

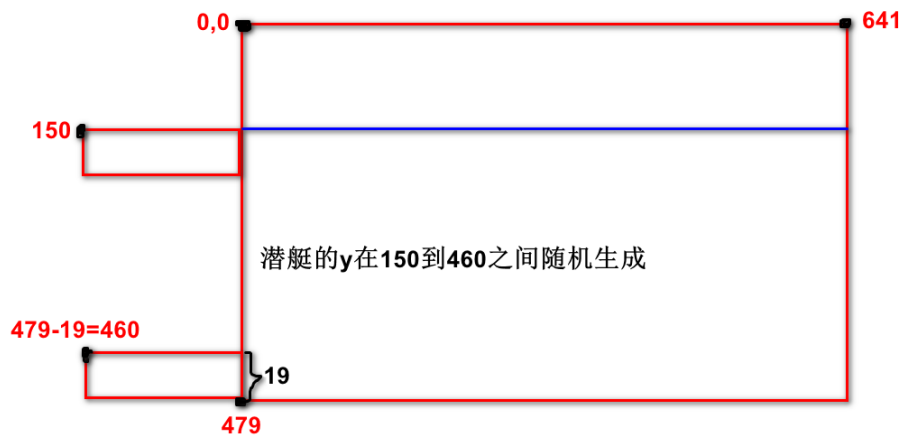
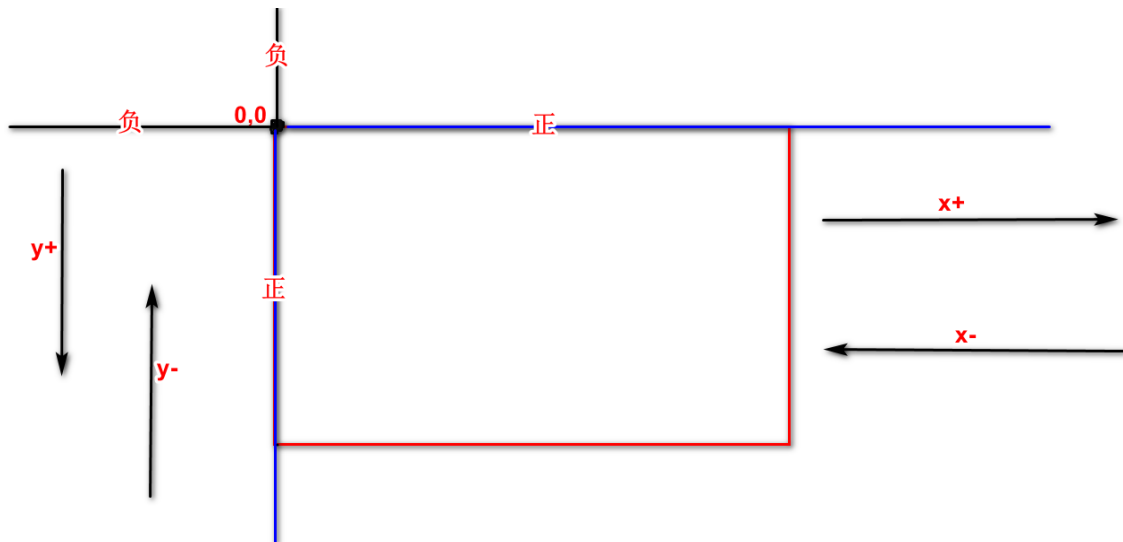
面向对象第2天:

潜艇游戏第1天:

1. 创建6个类，创建World类并测试

潜艇游戏第2天:

1. 给6个类添加构造方法，并测试



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始值代码
 - 作用：给成员变量赋初始值
 - 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，则编译器默认提供一个无参构造方法；若自己写了，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 (一般省略)

不能省略的情况: 当成员变量和局部变量同名时, 若想访问成员变量, 则this不能省略

- this.方法名(): 调用方法 (一般省略)

```
package ooday02;

//学生类
public class Student {
    String name; //成员变量(整个类中)
    int age;
    String address;

    //构造方法
    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;
    }

    //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 ooday02;

//构造方法的演示
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");
        Student ls = new Student("lisi", 24, "JMS");
        zs.sayHi();
        ls.sayHi();
    }
}
```

- this(): 调用重载的构造方法 (一般不用)

```

Student(){
    this("无名氏",1,"未知"); //调用构造方法（了解即可）
}

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;
}

Student ww=new Student();
ww.sayHi();
//大家好，我叫无名氏，今年1岁了，家住未知

```

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（a.speed），若操作则发生NullPointerException空指针异常。

