Shoot射击游戏第一天：

1. 创建6个对象类，并创建World类并测试

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

Shoot射击游戏第四天：

1. 设计FlyingObject超类，6个对象类继承超类
2. 在超类中设计2个构造方法，6个对象类分别调用
3. 画窗口

Shoot射击游戏第五天：

1. 将小敌机数组、大敌机数组、小蜜蜂数组组合为FlyingObject数组，并测试
2. 在6个派生类中重写step方法

Shoot射击游戏第六天：

1. 给类中成员添加访问控制修饰符
2. 设计Images工具类给对象准备图片

回顾：

1. 向上造型:

1) 超类型的引用指向派生类的对象

2) 能点出来什么，看引用的类型

1. 方法的重写:

1) 发生在父子类中，方法名相同，参数列表相同，方法体不同

2) 重写方法被调用，看对象的类型

1. 重写与重载的区别:

1) 重写:

1.1 发生在父子类中，方法名相同，参数列表相同，方法体不同

1.2 “运行期绑定”，看对象

2) 重载:

2.1发生在一个类中，方法名相同，参数列表不同，方法体不同

2.2“编译期绑定”，看参数/引用

正课：

1. package:包

1) 避免类的命名冲突(同包中的类名不冲突就可以)

2) 类的全称: 包名.类名

3) 包名有层次结构

4) 建议: 包名所有字母都小写

import:

1) 同一包中的类可以直接访问，不同包中的类不能直接访问，若想访问只有如下两种方式:

2.1）先import类再访问类

2.1) 类的全称----太繁琐不建议

1. 访问控制修饰符: 保护数据的安全

1) public: 公开的，任何类

2) private: 私有的，本类

3) protected: 受保护的，本类、派生类、同包类

4) 默认: 什么也不写，本类、同包类

说明:

1) 类的访问权限只能是public或默认的

2) 类中成员的访问权限如上4种都可以

3. static: 静态的

1) 静态变量:

1.1) 由static修饰

1.2) 属于类，存储在方法区中，只有一份

1.3) 常常通过类名点来访问

1.4) 何时用: 所有对象所共享的数据( 图片、音频、视频等 )

2) 静态方法:

2.1) 由static修饰

2.2) 属于类，存储在方法区中，只有一份

2.3) 常常通过类名点来访问

2.4) 静态方法中没有隐式的this传递，所以在静态方法中不能直接访问实例成员

2.5) 何时用: 方法的操作与对象无关

3) 静态块:

3.1) 由static修饰

3.2) 属于类，在类被加载期间自动执行，因为类只被加载一次，所以静态块也只执行一次

3.3) 何时用: 加载/初始化静态资源( 图片、音频、视频等 )

BufferedImage image = 是读取出来的 (不是new出来的)

具体读图片的代码-----不要求掌握

Int

boolean

Images类封装与图片相关的操作

工具类:

1. Math类封装与数学相关的操作
2. Arrays类封装与数组相关的操作

double d = Math.sqrt(25); //求平方根

假设sqrt()不是静态的:

无论m1,m2,m3,…,m100中的哪一个对象去sqrt(25)，最终的结果都是一样的（5）

说明sqrt()的操作与对象无关

Math m1 = new Math();

double d = m1.sqrt(25);--------------5.0

Math m2 = new Math();

double d = m2.sqrt(25);--------------5.0

Math m3 = new Math();

double d = m3.sqrt(25);--------------5.0

Scanner scan = new Scanner(System.in);

int a = scan.nextInt();

double d = scan.nextDouble();

double c = Math.random(); //生成随机数

double d = Math.sqrt(25); //求平方根

Arrays.sort(arr);

王牌对王牌第六季第一集.avi

桥边姑娘.mp3

小敌机---------有图片(只有一份)

堆：new出来的对象(包括实例变量)

栈：局部变量(包括方法的参数)

方法区：.class字节码文件(包括方法、静态变量)

成员变量分两种:

1. 实例变量: 没有static修饰，属于对象的，存储在堆中，有几个对象就有几份，通过对象点来访问
2. 静态变量: 有static修饰，属于类的，存储在方法区中，只有一份，通过类名点来访问

class Aoo{

int a; //实例变量，堆，

static int b; //静态变量，方法区，

}

/\*

class Card{

private String cardId; //卡号

private String cardPwd; //密码

private double balance; //余额

public boolean payMoney(double money){ //支付金额

if(balance>=money){

balance-=money;

return true;

}else{

return false;

}

}

public boolean checkPwd(String pwd){ //验证密码

if(密码正确){

return true;

}else{

return false;

}

}

}

\*/

/\*

\* import java.util.Scanner;

\* Scanner scan = new Scanner(System.in);

\* int a = scan.nextInt();

\* double d =scan.nextDouble();

\*

\* package java.util;

\* class Scanner{

\* Scanner(Stream s){

\* }

\* int nextInt(){

\* }

\* double nextDouble(){

\* }

\* }

\*

\*/

/\*

\* 民生银行----java

\* A公司:

\* package com.a.hgy.uio;

\* class Aoo{

\* }

\*

\* B公司:

\* package com.b.hgy.uio;

\* class Aoo{

\* }

\*

\* A项目-----有M一个模块(aproject)

\* B项目中也用到M模块(bproject)

\*

\*/

/\*

域名反写 . 项目名称 . 模块名称 . 类名

cn.tedu . aproject . stumamager .

cn.tedu . aproject . teachmanager .

cn.tedu . bproject

com.taobao

com.baidu

\*/

数据(成员变量)私有化(private)

行为(方法)公开化(public)

class Card{

String cardId;

String cardPwd;

double balance;

}

package a;

import b.Coo;

class Aoo{

Coo o = new Coo();

b.Coo oo = new b.Coo(); //全称

Boo o = new Boo();

//Coo oo = new Coo(); //编译错误

}

class Boo{

}

package b;

class Coo{

}

包的建议命名:

域名反写 . 项目名称 . 模块名称 . 类名

package a.b.c.d;

class Aoo{. //a.b.c.d.Aoo

}

package a;

class Aoo{ //a.Aoo

}

package b;

class Aoo{ //b.Aoo

}

一个项目包含100个类

100个类名-----不能同名

重载看Aoo，重写看Boo

Aoo o = new Boo();