Інкапсуляція

№ уроку: 2 **Курс:** Python Базовий

Засоби навчання: Персональний комп'ютер/ноутбук стандартної продуктивності

Огляд, мета та призначення уроку

Ознайомитись з тим, що таке інкапсуляція і як вона реалізована у Python. У даному уроці буде розглянуто саме поняття інкапсуляції і, на практичних прикладах, буде показано як вона реалізована в Python.

Вивчивши матеріал даного заняття, учень зможе:

- Розуміти, що таке інкапсуляція, і навіщо вона потрібна.
- Знати, як вона реалізована в Python і вміти застосувати її на практиці.

Зміст уроку

1. Що таке інкапсуляція
2. Як вона застосовна на практиці
3. Розв'язання задач

Резюме

- Інкапсуляція - ще одне з основних понять, на якому будується ООП. Вона потрібна для того, щоб організувати рівні доступу до різних архітектурних компонентів, які пише розробник.
- Є три можливі рівні доступу: private (приватний), protected (захищений) та public (публічний)).
- Інкапсуляція в Python відрізняється від її реалізації в інших мовах програмування і носить тут більш неявний характер. Якщо в інших мовах використовуються ключові слова на кшталт public, protected і private, то Python достатньо додати символ "_" (один символ нижнього підкреслення) перед іменем змінної або функції, щоб зробити її protected, а "__ " (два символи нижнього підкреслення) щоб зробити private. Public є всі інші імена, без символів "_" перед ними.
- Наприклад, self.__name = "Alex" є приватним атрибутом і може бути використаний тільки всередині класу, self._name = "Alex" є захищеним атрибутом і може бути використаний тільки всередині модуля, а self.name = "Alex" є публічним та може бути побачений звідки завгодно у коді.
- У даного підходу є свої переваги та недоліки. Перевагою, безумовно, є простота реалізації. Вам не потрібно використовувати ключові слова, а достатньо додати "_" або "__" перед ім'ям змінної або функції або взагалі не додавати. Недоліком є недосконаліть даного підходу. Приватні змінні за фактом не є такими. Якщо в інших мовах програмування ви ніяк не зможете до них "доступитися" поза класом, то в Python, якщо у вас є клас A, у нього є атрибут self._x= 10, то після того, як ви створите об'єкт класу a = A(), то якщо ви зробите print(a.x) буде помилка, але якщо ви зробите print(a._A_x), то виведеться значення змінної x = 10

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Тобто потрібно додати до імені змінної (_x) на початок ім'я класу з 1 "_" перед ним. Тобто "_ClassName variable". Але це дуже погана практика і не потрібно використовувати її в коді. Якщо робите приватні змінні, працюйте з ними як із приватними.
  Законним способом працювати з приватними та захищеними змінними є використання таких функцій як getters та setters. Це спеціальні функції самого класу, які ви самі прописуєте та вирішуєте, як саме вони повинні повертати значення приватних/захищених змінних (getters, наприклад за паролем або просто так) та як встановлювати значення таких атрибутів (setters).

Закріплення матеріалу

- Що таке інкапсуляція?
- Навіщо потрібна інкапсуляція?
- Чи зламається код, якщо в оголошеному класі всі атрибути та функції будуть публічні?
- Чи можна в Python "достукатись" до приватних атрибутів класу?
- Як зробити захищений (protected) модифікатор доступу для змінної self.size?

Додаткове завдання

Вивчити теорію інкапсуляції. Як інкапсуляція допомагає у розробці програмних продуктів? Яким є її призначення?

Самостійна діяльність учня

1. Написати клас, який описує користувача (class User), зробити йому приватний атрибут age, який передається в конструктор, публічний атрибут name, який також передається в конструктор.
2. Написати getter та setter для атрибуту age.
3. Додати до setter перевірку на валідний вік (не від'ємне, ціле число).

Рекомендовані ресурси

- https://www.geeksforgeeks.org/encapsulation-in-python/
- https://www.askpython.com/python/oops/encapsulation-in-python
- https://www.educative.io/edpresso/what-is-encapsulation-in-python

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com