补充：

1. 给成员变量赋初始值的代码写在构造方法中，其它业务代码写在普通方法中

2. 成员变量：写在类中方法外，作用范围为整个类

局部变量：写在方法中(包括方法的参数)，作用范围为当前方法

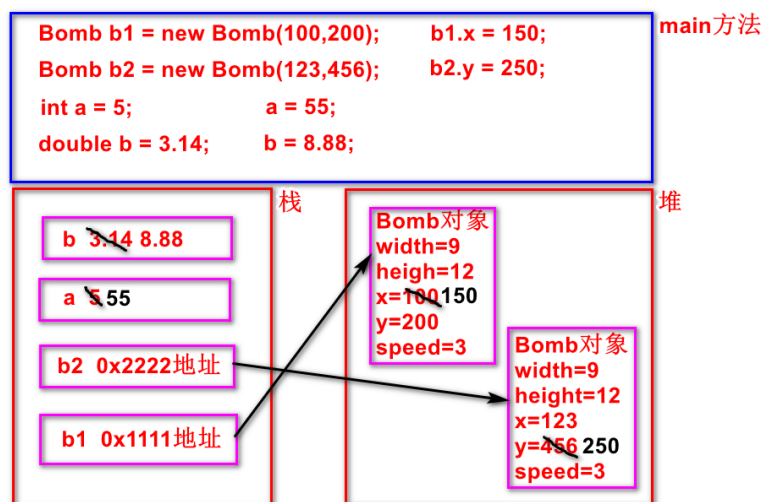
3. java规定：成员变量与局部变量是可以同名的，使用的时候默认采取的是就近原则

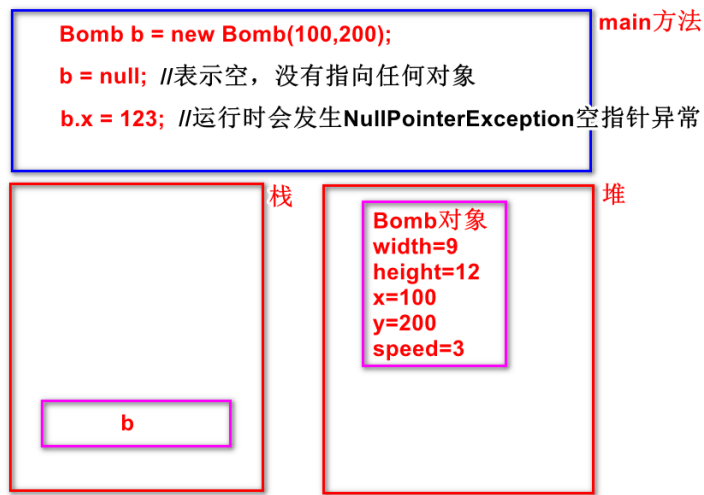
当成员变量和局部变量同名时，若想访问成员变量，则this不能省略，要用this.成员变量=局部变量

4. 显示：?? related problems，表示有关联错误，不用管，找到编译错误位置改好就可以了

5. 内存管理：由JVM来管理

- 堆：存储new出来的对象(包括成员变量)
- 栈：存储局部变量(包括方法的参数)
- 方法区：





6. 基本数据类型变量(变量)中装的是具体的数, 引用数据类型变量(引用)中装的是对象

7. 异常:

- `ArrayIndexOutOfBoundsException`: 数组下标越界异常
- `NullPointerException`: 空指针异常

8. 明日单词:

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

9. 变量的命名: 只能包含字母、数字、_、\$

一般都是纯字母

数字、_、\$, 只在特定情况下使用

10. 晚课:

- 问: `new Battleship()`后, `width`, `height`的值为什么都是0?

```
class Battleship {
    int width; //成员变量
    int height;

    Battleship() {
        int width = 58; //定义一个局部变量并赋值, 并没有改变成员变量
        int height = 80;
    }
}
```

•

```
class Test {
    public static void main(String[] args) {
        Aoo o = new Aoo(); //调用编译器默认提供的无参构造方法
        System.out.println(o.a); //0
        o.Aoo(); //调用Aoo类中的Aoo()方法
        System.out.println(o.a); //5
    }
}
```

```

class Aoo {
    int a;

    //这不是构造方法，而是一个普通方法(需要通过o.Aoo()来访问)
    void Aoo() {
        a = 5;
    }
}

```

- 堆栈存什么?

存在哪?堆中存new出来的对象(包括成员变量)、栈中存局部变量(包括方法参数)

装什么?引用数据类型变量中存的是地址，基本数据类型变量中存的是值

```

class Aoo {
    Student zs = new Student(); //堆中(装地址)
    int a = 5; //堆中(装值5)

    void test() {
        Student ls = new Student(); //栈中(装地址)
        int b = 6; //栈中(装值6)
    }
}

```

- 成员变量是有默认值的，局部变量是没有默认值的(必须自己赋值)

```

public class ooday02 {
    public void main(String[] args) {
        Aoo o = new Aoo(); //a=0，默认构造函数，赋为默认值0
        o.show(5); //b=5，外部传参数5
    }

    class Aoo {
        int a;

        void show(int b) {
            int c; //未赋值，无法输出
            System.out.println(a); //0
            System.out.println(b); //5
            System.out.println(c); //编译错误
        }
    }
}

```