面向对象第4天：

潜艇游戏第一天：

1. 创建了6个类，创建World类并测试

潜艇游戏第二天：

1. 给6个类添加构造方法，并测试

潜艇游戏第三天：

1. 设计侦察潜艇数组、鱼雷潜艇数组、水雷潜艇数组、水雷数组、炸弹数组，并测试
2. 设计SeaObject超类，6个类继承超类
3. 给超类设计两个构造方法，6个类分别调用

潜艇游戏第四天：

1. 将侦察潜艇数组、鱼雷潜艇数组、水雷潜艇数组，统一组合成SeaObject数组，并测试
2. 在6个类中重写move()，并测试
3. 画窗口：在World类中，共3步----------不需要掌握，CV大法
    - import JFrame+JPanel
    - 设计World类继承JPanel----------这一步特别容易忘记
    - main中代码CV大法

回顾：

1. 引用类型数组：
    - 想给元素赋值，需要new个对象
    - 若想访问对象的数据，需要通过元素打点来访问

```
Student[] stus = new Student[3];
stus[0] = new Student("zhangsan",25,"LF");
System.out.println(stus[0].name);
```

2. 继承：
    - 作用：代码复用
    - 通过extends来实现继承
    - 超类/父类：共有的属性和行为

      派生类/子类：特有的属性和行为
    - 派生类既能访问派生类的，也能访问超类的。但超类不能访问派生类的。
    - 一个超类可以有多个派生类，但一个派生类只能有一个超类----------单一继承
    - 继承具有传递性
    - java规定：构造派生类之前必须先构造超类

        - 在派生类的构造方法中若没有调用超类的构造方法，则默认super()调用超类的无参构造方法
        - 在派生类的构造方法中若自己调用了超类的构造方法，则不再默认提供
3. super：指代当前对象的超类对象

   super的用法：

    - super.成员变量名----------访问超类的成员变量
    - super.方法名()----------调用超类的方法
    - super()----------调用超类的构造方法

精华笔记：

1. 向上造型：----------代码复用
    - 超类型的引用指向派生类的对象
    - 能点出来什么，看引用的类型----------这是规定，记住就OK

    > 何时向上造型：
    >
    > - 多种角色能干的事都一样的时候，可以将多种角色统一造型到超类数组中，实现代码复用
    >
    >   eg: 学生/老师/医生都是输出名字+问好------干的事都一样，
    >
    >   就可以将学生/老师/医生统一造型到Person数组中，这样一个for即可-------代码复用

2. 方法的重写(override/overriding)：重新写、覆盖
    - 发生在父子类中，方法名相同，参数列表相同
    - 重写方法被调用时，看对象的类型----------这是规定，记住就OK
    - 重写需遵循"两同两小一大"原则：----------了解，一般都是一模一样的
        - 两同：
            - 方法名相同
            - 参数列表相同
        - 两小：
            - 派生类方法的返回值类型小于或等于超类方法的
                - void和基本类型时，必须相等
                - 引用类型时，小于或等于
            - 派生类方法抛出的异常小于或等于超类方法的----------API时讲
        - 一大：
            - 派生类方法的访问权限大于或等于超类方法的----------明天讲
3. 重写与重载的区别：----------常见面试题
    - 重写：发生在父子类中，方法名相同，参数列表相同
    - 重载：发生在同一类中，方法名相同，参数列表不同

笔记：

1. 向上造型：----------代码复用
    - 超类型的引用指向派生类的对象

- 能点出来什么，看引用的类型----------这是规定，记住就OK

> 何时向上造型：
>
> - 多种角色能干的事都一样的时候，可以将多种角色统一造型到超类数组中，实现代码复用
>
> eg: 学生/老师/医生都是输出名字+问好------干的事都一样，
>
> 就可以将学生/老师/医生统一造型到Person数组中，这样一个for即可-------代码复用

```
public class UploadDemo {
    public static void main(String[] args) {
        Aoo o1 = new Aoo();
        o1.a = 1;
        o1.show();
        //o1.b = 2;  //编译错误
        //o1.test(); //编译错误，超类不能访问派生类的

        Boo o2 = new Boo();
        o2.b = 1;
        o2.test();
        o2.a = 2;  //正确
        o2.show(); //正确，派生类可以访问超类的

        Aoo o3 = new Boo(); //向上造型
        o3.a = 1;
        o3.show();
        //o3.b = 2;  //编译错误
        //o3.test(); //编译错误，能点出来什么，看引用的类型
    }
}

class Aoo{
    int a;
    void show(){
    }
}
class Boo extends Aoo{
    int b;
    void test(){
    }
}
```

