**Задача "Покупка дома"**

Итак, мы с вами узнали, почему при разработке современных программ использование объектно-ориентированного подхода является обязательным условием. Также разобрались в понятиях **Класс**, **Объект**(Экземпляр), **Атрибут**, **Свойство**(Поле), **Метод**. Далее посмотрели, какими эти самые атрибуты, свойства и методы бывают. А еще научились отличать **Protected** атрибуты от **Private** и разобрались, как реализована модель уровней доступа к атрибутам непосредственно в Python. Теперь давайте постараемся эти знания применить на практике.  
  
Перед вами задача **"Покупка дома"**. С помощью подхода ООП и средств Python в рамках данной задачи необходимо реализовать следующую предметную структуру:

**Классовая структура**  
Есть **Человек**, характеристиками которого являются:

1. Имя
2. Возраст
3. Наличие денег
4. Наличие собственного жилья

**Человек** может:

1. Предоставить информацию о себе
2. Заработать деньги
3. Купить дом

Также же есть **Дом**, к свойствам которого относятся:

1. Площадь
2. Стоимость

Для **Дома** можно:

1. Применить скидку на покупку

Также есть **Небольшой Типовой Дом**, обязательной площадью 40м2.

---------------------------------------------------------------------------------------------------------------------------------------

Задание. Часть 1. Класс **Human**

1. Создайте класс **Human**.
2. Определите для него два статических поля: **default\_name** и **default\_age**.
3. Создайте метод **\_\_init\_\_()**, который помимо **self** принимает еще два параметра: **name** и **age**. Для этих параметров задайте значения по умолчанию, используя свойства **default\_name** и **default\_age**. В методе **\_\_init\_\_()** определите четыре свойства: Публичные - **name** и **age**. Приватные - **money** и **house**.
4. Реализуйте справочный метод **info()**, который будет выводить поля **name**, **age**, **house** и **money**.
5. Реализуйте справочный статический метод **default\_info()**, который будет выводить статические поля **default\_name** и **default\_age**.
6. Реализуйте приватный метод **make\_deal()**, который будет отвечать за техническую реализацию покупки дома: уменьшать количество денег на счету и присваивать ссылку на только что купленный дом. В качестве аргументов данный метод принимает объект дома и его цену.
7. Реализуйте метод **earn\_money()**, увеличивающий значение свойства **money**.
8. Реализуйте метод **buy\_house()**, который будет проверять, что у человека достаточно денег для покупки, и совершать сделку. Если денег слишком мало - нужно вывести предупреждение в консоль. Параметры метода: ссылка на дом и размер скидки

Задание. Часть 2. Класс **House**

1. Создайте класс **House**
2. Создайте метод **\_\_init\_\_()** и определите внутри него два динамических свойства: **\_area** и **\_price**. Свои начальные значения они получают из параметров метода **\_\_init\_\_()**
3. Создайте метод **final\_price()**, который принимает в качестве параметра размер скидки и возвращает цену с учетом данной скидки.

Задание. Часть 3. Класс SmallHouse

1. Создайте класс **SmallHouse**, унаследовав его функционал от класса **House**
2. Внутри класса **SmallHouse**переопределите метод **\_\_init\_\_()**так, чтобы он создавал объект с площадью 40м2

Задание. Часть 4. Тесты

1. Вызовите справочный метод **default\_info()**для класса **Human**
2. Создайте объект класса **Human**
3. Выведите справочную информацию о созданном объекте (вызовите метод **info()**).
4. Создайте объект класса **SmallHouse**
5. Попробуйте купить созданный дом, убедитесь в получении предупреждения.
6. Поправьте финансовое положение объекта - вызовите метод **earn\_money()**
7. Снова попробуйте купить дом
8. Посмотрите, как изменилось состояние объекта класса **Human**