

面向对象第一天：

潜艇游戏第一天：

1. 创建6个类，创建World类并测试

潜艇游戏需求：

1. 所参与的角色：战舰、炸弹、侦察潜艇、鱼雷潜艇、水雷潜艇、水雷
2. 角色间的关系：
 - 战舰发射炸弹
 - 炸弹打潜艇(侦察潜艇、鱼雷潜艇、水雷潜艇)，若打中了：
 - 炸弹去死、潜艇去死
 - 打掉侦察潜艇，玩家得10分
 - 打掉鱼雷潜艇，玩家得40分
 - 打掉水雷潜艇，战舰得1条命
 - 水雷潜艇发射水雷
 - 水雷打战舰，若打中了：
 - 水雷去死
 - 战舰减1条命(命数为0，则游戏结束)

笔记：

1. 什么是类？什么是对象？
 - 现实生活是由很多很多对象组成的，基于对象抽象出了类
 - 对象：软件中真实存在的单个的个体/东西
 - 类：类型/类别，代表一类个体
 - 类是对象的模板/模子，对象是类的具体实例
 - 类中可以包含：
 - 对象的属性/特征/数据：成员变量
 - 对象的行为/动作/功能：方法
 - 一个类可以创建多个对象
2. 如何创建类？如何创建对象？如何访问成员？

```
package ooday01;

//学生类
public class Student {
    //成员变量
    String name;
    int age;
    String address;

    //方法
    void study() {
```

```

        System.out.println(name + "在学习...");
    }

    void sayHi() {
        System.out.println("大家好，我叫" + name + "，今年" + age + "岁了，家住"
+ address);
    }
}

```

```

package ooday01;

//学生类的测试类
public class StudentTest {
    public static void main(String[] args) {
        //创建一个学生对象
        Student zs = new Student();

        //给成员变量赋值
        zs.name = "zhangsan";
        zs.age = 25;
        zs.address = "河北廊坊";

        //调用方法
        zs.study();
        zs.sayHi();

        Student ww = new Student();
        //1)创建了一个学生对象
        //2)给所有成员变量赋默认值
        ww.study(); //null
        ww.sayHi(); //null 0 null
    }
}

```

3. 方法的签名：方法名+参数列表

4. 方法的重载(overload)：大大方便用户的调用

- 发生在同一类中，方法名相同，参数列表不同
- 编译器在编译时会根据方法的签名自动绑定调用方法

```

package ooday01;

class Aoo {
    void show() {}
    void show(String name) {}
    void show(int age) {}
    void show(String name, int age) {}
    void show(int age, String name) {}

    //int show(){ return 1; } //编译错误，重载与返回值类型无关
    //void show(String address){} //编译错误，重载与形式参数名称无关
}

```

```

package ooday01;

//方法重载的演示
public class OverloadDemo {
    public static void main(String[] args) {
        Aoo o = new Aoo();
        o.show();
        o.show(25);
        o.show("zhangsan");
        o.show("zhangsan", 25);
        o.show(25, "zhangsan");
    }
}

```

补充:

1. OO: 面向对象 (Object Oriented)

OOA: 面向对象分析 (Analysis)

OOD: 面向对象设计 (Design)

OOP: 面向对象编程 (Programming)

2. 高质量的代码:

- 复用性好
- 扩展性好
- 维护性好
- 可移植性好
- 健壮性好
- 可读性好
- 效率高
-

3. 类: 是一种引用数据类型(是我们自己创造的一种数据类型)

4.

```

//          引用
//数据类型 引用类型变量 指向      对象
Student    zs      = new Student();

```

5. new对象时会给所有成员变量赋默认值, 规则如下:

```

byte, short, int, long, char: 0
float, double: 0.0
boolean: false
引用数据类型: null

```

6. 明日单词:

- 1) this: 这个
- 2) pointer: 指针
- 3) random: 随机

