类与对象基础概念

package b;

public class 类与对象 {

/\*

\* java文件的格式:

\* 权限 class 类名{

\* //类体

\* }

\* 说明:

\* 1.权限有四种: public protected private 默认.

\* 2.类名通常必须为当前java文件名.

\*

\* 方法: 由一段代码组成,完成特定的功能.

\* 方法格式:

\* 权限 [static/native/abstract等] 返回值类型 方法名([参数类型 参数名,参数类型 参数名,..]){

\* 方法体

\* }

\*

\* 返回值: 由当前方法内向方法外传递数据的一种方式

\* 返回值类型: void 数据类型

\* void表示不返回且不能返回任何数据

\* 数据类型表示必须返回指定类型的数据

\*

\* 参数: 由当前方法外向方法内传递数据的一种方式

\* 参数处理原则: 必须按照参数的个数,顺序,类型传入相应数据.

\*

\*

\* Java对象:

\* java属于面向对象的语言,操作操作都要围绕对象进行.

\* 问题1: 哪些事物可以作为Java对象

\* 一切皆对象,万物皆对象.

\*

\* 问题2: 如何将现实事物抽取到java 程序中

\* 将现实事物的静态特征抽象为java对象的属性,将现实事物的动态特征抽象为java对象的方法.

\*

\* 对象: 仙人球

\* 属性: 绿色 圆形 刺

\* 功能: 扎人 观赏

\*

\* 属性: 实质就是全局变量

\* 属性类型 属性名=值; 属性类型 属性名;

\*

\* 功能: 即方法

\*

\* 问题3: 如何模拟/创建Java对象

\* 思路: java文件记录了对象的信息[静态特征和动态特征],java文件就是创建对象的设计图纸.

\* [类与对象的关系: 类是创建对象的设计图纸,对象是设计图纸的实例/体现]

\*

\* new 设计图纸名(); //设计图纸名就是文件名,文件名与类型相同: new 类名()

\*

\* 操作对象:

\* 对象.属性名;

\* 对象.方法名()

\*

\*

\*

\* 方法分类: main方法 普通方法 构造方法

\* 调用方法:

\* 直接调用:

\* 格式: 方法名(....)

\* 要求: 被调用的方法必须与当前方法在同一个类中

\*

\* 对象调用:

\* 格式: 对象名.方法名(....);

\* 要求: 必须先创建对象,才能使用

\*

\* 类名调用:

\* 格式: 类名.方法名(...)

\* 要求: 被调用的方法必须被static修饰

\*

\*

\* 导包:

\* import 包名1.包名2....类名;

\* 当指定的类与当前类不在同一个包中时,需要执行导包指令.

\*/

/\*

\* public为权限

\* static为可选内容

\* void为返回值类型

\* main为方法名

\* String[] 为参数类型

\* args 为参数名

\*/

public static void main(String[] args) {

}

}

仙人球对象

package b;

public class 仙人球 {

/\*

\* 对象: 仙人球

\* 属性: 绿色 圆形 刺

\* 功能: 扎人 观赏

\*/

String color="绿色";

String type="圆形";

public void 扎人() {

System.out.println("扎人了....");

}

public void 观赏() {

System.out.println("正在观赏.....");

}

}

调用

package b;

public class Main {

public static void main(String[] args) {

仙人球 obj=new 仙人球();

obj.扎人();

obj.观赏();

仙人球 obj1=new 仙人球();

obj1.扎人();

obj1.观赏();

System.out.println(obj);

System.out.println(obj1);

}

}