доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій,
Миколаївський національний аграрний університет
м. Миколаїв, Україна

УДК 004.4

РОЛЬ МОДУЛІВ У РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА PYTHON

Чернега Катерина Петрівна,
здобувач вищої освіти спеціальності 122 «Комп'ютерні науки»
Миколаївський національний аграрний університет
м. Миколаїв, Україна

Анотація: *Модулі відіграють ключову роль у мові програмування Python, забезпечуючи модульність, повторне використання коду та кращу організацію проектів. Ця доповідь розкриває поняття модулів, способи їх підключення до програми з прикладами та детально розглядає призначення та можливості найпопулярніших вбудованих модулів, таких як math, random, turtle, tkinter, pyqt та pygame. Використання модулів дозволяє програмістам ефективно вирішувати різноманітні завдання, покладаючись на готову функціональність та зосереджуючись на логіці власного коду.*

Ключові слова: *Python, модулі, math, random, turtle, tkinter, pyqt, pygame.*

Python - це високорівнева мова програмування, яка відрізняється своєю лаконічністю та зрозумілістю коду. Однією з ключових особливостей Python є наявність величезної кількості вбудованих модулів, які значно розширюють функціональність цієї мови та допомагають програмістам реалізовувати різноманітні завдання. Модулі – це окремі файли з кодом Python, які містять набір функцій, класів та змінних, призначених для певної мети. Вони дозволяють організовувати код у логічні розділи та багаторазово використовувати його в різних проектах.

Для підключення модуля в Python використовується ключове слово import. Існує кілька способів імпортування модулів:

- Імпортування всього модуля: import module_name
- Імпортування окремих функцій або класів з модуля: from module_name import function_name, class_name
- Імпортування всіх функцій та класів з модуля (не рекомендується): from module_name import *
- Імпортування модуля з присвоєнням альтернативного імені: import module_name as alias

Після імпортування модуля його функції та класи стають доступними для використання в програмі.

У Python є багато корисних вбудованих модулів, які допомагають вирішувати різноманітні завдання. Ось деякі з найпопулярніших:

Модуль math надає доступ до математичних функцій, таких як тригонометричні, логарифмічні, степеневі та інші. Наприклад, math.sin(x) повертає синус числа x.

Модуль random дозволяє генерувати псевдовипадкові числа та здійснювати випадковий вибір елементів. Він корисний для створення ігор, моделювання, тестування та інших випадків, де потрібна випадковість.

Модуль turtle забезпечує простий спосіб створення графічних зображень та анімацій. Він особливо корисний для навчання програмування дітей та початківців.

Модуль tkinter є стандартною бібліотекою для створення графічних інтерфейсів користувача (GUI) у Python. Вона дозволяє розробляти вікна, кнопки, меню та інші елементи інтерфейсу.

PyQt – це набір Python-зв'язків для Qt, однієї з найпопулярніших бібліотек для створення крос-платформених графічних інтерфейсів користувача. Він забезпечує більш потужні та гнучкі можливості для розробки GUI, ніж tkinter.

Pygame – це набір модулів Python, призначених для створення відеоігор. Він надає засоби для роботи з графікою, звуком, іграми та мультимедіа.

NumPy – це потужна бібліотека для роботи з багатовимірними масивами та математичними обчисленнями над ними. Вона широко використовується в наукових обчисленнях, обробці даних та машинному навчанні.

Pandas – це бібліотека для аналізу та обробки структурованих (табличних) даних. Вона забезпечує високопродуктивні, гнучкі та експресивні структури даних та інструменти для аналізу даних.

У Python є кілька способів вивести список всіх функцій, що містяться в модулі. Найчастіше використовується функція dir(). Функція dir(module_name) повертає список імен: змінних, модулів, функцій тощо, які визначені в модулі. Наприклад, для модуля math:

```
import math
print(dir(math))
```

Деякі модулі визначають спеціальну змінну __all__, яка є списком об'єктів, які повинні експортуватися з модуля.

```
import math
print(math.__all__)
```

Це виведе список функцій та імен, які розробники модуля math вирішили зробити публічними.

Також можна отримати список всіх атрибутів модуля за допомогою dir(module_name), а потім перевірити, які з них є функціями, використовуючи isinstance() та types.FunctionType або callable().

```
import math
import types
```

```
functions = [attr for attr in dir(math) if isinstance(getattr(math, attr), types.FunctionType)]
print(functions)
```

Цей код створить список всіх атрибутів модуля math, які є функціями.

Усі ці способи дозволяють вивести список функцій, визначених у модулі, що може бути корисним для вивчення модуля, розробки документації чи налагодження.

Використання модулів є важливою практикою, яка дозволяє дотримуватися принципів об'єктно-орієнтованого програмування (ООП) в Python. Модулі забезпечують інкапсуляцію коду, одну з ключових концепцій ООП, оскільки вони дозволяють згрупувати пов'язані дані та функції в одному місці. Це сприяє підвищенню модульності та підтримуваності програмного забезпечення, оскільки кожен модуль відповідає за окрему функціональність і може бути легко замінений або модифікований без впливу на решту системи. Крім того, модулі забезпечують абстракцію коду, приховуючи внутрішні деталі реалізації та надаючи зовнішній інтерфейс для взаємодії з функціоналом. Python пропонує величезну кількість вбудованих модулів, які охоплюють широкий спектр завдань, таких як математичні обчислення, робота з графікою, створення графічних інтерфейсів користувача, розробка ігор та багато іншого.

Таким чином, модулі є невід'ємною частиною ідеології Python, як мови з об'єктно-орієнтованою парадигмою, і їх правильне використання допомагає створювати гнучке, масштабоване та легко підтримуване програмне забезпечення.

Список використаних джерел:

1. Баркалов О. О., Бабаков Р. М. Дослідження продуктивності кросплатформних бібліотек для побудови графічного користувацького інтерфейсу. Прикладні аспекти сучасних міждисциплінарних досліджень. 2022, 78-81.

2. Alan D. Moore. Mastering GUI Programming with Python. Birmingham: Packt Publishing Ltd, 2019. 509 p.

3. Васильєв О. Програмування мовою Python. Тернопіль: «Навчальна книга – Богдан», 2019. 504 с.

Abstract: *Modules play a key role in the Python programming language, providing modularity, code reuse, and better organization of projects. This talk explains the concept of modules, how to connect them to a program with examples, and looks in detail at the purpose and capabilities of the most popular built-in modules, such as math, random, turtle, tkinter, pyqt, and pygame. The use of modules allows programmers to effectively solve various tasks, relying on ready-made functionality and focusing on the logic of their own code.*

Keywords: *Python, modules, math, random, turtle, tkinter, pyqt, pygame.*

Науковий керівник: Тищенко С.І.,

канд. пед., наук, доцент доцент кафедри економічної кібернетики,
комп'ютерних наук та інформаційних технологій,
Миколаївський національний аграрний університет
м. Миколаїв, Україна

СЕКЦІЯ: «ЕКОНОМІКА ПІДПРИЄМСТВ»

УДК 338.24

ВАЖЛИВІСТЬ СТРАТЕГІЧНИХ ЦІЛЬОВИХ ОРІЄНТИРІВ У ПРОЦЕСІ ТРАНСФОРМАЦІЇ ПІДПРИЄМСТВА