалізується змінна для визначення кількості доданків вказує, що після проходження першої ітерації в сумі буде враховано два доданки A_0 та A_1. Але в самій ітерації значення n збільшується на 1 і по завершенні усього циклу змінна n міститиме кількість доданків, на одиницю більшу за справжню кількість доданків у сумі. Насправді ж n −ий доданок буде обчислено в змінній u, але не додано до суми S. Тому для вказання кількості доданків потрібно виводити значення $n - 1$. Змінити ініціалізацію $n = 2$ не можна, бо ця змінна бере участь в обчисленні наступного члена ряду.

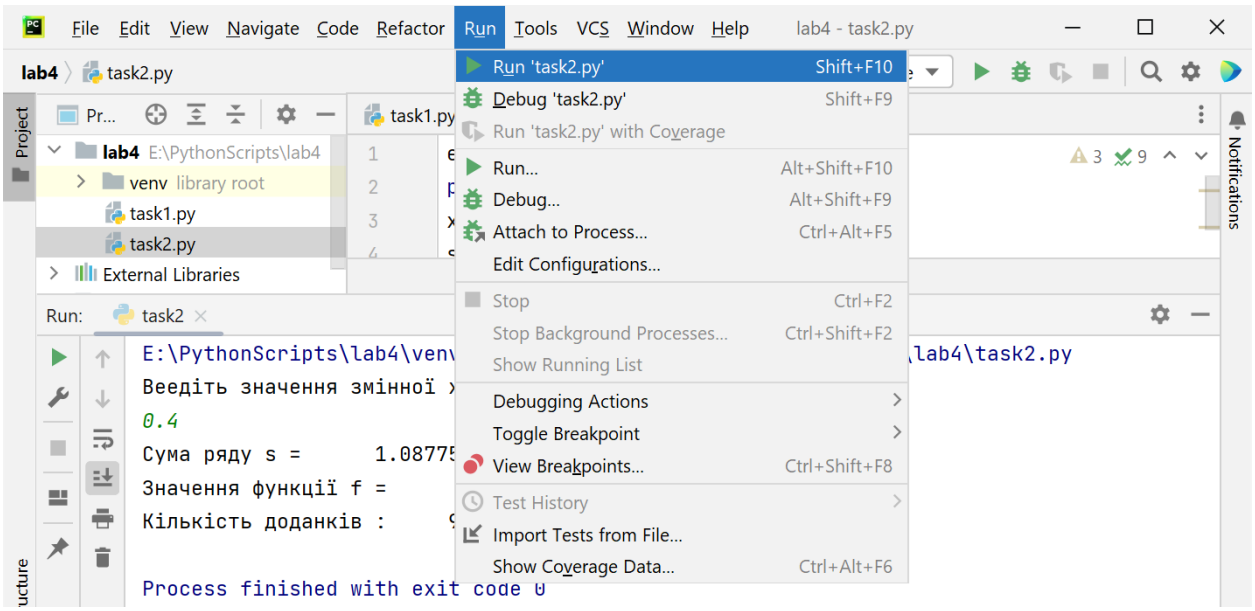
Для коду до задачі 2 скористаємося проєктом ***PyCharm***, створеним для розв'язування завдання1. Проєкти можуть містити багато файлів з кодом, зокрема це використовується для створення власних модулів та підпрограм. ***PyCharm*** дозволяє запускати окремо програми з різних файлів одного і того ж проєкту. Найшвидший спосіб створення нового файла з кодом – виконати команду ***`New=>Python File`*** з контекстного меню вузла проєкту у вікні ***`Project`***.



Назвмемо файл, наприклад, task2 та наберемо в ньому наступний код:

```
eps = 0.00001
print('Введіть значення змінної x: ')
x = float(input())
s = 1; n = 2; u = x / 4
while abs(u) > eps:
    s += u
    u *= -x * (4 * n - 5) / (4 * n)
    n += 1
f = (1 + x)**0.25
print('Сума ряду s = %12.5f' % s)
print('Значення функції f = %12.5f' % f)
print('Кількість доданків : %5d'% (n - 1))
```

Якщо проєкт PyCharm містить декілька файлів з кодом, то виконання команди Run з меню Run, так само, як і комбінації клавіш Shift+F10 буде запускати код з файлу, котрий відкрито в активному вікні.



Активуючи по черзі файли з кодом за допомогою вікна Project, можемо продемнствати роботу кожного з них.

Запитання для самоконтролю

1. Які завдання в програмах вирішують циклічні алгоритмічні конструкції?
2. Які оператори циклу реалізовано в мові програмування ***Python***?
3. Опишіть процес виконання команди циклу **`while`**.
4. Яка синтаксична конструкція ***Python*** призначена для реалізації циклів з лічильником?
5. Чи можна в ***Python*** реалізувати цикл з умовою після тіла циклу?
6. Які розділові знаки та синтаксичні особливості використовуються при написанні команд циклів у ***Python***?