

少儿编程

21工作室出品

抽象

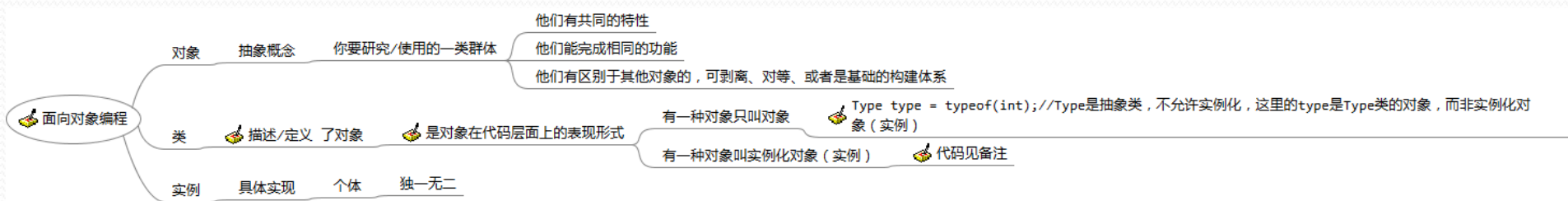
- 这一张我们会学习一些抽象的东西
- 我们把一堆人分类，看成一类人，比如：
 - 按职业：老师、警察、医生、军人
 - 按年龄：老年人、青年、幼儿
- 把食物分类，比如：
 - 按营养：蔬菜、水果、主食、肉类、坚果
 - 按烹饪：熟食、生食

具体

- 老板，我要买一个水果——老板懵
- 老板，我要买一斤苹果——挑选好的一袋子水果是独一无二的
- 老板，我要买这个苹果

对象、类、实例

- 抽象具有研究和使用的意义的一类群体，我们定义为对象（object）
- 那么描述这类人或物或事的代码，我们定义为类（class）
- 具体产生到个体，能够区分出来的独一份，我们定义为实例（instance）



作文题目：描述一场战争

- 流水账式描写：

士兵许，枪ak47:弹量0

装弹50发

准备完毕，剩余弹量50

自动射击30发，剩余弹量20

士兵吴，枪巴雷特:弹量0

装弹30发

准备完毕，剩余弹量30

点射,剩余弹量29

点射,剩余弹量28

点射,剩余弹量27

点射,剩余弹量26

点射,剩余弹量25

士兵郭，枪ump45:弹量0

装弹15发

准备完毕，剩余弹量15

点射射击1发,剩余弹量14

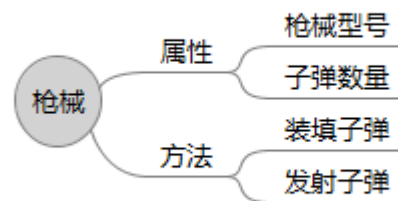
点射射击1发,剩余弹量13

这是在干吗？

- 是在演习吗？
- 是什么兵种？
- 只有陆军吗？

面向对象思维

- 抽象：
 - 战争要素：
 - 人
 - 武器
 - 打仗过程
- 主次：
 - 提纲挈领、一言以蔽之
 - 迅速获得主要信息并且可以1.快速、2.深入了解事实
 - 极大的扩展性




```
class Soldier:
    """士兵"""

    def __init__(self, name): # 特殊函数1: 初始化
        self.name = name # 名字
        self.gun = None # 拿了什么枪

    def __str__(self): # 特殊函数2: 字符串化
        return "士兵%s, 枪%s" % (self.name, self.gun)

    def prepare(self, num):...

    def fire(self, num, fire_mode="点射"):...
```

```
class Gun:
    """枪"""

    def __init__(self, model):
        self.model = model # 枪械型号
        self.bullet_count = 0 # 子弹量

    def __str__(self):
        return "%s:弹量%d" % (self.model, self.bullet_count)

    def add_bullet(self, count):...

    def sub_bullet(self, num, fire_mode):...
```

```
xu = Soldier("许") # 1.培养一个战士
xu.gun = Gun("AK47") # 2.给一个士兵一把枪
print(xu) # 打印战士状态
xu.prepare(50) # 3.让枪装上子弹，一等兵配额15发，二等兵配额30发，三等兵配额50发
xu.fire(30, "自动") # 4.让士兵开枪
print("-----")
wu = Soldier("吴")
wu.gun = Gun("巴雷特")
print(wu)
wu.prepare(30)
wu.fire(5) # 打5颗子弹
print("-----")
guo = Soldier("郭")
print(guo)
guo.prepare(15)
guo.fire(1)
guo.fire(1)
```

