Shoot射击游戏第一天：

1. 创建6个对象类，并创建World类测试

Shoot射击游戏第二天：

1. 给6个对象类添加构造方法，并测试

Shoot射击游戏第三天：

1. 设计小敌机数组、大敌机数组、小蜜蜂数组、子弹数组，并测试

回顾：

1. 方法的签名：方法名+参数列表
2. 方法的重载(Overload)：

1) 发生在一个类中，方法名相同，参数列表不同，方法体不同

2) 编译器在编译时会根据方法的签名自动绑定方法

1. 构造方法：

给成员变量赋初值，与类同名、没有返回值类型，new对象时被自动调用

若自己不写构造则默认无参构造，若自己写了构造则不再默认提供，可以重载

1. 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，访问成员变量之前默认有个this.

-----成员变量与局部变量同名时，this不能省略

this用法:

1) this.成员变量名----------访问成员变量

2) this.方法名()-------------调用方法

3) this()-------------------调用构造方法

正课：

1. 内存管理: 由JVM管理------------先知道每块区域中存什么，具体细节面向对象最后一天时讲

1) 堆：存储new出来的对象(包括成员变量)

2) 栈：局部变量(包括方法的参数)

3) 方法区：.class字节码文件(包括方法)--------后面讲static时用

1. null：表示空，没有指向任何对象

若引用的值为null，则该引用不能再进行任何操作了，若操作则NullPointerException空指针异常

1. 引用类型变量画等号:

1) 指向同一个对象(数据只有一份)

2) 通过一个引用对数据的修改会影响另一个引用对数据的访问

eg: 房子钥匙

基本类型变量画等号:

1) 赋值(数据是两份)

2) 对一个变量的修改不影响另一个变量的访问

eg: 身份证复印件

1. 引用类型数组：

1) Student[ ] stus = new Student[3];

stus[0] = new Student(“zhangsan”, 25, “LF”);

stus[1] = new Student(“lisi”, 26, “JMS”);

stus[2] = new Student(“wangwu”, 28, “SD”);

stus[0].age = 28; //给第一个学生的年龄赋值为28

System.out.println(stus[1].name); //输出第二个学生的名字

stus[2].sayHi();

for(int i=0; i<stus.length; i++){

System.out.println( stus[i].name ); //stus[i]代表每个学生

stus[i].sayHi();

}

2) Student[ ] stus = new Student[ ] {

new Student(“zhangsan”, 25, “LF”) ,

new Student(“lisi”, 26, “JMS”) ,

new Student(“wangwu”, 28, “SD”)

};

任务:

1. 射击游戏今天的项目代码--------至少2次
2. 画引用类型数组内存图

晚上7点晚课-------------小结+答疑

//声明Student型数组stus，包含3个元素，

//每个元素都是Student类型，默认值为null

Student[ ] stus = new Student[3];

stus[0], stus[1], stus[2]

Student[ ] stus = new Student[3]; //null, null, null

2秒钟5个敌人

1分钟150个敌人

10分钟1500个敌人

int[ ] arr = new int[ ] {

1,

4,

7

} ;

Student[ ] stus = new Student[ ] {

new Student(“zhangsan”, 25, “LF”) ,

new Student(“lisi”, 26, “JMS”) ,

new Student(“wangwu”, 28, “SD”)

};

int[ ] arr = new int[3];

arr[0] = 100; //arr[0]为int类型

Student[ ] stus = new Student[3];

stus[0] = new Student(“zhangsan”,25,“LF”); //stus[0]为Student类型

stus[1] = new Student(“lisi”,26,“JMS”);

stus[0].age = 28; //给第一学生的年龄修改为28

System.out.println(stus[1].name); //输出第二个学生的名字

stus[0].sayHi();

stus[0].name/age/address/study()/sayHi();

Airplane[ ] as = new Airplane[10];

as[0] = new Airplane(); //as[0]为Airplane类型

Bee[ ] bs = new Bee[20];

bs[0] = new Bee(); //bs[0]为Bee类型

//声明int型数组arr，包含3个元素，

//每个元素都是int类型，默认值为0

int[ ] arr = new int[3];

//声明Student型数组stus，包含3个元素，

//每个元素都是Student类型，默认值为null

Student[ ] stus = new Student[3];

//声明Airplane型数组as，包含10个元素，

//每个元素都是Airplane类型，默认值为null

Airplane[ ] as = new Airplane[10];

//声明Bee型数组bs，包含20个元素，

//每个元素都是Bee型，默认值为null

Bee[ ] bs = new Bee[20];

Student[ ] stus = new Student[3]; //创建Student数组对象

stus[0] = new Student(); //创建Student对象

int [ ] arr = new int[3] ; //基本类型数组

Student [ ] stus = new Student[3] ; //引用类型数组

房子: 对象(数据)

房子钥匙: 引用

配一把钥匙: 引用

身份证原件: 数据

身份证复印件: 数据

Student zs = new Student();

zs.age = 25;

zs = null;

int a = null; //编译错误

成员变量: 类中，方法外

局部变量: 方法中

void println(){

}

void printlnInt(){

}

void printlnDouble(){

}

void printlnChar(){

}

void printlnString(){

}

System.out.println();

System.out.printlnInt(67);

System.out.printlnDouble(3.14);

System.out.printlnChar(‘h’);

System.out.printlnString(“Hello”);

API

定义、初始化、访问、遍历、复制、排序

Student[ ] stus;

int[ ] arr;

double[ ] dou;

char[ ] cs;