第二十五次课

综合案例

需求

1. 设计一个 Game 类
2. 属性：

    定义一个 类属性 top_score 记录游戏的 历史最高分

    定义一个 实例属性 player_name 记录 当前游戏的玩家姓名
3. 方法：

    静态方法 show_help 显示游戏帮助信息

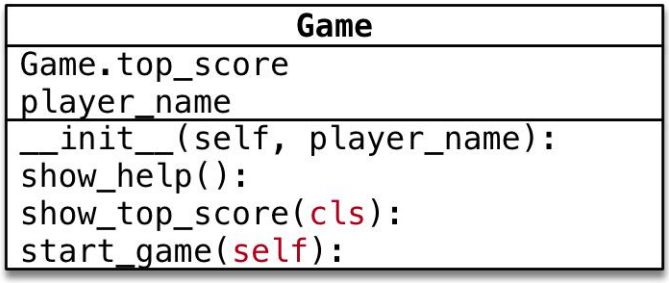
    类方法 show_top_score 显示历史最高分

    实例方法 start_game 开始当前玩家的游戏
4. 主程序步骤

    1) 查看帮助信息

    2) 查看历史最高分

    3) 创建游戏对象，开始游戏



案例小结

1. **实例方法** —— 方法内部需要访问 实例属性

    实例方法 内部可以使用 **类名**. 访问类属性
2. **类方法** —— 方法内部 只 需要访问 类属性
3. **静态方法** —— 方法内部，不需要访问 实例属性 和 类属性

提问：

如果方法内部 即需要访问 **实例属性**，又需要访问 **类属性**，应该定义成什么方法？

应该定义 **实例方法**

因为，**类只有一个**，在 实例方法 内部可以使用 **类名.访问类属性**

```
class Game(object):

    # 游戏最高分，类属性
    top_score = 0

    @staticmethod
    def show_help():
        print("帮助信息：让僵尸走进房间")

    @classmethod
    def show_top_score(cls):
        print("游戏最高分是 %d" % cls.top_score)

    def __init__(self, player_name):
        self.player_name = player_name

    def start_game(self):
        print("[%s] 开始游戏..." % self.player_name)

        # 使用类名.修改历史最高分
        Game.top_score = 999

# 1. 查看游戏帮助
Game.show_help()

# 2. 查看游戏最高分
Game.show_top_score()

# 3. 创建游戏对象，开始游戏
game = Game("小明")

game.start_game()

# 4. 游戏结束，查看游戏最高分
Game.show_top_score()
```