# Лабораторна робота 4 Функції та рекурентні співвідношення в Python

**Мета роботи:** Ознайомитись з технологією використання функцій в Python та застосуванням рекурентних співвідношень. Побудувати блок-схему алгоритму, розробити програму з використанням рекурентних співвідношень для розкладання математичних функцій в ряди Тейлора або Маклорена.

## Рейтинг лабораторної роботи №4

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

## Теоретичний матеріал

Функції в програмуванні можна представити як іменований блок коду, звернення до якого в процесі виконання програми може бути багаторазовим.

Різновиди функцій в Python:

1. Вбудовані функції
2. Функції з імпортованих модулів
3. Функції користувача/

### Вбудовані функції Python

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Built-in Functions** <https://pythoner.name/documentation/library/functions> | | | | |
| abs() | [delattr()](https://docs.python.org/3/library/functions.html) | hash() | memoryview() | set() |
| all() | dict() | help() | min() | [setattr()](https://docs.python.org/3/library/functions.html) |
| any() | dir() | hex() | next() | slice() |
| ascii() | divmod() | id() | object() | sorted() |
| bin() | enumerate() | input() | oct() | [staticmethod()](https://docs.python.org/3/library/functions.html) |
| bool() | eval() | int() | open() | [str()](https://docs.python.org/3/library/functions.html) |
| breakpoint() | exec() | isinstance() | ord() | sum() |
| bytearray() | filter() | issubclass() | pow() | super() |
| bytes() | float() | iter() | print() | tuple() |
| callable() | format() | len() | property() | type() |
| chr() | frozenset() | list() | range() | [vars()](https://docs.python.org/3/library/functions.html) |
| classmethod() | getattr() | locals() | repr() | zip() |
| compile() | globals() | map() | reversed() | \_\_import\_\_() |
| complex() | hasattr() | max() | round() |  |

### Функції користувача можна поділити на:

1. Функції без параметрів, що не повертають значення
2. Функції з параметрами, що не повертають значення
3. Функції з параметрами, що повертають значення.

### Оголошення та виклик функції

|  |  |
| --- | --- |
| Формат оголошення | Оператор виклику функції |
| def <ім’я\_функції>() :  блок коду | ім’я функції() |
| def <ім’я\_функції>(параметри) :  блок коду | ім’я функції(аргументи) |
| def <ім’я\_функції>(параметри) :  блок коду з оператором return | Змінна = ім’я функції(аргументи) |

Визначення функції має бути розташоване перед викликом функції.

***Параметрами*** називають змінні, за допомогою яких здійснюється передавання даних у функцію, що викликається, з програмного блока, який здійснює виклик.

Значення параметрів функції, що вказуються під час її виклику, називають ***аргументами* *функції*.**

### Приклад програми визначення чисел Фібоначчі.

Числа Фібоначчі - це ряд чисел, в якому кожне наступне число дорівнює сумі двох попередніх: 1, 1, 2, 3, 5, 8, 13, ...

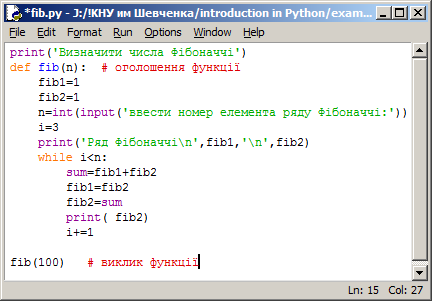


Рис.1. Код програми визначення послідовності Фібоначчі - – варіант1.

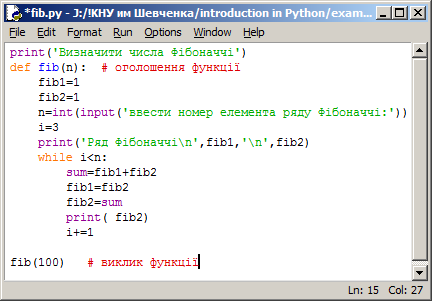


Рис.2. Результат роботи програми визначення послідовності Фібоначчі.

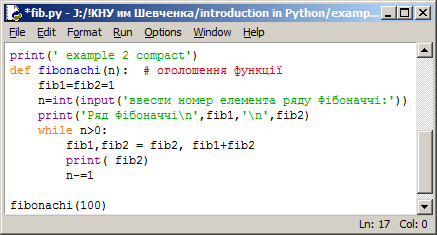


Рис.3. Код програми визначення послідовності Фібоначчі – варіант2.

### Функції, що повертають значення

Функція може повернути значення у точку її виклику. Під час виклику функції, що повертає значення в точку виклику, її ім’я може бути інтерпретовано як ім’я деякої змінної величини. Функцію, що повертає значення, можна викликати у виразах. Для повернення значення з функції в точку виклику використовують оператор повернення **return**.

Оператор повернення значення з функції:

**return (вираз)**

В Python функції здатні повертати кілька значень одночасно:

Наприклад, **return x1, x2**

Всередині функції може міститися довільна кількість операторів **return**. Однак спрацює лише один з них.

Якщо функція не викликає return явно, буде отримано результат **None**. **None**– це спеціальне значення в Python, яке визначає порожній об’єкт, якщо функція нічого не повертає.

