Инкапсуляция

№ урока:	2	**Курс:**	Python Базовый

Средства обучения:	Персональный компьютер/ноутбук стандартной производительности

Обзор, цель и назначение урока

Познакомиться с тем, что такое инкапсуляция и как она реализована в Python. В данном уроке будет рассмотрено само понятие инкапсуляции и, после, на практических примерах, будет показано как она реализована в Python.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое инкапсуляция и зачем она нужна.
- Знать, как она реализована в Python и уметь применить ее на практике.

Содержание урока

1. Что такое инкапсуляция
2. Как она применима на практике
3. Решение задач

Резюме

- Инкапсуляция - еще одно из основных понятий, на котором строится ООП. Она нужна для того, чтобы организовать уровни доступа к различным архитектурным компонентам, которые пишет разработчик.
- Есть 3 возможных уровня доступа: private (приватный), protected (защищенный) и public (публичный).
- Инкапсуляция в Python отличается от ее реализаций в других языках программирования и носит здесь более неявный характер. Если в других языках используются ключевые слова вроде public, protected и private, то в Python достаточно добавить символ "_" (один символ нижнего подчеркивания) перед именем переменной или функции, чтобы сделать ее protected, а "__" (два символа нижнего подчеркивания) чтобы сделать private. Public являются все остальные имена, без символов "_" перед ними.
- Например *self.__name = "Alex"* является приватным атрибутом и может быть использован только внутри класса, self._name = "Alex" является защищенным атрибутом и может быть использован только внутри модуля, а *self.name = "Alex"* является публичным и может быть увиден откуда угодно в коде.
- У данного подхода есть свои преимущества и недостатки. Преимуществом безусловно является простота реализации. Вам не нужно использовать ключевые слова, а достаточно добавить "__" или "_" перед именем переменной или функции или не добавлять вообще. Недостатком же является несовершенство данного подхода. Приватные переменные по факту не являются таковыми. Если в других языках программирования вы никак не сможете до них "достучаться" из вне класса, то в Python, если у вас есть класс A, у него есть атрибут self.__x= 10, то после того как вы создадите

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

объект класса a = A(), то если вы сделаете print(a.x) будет ошибка, но если вы сделаете print(a._A__x), то выведется значение переменной __x = 10.

- То есть вам нужно добавить к имени переменной (__x) в начало имя класса с 1 "_" перед ним. То есть "_ClassName__variable". Но это очень плохая практика и не нужно ее использовать в коде. Если делаете приватные переменные, работайте с ними как с приватными.
  Законным способом работать с приватными и защищенными переменными является использование таких функций как getters и setters. Это специальные функции самого класса, которые вы сами прописываете и решаете, как именно они должны возвращать значения приватных/защищенных переменных (getters, например по паролю или просто так) и как устанавливать значения таких атрибутов (setters).

Закрепление материала

- Что такое инкапсуляция?
- Зачем нужна инкапсуляция?
- Поломается ли код, если в объявленном классе все атрибуты и функции будут публичные?
- Можно ли в Python "достучаться" до приватных атрибутов класса?
- Как сделать защищенный (protected) модификатор доступа для переменной self.size?

Дополнительное задание

Изучить теорию инкапсуляции. Каким образом инкапсуляция помогает в разработке программных продуктов? Каково ее назначение?

Самостоятельная деятельность учащегося

1. Написать класс, который описывает пользователя (class User), сделать ему приватный атрибут age, который передается в конструктор, публичный атрибут name, который так же передается в конструктор.
2. Написать getter и setter для атрибута age.
3. Добавить в setter проверку на валидный возраст (не отрицательное, целое число).

Рекомендуемые ресурсы

- https://habr.com/ru/post/444338/
- 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