2. 方法的重写(override/overriding)：重新写、覆盖

- 发生在父子类中，方法名相同，参数列表相同
- 重写方法被调用时，看对象的类型----------这是规定，记住就OK

```
class 餐馆{
    void 做餐(){ 做中餐 }
}
//1)我还是想做中餐------------------不需要重写
class Aoo extends 餐馆{
}
//2)我想改做西餐-------------------需要重写
class Aoo extends 餐馆{
```

```
    void 做餐(){ 做西餐 }
}
//3)我想在中餐基础之上加入西餐--------需要重写(先super中餐，再加入西餐)
class Aoo extends 餐馆{
    void 做餐(){
        super.做餐();
        做西餐
    }
}
```

- 重写需遵循"两同两小一大"原则：----------了解，一般都是一模一样的
  - 两同：
    - 方法名相同
    - 参数列表相同
  - 两小：
    - 派生类方法的返回值类型小于或等于超类方法的
      - void和基本类型时，必须相等
      - 引用类型时，小于或等于

```
//超类大，派生类小----------爸爸大，儿子小
class Coo{
    void show(){}
    double test(){ return 0.0; }
    Student say(){ return null; }
    Person sayHi(){ return null; }
}
class Doo extends Coo{
    //int show(){ return 1; } //编译错误，void时必须相等
    //int test(){ return 0; } //编译错误，基本类型时必须相等
    //Person say(){ return null; } //编译错误，引用类型时必须小于或等于
    Student sayHi(){ return null; } //正确，Student小于Person
}
```

    - 派生类方法抛出的异常小于或等于超类方法的----------API时讲
  - 一大：
    - 派生类方法的访问权限大于或等于超类方法的----------明天讲

3. 重写与重载的区别：----------常见面试题
   - 重写：发生在父子类中，方法名相同，参数列表相同
   - 重载：发生在同一类中，方法名相同，参数列表不同

补充：

1. 继承要符合is(是)的关系
2. 超类的意义：
   - 封装共有的属性和行为----------实现代码复用
   - 为所有派生类提供了统一的类型----------向上造型(实现代码复用)
3. 明日单词：

```
```

```
1)override:重写
2)package:包
3)import:导入
4)public:公开的
5)protected:受保护的
6)private:私有的
7)card:卡
8)id:号码
9)password/pwd:密码
10)balance:余额
11)pay:支付
12)money:金额
13)check:检查
14)static:静态的
15)image:图片
16)icon:图标
17)get:获取
18)status:状态
```

练习：----------ooday05包中

```
1.创建Person类，包含：       //------------如下的类必须分在不同的文件中写
  1)成员变量:name,age,address
  2)构造方法:Person(3个参数){ 赋值 }
  3)方法:sayHi(){ 输出3个数据 }
2.创建学生类Student，继承Person，包含:
  1)成员变量:学号stuId(String)
  2)构造方法:Student(4个参数){ super调超类3参构造、赋值stuId }
  3)方法:重写sayHi(){ 输出4个数据 }
3.创建老师类Teacher，继承Person，包含:
  1)成员变量:工资salary(double)
  2)构造方法:Teacher(4个参数){ super调超类3参构造、赋值salary }
  3)方法:重写sayHi(){ 输出4个数据 }
4.创建医生类Doctor，继承Person，包含:
  1)成员变量:职称level(String)
  2)构造方法:Doctor(4个参数){ super调超类3参构造、赋值level }
5.创建测试类Test，main中:
  1)创建Person数组ps,包含5个元素,给元素赋值(学生/老师/医生),遍历输出名字并问好
```