面向对象编程的作业

基本概念

1. 面向过程编程与面向对象编程的区别与应用场景？
2. 初始化函数、析构函数与普通函数的区别？
3. 绑定对象的函数、绑定类的函数、静态函数的区别？
4. 类的属性和对象的属性有什么区别？
5. 类和对象的属性与方法在内存中是如何保存的（__dict__）。
6. 描述继承、派生和组合的基本概念、适用场景，及典型用法示例。
7. 描述抽象和多态的基本概念及适用场景，python“鸭子类型”的含义，及典型用法。
8. 描述封装和接口的基本概念，封装一共分为几种类型，各自的适用场景，及典型用法示例。

编程操作（必做）

1. 用面向对象理念，重构“进击的小鸟”
2. 面向对象编程的基本练习：
   - 编写一个课程类(Course)
     - 包含属性：名称(name, string)，教师列表（teachers, list），学生列表（students，list)，成绩（scores，dict）
     - 主要方法：添加/删除学生， add_student/drop_student，添加/删除教师，add_teacher/drop_teacher，添加/查找某位同学的分数：set_score/get_score
   - 编写一个学生类(Student)，
     - 包含属性：姓名(name , string)，性别(sex, int) ，学号(student_id , int)，已选修课程(courses，列表)
     - 主要方法： 选修/退出某课程 select_course/ drop_course，查看自己某个课

程的成绩， get_score

- 编写一个教师类(Teacher)：
  - 包含属性：姓名(name , string)，性别(sex , int) ，拥有的课程(courses，列表)
  - 主要方法：选修/退出某课程 select_course/ drop_course，给某个课程和某个课程打成绩， set_score。
- 完成如下操作：
  - 定义 10 个学生，3 个教师、5 门课程。
  - 某一名学生选择某一门课程
  - 某一名教师选择某一门课程
  - 教师打分。
  - 学生查询分数。

3. 继承的练习：
   - 定一个动物类 Animal，再定义两个子类：Human、Dog
     - Animal，属性： id，name, life, power, weight, pos， 方法： move(self, pos)
     - Human，新增加属性： father, mother，新增加方法: shout(self, info)，attack(self, dog)
     - Dog， 新增加属性：price， 新增加方法: bark(self, info) ， bite(self, human)
   - 生成两个队列， 各有 10 名成员，随机碰撞，碰到则互相进攻，血条降为 0，则死亡，最终剩余的一队，宣布胜利。
4. 多态的练习：几何形状
   - 定义 3 个类： Rect, Circle, Square
   - 实现两个函数功能的多态化：
     - 求解周长：perimeter()
     - 求解面积：area()
   - 两种方式来生成对象：静态、动态。

- 封装的练习：
  - 定一个类（Person)：
    - 把年龄和体重属性隐藏起来
    - 通过 property，确保 name 不允许非字符串式赋值。。。

- 通过 property，确保输入每个月的 salary 值不允许非数字赋值
- 通过 property，确保 height 不允许非数字赋值
- 通过 property，确保 age 不允许非整形数字赋值
- 通过 property，返回人体的 BMI 指标（体重千克数除以身高米数的平方得出的数字）
- 通过 property，返回当年的平均月工资、年工资总和。

5. 自己封装 1 个 MyDict，支持以下操作：
   - 支持 for … in
   - [] 操作， 访问 key-value 值，类似于：dic[key] = value
   - 通过.操作，访问 key-value 值，类似于：dic.key = value
   - 要求: key 只能是字符串（字符串的字符构成只能是下划线、数字或字母）
   - 支持 get 操作， 输入 key， 返回 value…
   - 支持 len() 操作
6. 自己封装 1 个 MyList，支持以下操作：
   - 支持 for … in
   - [] 操作， 输入下标，返回下标对应的结果，比如：lst[1] = value
   - 元素只支持字符串、数字，不支持其他类型的变量。
   - 支持 find()操作
   - 支持 len() 操作
7. 自己封装 1 个素数发生器 MyPrimeGenerator 的类：
   - 支持设置最小值、最大值
   - 支持 for … in
8. 自己封装 1 个文件接口类 File：
   - 支持 with 语法
   - 支持 open、close 操作
   - 支持 read、readline、readlines 等操作
   - 支持 write、writeline、writelines 等操作
   - 支持几种典型的异常操作。

编程操作（选做）

1. 定义 MyBTree、MyBNode 类

- MyBTree 要求：
  - 成员变量：__root, 含义：根节点
  - 用 property 去封装/访问 私有变量
  - 成员函数：添加节点：add_node(node)、查找节点：find_node(node, style = “wfs”)
- MyBNode 要求：
  - 成员变量：name(名称), __value(值), __parent(父节点)、__left(左节点)、__right(右节点)
  - 用 property 去封装/访问 私有变量
  - name 只是能字符串, value 只能是整形或浮点型数。
  - 成员函数：add_node(node)、find_node(node, style = ‘wfs’) ，封装函数：
    - 广度优先搜索：__find_node_wfs(node)
    - 深度优先搜索：__find_node_dfs(node)
    - 利用 2 叉树特性：__find_node_btree(node)