面向对象第二天：

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

回顾：

1. 什么是类？什么是对象？
2. 如何创建类？如何创建对象？如何访问成员？
3. 方法的重载(overload/overloading)：
    - 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

精华笔记：

1. 构造方法：构造函数、构造器、构建器----------复用给成员变量赋初值的代码
    - 作用：给成员变量赋初值
    - 与类同名，没有返回值类型(连void都没有)
    - 在创建(new)对象时被自动调用
    - 若自己不写构造方法，编译器默认提供一个无参构造方法，若自己写了构造方法，则不再默认提供
    - 构造方法可以重载
2. this：指代当前对象，哪个对象调用方法它指的就是哪个对象

    只能用在方法中，方法中访问成员变量之前默认有个this.

    this的用法：
    - this.成员变量名----------访问成员变量

        > 当成员变量与局部变量同名时，若想访问成员变量，则this不能省略

    - this.方法名()----------调用方法(一般不用)
    - this()----------调用构造方法(一般不用)
3. null：表示空，没有指向任何对象。
    - 若引用的值为null，则该引用不能进行任何点操作了，若操作则发生NullPointerException空指针异常。

笔记：

1. 构造方法：构造函数、构造器、构建器----------复用给成员变量赋初值的代码

- 作用：给成员变量赋初值
- 与类同名，没有返回值类型(连void都没有)
- 在创建(new)对象时被自动调用
- 若自己不写构造方法，编译器默认提供一个无参构造方法，若自己写了构造方法，则不再默认提供
- 构造方法可以重载

```
public class Student {
    String name; //成员变量(整个类中)
    int age;
    String address;
    //给成员变量赋初值
    Student(String name,int age,String address){ //局部变量(当前方法中)
        this.name = name;
        this.age = age;
        this.address = address;
    }

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

public class ConsDemo {
    public static void main(String[] args) {
        //Student zs = new Student(); //编译错误，Student类没有无参构造方法
        Student zs = new Student("zhangsan",25,"LF");
        zs.sayHi();

        Student ls = new Student("lisi",24,"JMS");
        ls.sayHi();
    }
}
```

2. this：指代当前对象，哪个对象调用方法它指的就是哪个对象

   只能用在方法中，方法中访问成员变量之前默认有个this.

   this的用法：

   - this.成员变量名----------访问成员变量

     > 当成员变量与局部变量同名时，若想访问成员变量，则this不能省略

   - this.方法名()----------调用方法(一般不用)
   - this()----------调用构造方法(一般不用)

3. null：表示空，没有指向任何对象。
   - 若引用的值为null，则该引用不能进行任何点操作了，若操作则发生NullPointerException空指针异常。

补充：

1. java规定：成员变量和局部变量是可以同名的，使用的时候默认采取的是就近原则
2. 构造方法到底要不要参数，要看对象的数据能不能写死

   如果对象的数据都一样，意味着可以写死，就不需要传参。

   如果对象的数据都不一样，意味着不能写死，那就需要传参。
3. 内存管理：由JVM来管理的
   - 堆：new出来的对象(包括成员变量)
   - 栈：局部变量(包括方法的参数)
   - 方法区：----------面向对象第5天再讨论
4. 明日单词：

```
1)reference:引用
2)extends:继承
3)super:超级
4)Sea:海洋
5)object:对象
```

null只是相对于引用类型而言，它与基本类型之间一点关系都没有

```
Student zs = null;
int a = null; //编译错误
```

```
class Aoo{
    int a; //成员变量
    void show(int b){ //局部变量
        int c = 5;
    }
}

new Aoo(); //--------堆中
a //-----------------堆中
b //-----------------栈中
c //-----------------栈中
```

```
class Battleship{
    Battleship(){ }
}
class ObserveSubmarine{
    ObserveSubmarine(){ }
}
class TorpedoSubmarine{
    TorpedoSubmarine(){ }
}
class MineSubmarine{
    MineSubmarine(){ }
}
class Mine{
    Mine(int x,int y){ }
}
class Bomb{
    Bomb(int x,int y){ }
}
```