



Become QA Auto

```
le.py > ...
x = "PiMyLifeUp" #First assign the 'x' variable a string
type(x) #Output the data type of our 'x' variable
# 43 #Now assign it a integer value
type(x) #Output its type again

days = 5
int(f"There are " + str(days) + " days r

b, c = "Apple", "Banana", "Cherry"

b = c = "Lime"

= "Hello World"
= 453
int(x)
int(y)

= "Life Up"
```



Що таке ООР?

Бутенко
Сергій



План лекції



Парадигми програмування



**Порівняння процедурного та
об'єктно-орієнтованого
програмування**



Історія появи ООП



Недоліки ООП



Основні принципи ООП



Парадигми програмування



Імперативне програмування

це спосіб програмування, що визначає яку послідовність дій потрібно виконати для досягнення результату

- Структурне програмування
- Процедурне програмування
- Об'єктно-орієнтоване програмування

Декларативне програмування

це спосіб програмування, відповідно до якого програма деяким чином описує, що потрібно отримати як результат, а не як це треба зробити.

- Функціональне програмування

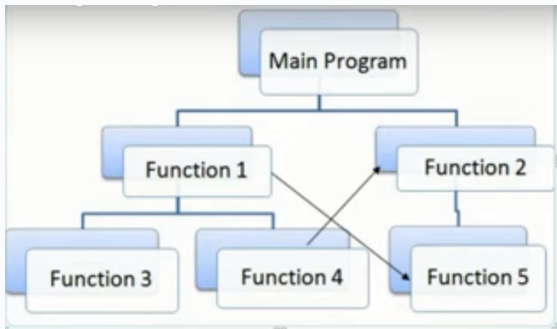


Імперативне програмування



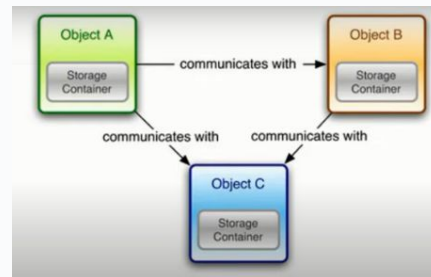
Процедурне програмування –

передбачає набір інструкцій, об'єднаних в процедури, які послідовно виконуються в програмі



Об'єктно-орієнтоване програмування (ООП) –

розглядає програму як сукупність об'єктів, які мають певний стан та певну поведінку і взаємодіють між собою.



Структурне програмування – використовує базові алгоритмічні конструкції



Історія появи ООП



● 1960-1970-і роки

Роботи над мовами Simula. Оле-Джон Даль і Крістен Нігаард, Осло, Норвегія



● 1980-і роки

Інтегрування ООП у мову С та створення C++. Б'ярн Страуструп, Данія



● 1990-і роки

Розроблена проста версія C++ під назвою Java для розробки додатків в Інтернеті. Джеймс Гослінг, Канада

○ Принципи ООП

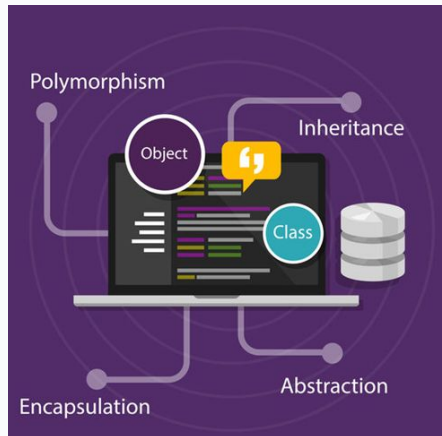


Поліморфізм

забезпечує можливість об'єкта реагувати на одну і ту ж дію по різному.

Інкапсуляція

це механізм зв'язування даних і коду, який опрацьовує ці дані як єдине ціле.



Наслідування

це механізм, завдяки якому один об'єкт може отримувати властивості та функції іншого об'єкта, доповнюючи їх новим функціоналом

Абстракція

це концепція об'єктно-орієнтованого програмування, яка «показує» лише необхідні атрибути та «приховує» непотрібну інформацію



Порівняння процедурного та об'єктно-орієнтованого програмування



Процедурне програмування	ООП
Основний акцент робиться на "як зробити завдання", тобто на процедурі або структурі програми.	Основна увага приділяється введенню додаткових способів подання інформації про предметну область та методів її аналізу
Велика програма ділиться на одиниці, звані функціями.	Вся програма ділиться на об'єкти.
Немає належного способу приховування даних, тому дані є небезпечними	Дані приховані в різних режимах
Глобальні дані розподіляються між функціями програми.	Дані розподіляються між об'єктами.



Недоліки об'єктно-орієнтованого підходу



Ускладнення методології.

Застосування ООП підходу вимагає введення додаткових способів подання інформації про предметну область і методів її аналізу. Для успішного використання подібного механізму потрібна наявність певного рівня кваліфікації у фахівців. Для невеликих проектів більш ефективним може виявитися застосування класичних методів розробки.

Складність реалізації.

Об'єктно-орієнтовані проекти і їх програмна реалізація на об'єктно-орієнтованій мові, вимагають великих витрат часу і призводять до побудови більш складної і вимогливої до ресурсів програми, ніж класичні методи, які можуть виявитися більш ефективними для деяких завдань.



Підсумки



ООП – це перспективна парадигма програмування.



ООП базується на принципах:

- абстракції
- інкапсуляції
- наслідування
- поліморфізму



Недоліки ООП:

- потребує більш ретельного планування
- можливе зниження продуктивності



Переваги ООП:

- повторне використання коду
- код легше обслуговувати
- програми, написані мовами ООП, безпечніші
- код читається простіше