### Глобальні змінні

Змінні, які оголошені всередині функції, називають ***локальними***. Ці змінні стають недоступными після виходу з функції.

Змінні, які оголошені поза функцією, але доступні всередині функції, називають **глобальними**. Щоб змінити значення глобальної змінної всередині функції, треба явно вказати що ця змінна є глобальною. Робиться це за допомогою інструкції **global**.

### Параметри за замовчуванням

Значення за замовчуванням використовуються в тому випадку, якщо викликаючи функцію не було вказано відповідний аргумент. **Механізм параметрів за замовчуванням** діє так: якщо змінну проініціалізовано до виклику функції, то у функцію передається саме це значення, в іншому випадку у функцію передається значення за замовчуванням.

def func(a, b, c=2):

return a+b\*c

print(func(1, 2))

### Передавання у функцію змінної кількості аргументів

Для передавання у функцію змінного числа аргументів в Python можна скористатися символом \* перед списком аргументів змінної довжини.

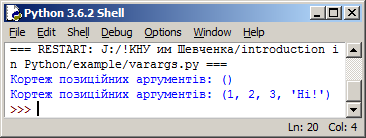
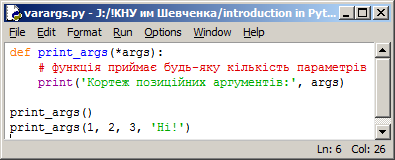


Рис. 4. Приклад функції зі змінною кількістю параметрів та результати роботи програми

### Приклад фрагменту коду лабораторного завдання

**Завдання**. Використовуючи розкладання функції sin(x) в степеневий ряд Маклорена (або Тейлора), написати програму обчислення значення функції sin(x) при заданих аргументах. Для розкладання в степеневий ряд створити функцію мовою Python, що повертає значення і має параметри.

**Теорія**. Розкладання функції sin(x) у степеневий ряд Маклорена має такий вираз:



**Алгоритм**.

1. Ввести значення аргумента функції.
2. Ввести точність розрахунку.
3. Задати початкове значення суми ряду, рівним значенню аргумента.
4. Задати початкове значення елемента ряду, рівним значенню аргумента.
5. Задати початкове значення лічильника циклу рівним 2.
6. Поки модуль поточного елемента ряду більше точності розрахунку повторювати такі дії:

6.1. Обчислити поточний член ряду за формулою:

6.2. Додати поточний елемент до суми ряду.

6.3. Збільшити лічильник циклу на 2. Перейти на п. 6.1.

7. По закінченні циклу повернути значення суми ряду.

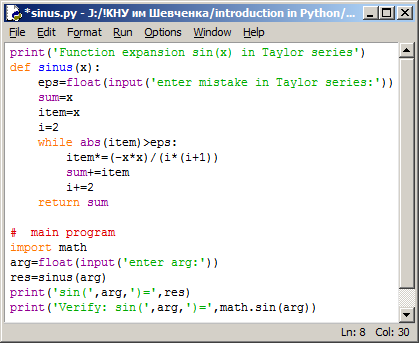
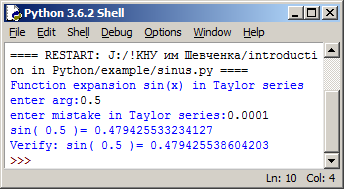
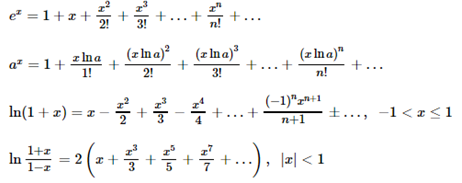
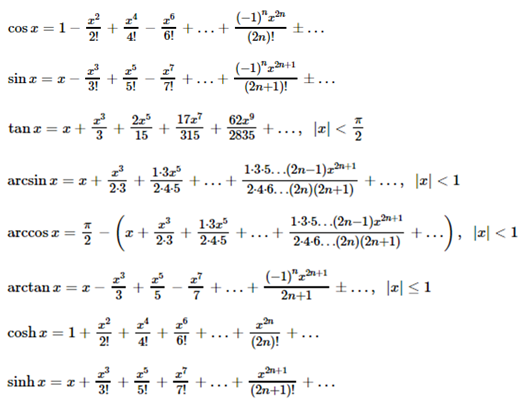
 

Рис. 5. Приклад коду та результати роботи

### Розкладання деяких функцій в ряд Маклорена





## 

## Завдання для самостійної роботи (за варіантами)

Обчислити значення функції, розвинувши її у ряд Маклорена (абоТейлора). Параметр функції має змінюватися від заданого з клавіатури початкового значення, до заданого користувачем кінцевого значення із певним кроком. Розвинення функції в ряд здійснювати із заданої з клавіатури точністю (наприклад, 1е-5). Для розвинення функції у ряд Маклорена (абоТейлора) створити власну функцію, яка розраховує суму ряду за рекурентним співвідношенням. У випадку, коли для вибраних значень аргументів функція не визначена, вивести відповідне повідомлення. Виведення результатів має відповідати рис. 6.