面向过程思维

- 详细描述一个细节，盲人摸象、管中窥豹、坐井观天
- 不厌其烦的描述完所有细节，才能总揽全局
- 不听完就不知道在说什么
- 对程序的坏处：
 - 很难找到要修改的位置
 - 一修改会牵扯到很多其他人、其他事、影响他人，耦合度高难以剥离、难以隔离处理
 - 难以抽象化处理，一人一事不能批量、集中、智能化处理

- 面向对象的思维：
 - 第一层领导人视角：海陆空-进军
 - 第二层指挥官视角：舰上炮兵-装炮弹-开火、陆军-装子弹-射击
 - 第三层士兵视角：步兵9527号，二等兵，m762，自动档位，射出25颗子弹
- 过程式思维：
 - 喂，你这个士兵叫什么？带武器了吗？拿了什么样的枪？配额是多少？开火之前有子弹吗？开火射了多少颗子弹？还剩多少子弹？
 - 喂，你这个士兵叫什么？带武器了吗？拿了什么样的枪？配额是多少？开火之前有子弹吗？开火射了多少颗子弹？还剩多少子弹？
 - 喂，你这个士兵叫什么？带武器了吗？拿了什么样的枪？配额是多少？开火之前有子弹吗？开火射了多少颗子弹？还剩多少子弹？
 -
- 如果按过程，一个战士一个战士的去布置任务，没开战就已经输了

用面向对象的方法设计一个小游戏

- 射击小游戏(最基础的要素):
 - 士兵（面向对象设计），以对象的思想进行设计
 - 特性：有名字、有枪
 - 动作：会开枪
 - 枪（面向对象设计）
 - 特性：枪型号、子弹个数
 - 动作：装子弹、射出子弹
 - 战争过程（面向过程设计），以过程的思想进行设计
 - 1.培养一个战士
 - 2.给一个士兵一把枪
 - 3.让枪装上子弹
 - 4.让士兵开枪

果盘

- 用过程的思维去做：
 - 洗苹果-削苹果皮-削苹果块
 - 洗芒果-削芒果皮-削芒果块
 - 洗梨-削梨皮-削梨块
- 用面向对象的思维去削
 - 洗水果（水果=苹果、芒果、梨）-削水果-削块-拼盘
- 步骤简化了、清晰明了了
- 可移植性增强了、可套用、可复用、可拓展

你学到了什么

- 什么是对象。
- 属性和方法。
- 什么是类。
- 创建类的一个实例。
- 特殊方法: `__init__()` 和 `__str__()`。
- 多态。
- 继承。
- 代码桩。

测试题

1. 定义一个新的对象 类型时用什么关键字?
2. 什么是属性?
3. 什么是方法?
4. 类和实例之间有什么区别?
5. 方法中实例引用通常用什么名字?
6. 什么是多态?
7. 什么是继承?

动手试一试

1. 为 `BankAccount` 建立一个类定义。它应该有一些属性，包括账户名（一个字符串）、账号（一个字符串或整数）和余额（一个浮点数），另外还要有一些方法显示余额、存钱和取钱。
2. 建立一个可以挣利息的类，名为 `InterestAccount`。这应当是 `BankAccount` 的一个子类（所以会继承 `BankAccount` 的属性和方法）。`InterestAccount` 还应当有一个对应利息率的属性，另外有一个方法来增加利息。为了力求简单，假设每年会调用一次 `addInterest()` 方法计算利息并更新余额。