

面向对象第三天：

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，设计6个类继承SeaObject
3. 给SeaObject设计两个构造方法，6个派生类分别调用

笔记：

1. 引用类型数组：

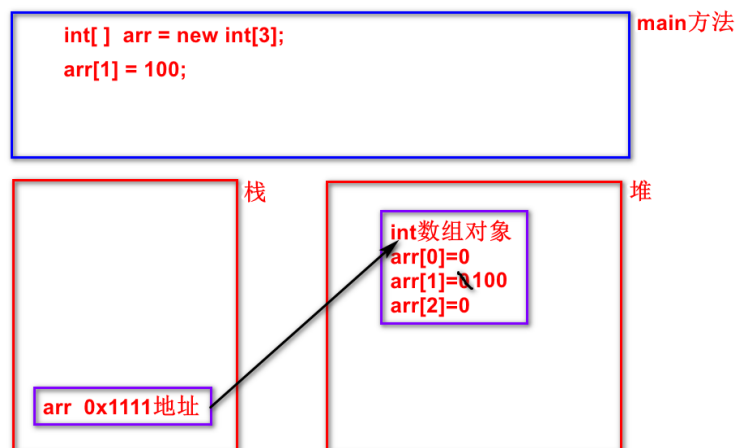
- 给引用类型数组的元素赋值，需要先new那个数组元素
- 若想访问对象的属性或调用对象的方法，需要通过数组元素（students[i].name）

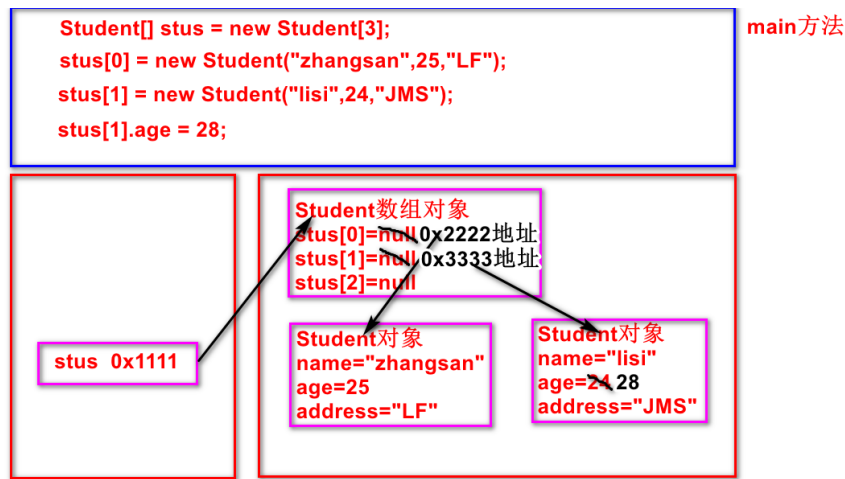
//注意：对于引用数组，必须给所有元素赋值，否则访问时会发生NullPointerException空指针异常

```
Student[] stus = new Student[3]; //创建Student数组对象
stus[0] = new Student("zhangsan", 25, "LF"); //创建Student对象
stus[1] = new Student("lisi", 24, "JMS");
stus[2] = new Student("wangwu", 26, "SD");
System.out.println(stus[0].name); //输出第1个学生的名字
stus[1].age = 28; //修改第2个学生的年龄为28
stus[2].sayHi(); //第3个学生跟大家问好

for (int i = 0; i < stus.length; i++) { //遍历学生数组
    System.out.println(stus[i].name); //输出每个学生的名字
    stus[i].sayHi(); //每个学生跟大家问好
}
```

- 内存图：





2. 继承:

- 作用: 代码复用
- 通过extends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
- 派生类既可以访问派生类的, 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, 但一个派生类只能继承一个超类 (java只能单继承)
- 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, 若没有调用超类构造方法, 则默认super()调用超类无参构造方法

```

package ooday03;

public class SuperDemo {
    public static void main(String[] args) {
        Boo o = new Boo();
    }
}

class Aoo {
    Aoo() {
        System.out.println("超类构造方法");
    }
}

class Boo extends Aoo {
    Boo() {
        super(); //没有调用超类构造方法, 默认调用超类的无参构造方法
        System.out.println("派生类构造方法");
    }
}

```

- 在派生类的构造方法中, 若自己调用了超类构造方法, 则不再默认提供

注意: 若使用super()调用超类构造方法, 则必须位于派生类构造方法的第一行

```

class Coo {
    Coo(int a) {

```

```

    }
}

class Doo extends Coo {
    Doo() {
        super(5); //自己调用了超类构造方法，则不会再调用默认的super()
    }
    /*
    //如下代码为默认的
    Doo() {
        super();
    }
    */
}

```

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

练习:

```

/*
练习:写在ooday04包中(明天会用)
如下的类必须分在不同的文件中写
1.创建Person类, 包含:
    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.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 }
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)创建学生数组stus, 包含3个元素, 给元素赋值, 遍历输出名字并问好
    2)创建老师数组tes, 包含3个元素, 给元素赋值, 遍历输出名字并问好
    3)创建医生数组docs, 包含2个元素, 给元素赋值, 遍历输出名字并问好
*/

```

```

//Person.java
package ooday04;

public class Person {
    String name;
    int age;
    String address;

    Person(String name, int age, String address) {
        this.name = name;
    }
}

```

```

        this.age = age;
        this.address = address;
    }

    void sayHi() {
        System.out.println("我的名字是" + name + ", 年龄是" + age + ", 地址是" +
address);
    }
}

```

```

//Student.java
package ooday04;

public class Student extends Person {
    String stuID;

    Student(String name, int age, String address, String stuID) {
        super(name, age, address);
        this.stuID = stuID;
    }
}

```

```

//Teacher.java
package ooday04;

public class Teacher extends Person {
    double salary;

    Teacher(String name, int age, String address, double salary) {
        super(name, age, address);
        this.salary = salary;
    }
}

```

```

//Doctor.java
package ooday04;

public class Doctor extends Person {
    String level;

    Doctor(String name, int age, String address, String level) {
        super(name, age, address);
        this.level = level;
    }
}

```

```

//Test.java
package ooday04;

public class Test {
    public static void main(String[] args) {
        Student[] stus = new Student[3];
        stus[0] = new Student("张三", 21, "广州", "202201");
        stus[1] = new Student("李四", 22, "天河", "202202");
        stus[2] = new Student("王五", 23, "珠江新城", "202203");
    }
}

```

```
for (int i = 0; i < stus.length; i++) {
    System.out.println(stus[i].name);
    stus[i].sayHi();
}

Teacher[] tes = new Teacher[3];
tes[0] = new Teacher("张三1", 31, "广州1", 5000.0);
tes[1] = new Teacher("李四1", 32, "天河1", 6000.0);
tes[2] = new Teacher("王五1", 33, "珠江新城1", 7000.0);
for (int i = 0; i < tes.length; i++) {
    System.out.println(tes[i].name);
    tes[i].sayHi();
}

Doctor[] docs = new Doctor[2];
docs[0] = new Doctor("张三2", 41, "广州2", "高级");
docs[1] = new Doctor("李四2", 42, "天河2", "副高");
for (int i = 0; i < docs.length; i++) {
    System.out.println(docs[i].name);
    docs[i].sayHi();
}
}
```

补充:

1. 明日单词:

- 1)up: 向上
- 2)load: 加载
- 3)animal: 动物
- 4)tiger: 老虎
- 5)override: 重写