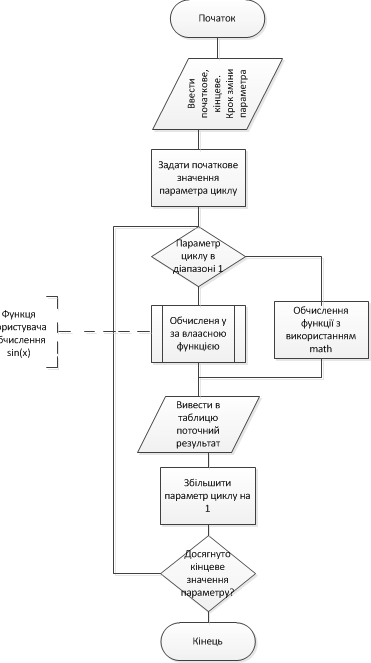


Рис. 6. Блок-схема основної програми

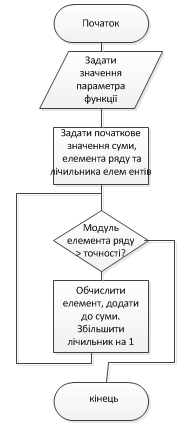


Рис.7. Блок-схема функції обчислення ex за рекурентним співвідношенням

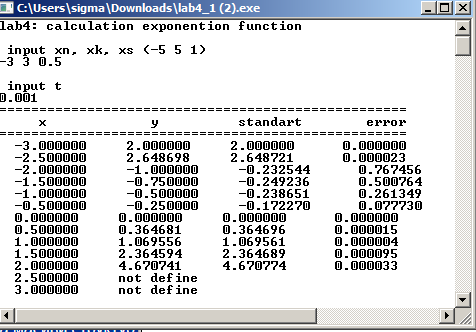


Рис. 8. Зразок результату роботи програми лабораторної роботи №4

### Варіанти завдань

|  |  |
| --- | --- |
| Обчислити значення функції у, розвинувши функцію cos(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var1.gif | Обчислити значення функції у, розвинувши функцію у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var2.gif |
| Обчислити значення функції у, розвинувши функцію е-ху ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку.J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var3.gif | Обчислити значення функції у, розвинувши функцію sin(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. ВизначитиJ:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var4.gif |
| Обчислити значення функції у, розвинувши функцію ln(x) у ряд Тейлора. Аргумент х змінюється від -1 до 3 з кроком 0.5. Визначити похибкуJ:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var5.gif | Обчислити значення функції у, розвинувши функцію arctg(x) у ряд Тейлора. Аргумент х змінюється від 0 до 3 з кроком 0.5. Визначити похибку J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var6.gif |
| Обчислити значення функції у, розвинувши функцію sh(x) у ряд Тейлора. Аргумент х змінюється від 0 до 3 з кроком 0.5. Визначити похибкуJ:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var7.gif | Обчислити значення функції у, розвинувши функцію ex у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var8.gif |
| 1. Обчислити значення функції у, розвинувши функцію ln(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити **похибку** | Обчислити значення функції у, розвинувши функцію sin(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var10.gif |
| Обчислити значення функції у, розвинувши функцію cos(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var11.gif | Обчислити значення функції у, розвинувши функцію e-x у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var12.gif |
| Обчислити значення функції у, розвинувши функцію у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var13.gif | Обчислити значення функції у, розвинувши функцію tg(x) у ряд Тейлора. Аргумент х змінюється від -3 до 3 з кроком 0.5. Визначити похибку J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var14.gif |
| Обчислити значення функції у, розвинувши функцію cos(x) у ряд Тейлора. Аргумент х змінюється від -2 до 3 з кроком 0.5. Визначити похибку J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var15.gif | Обчислити значення функції у, розвинувши функцію arctg(x) у ряд Тейлора. Аргумент х змінюється від -3 до 3 з кроком 0.5. Визначити похибку |
| Обчислити значення функції у, розвинувши функцію tg(x) у ряд Тейлора. Аргумент х змінюється від -3 до 3 з кроком 0.5. Визначити похибкуJ:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var17.gif | Обчислити значення функції у, розвинувши функцію sh(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку y=J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var18.gif |
| Обчислити значення функції у, розвинувши функцію ln(1+x) у ряд Тейлора. Аргумент х змінюється від -3 до 3 з кроком 0.5. Визначити похибку | Обчислити значення функції у, розвинувши функцію ex у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку. |
| Обчислити значення функції у, розвинувши функцію e-x у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку.J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var21.gif | Обчислити значення функції у, розвинувши функцію sh(x) у ряд Тейлора. Аргумент х змінюється від -3 до 3 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var20.gif |
| Обчислити значення функції у, розвинувши функцію ln(x) у ряд Тейлора. Аргумент х змінюється від -2 до 4 з кроком 0.5. Визначити похибку. J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var23.gif | Обчислити значення виразу,J:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var24.gifвикориставши ряди для визначення констант π та е: π/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9 -… ;e = 1 + 1/1! + 1/2! + 1/3! +…. Визначити похибку обчислень. |

## 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.