Java 学习总结：

1. 包的学习（package）

在开头出声明为package cn.skd.dztx.zhaohong;

在命令输入框内命令为 javac –d . Test.java

一旦定义了包，此时里面的类的名字必须是包括包名了，不再是简单的类名，如cn.skd.dztx.zhaohong.Test.java 也就是说，一个类的全名应该是 包名+类名。

打包需要使用package命令。

包名的命名规定：1.名字必须为小写字母；2. 域名倒过来写。

1. 包的导入：

关键词：import

偷懒的方法时直接 import cn.\* 利用\*号直接代表了所有

如果没有导入另外的包，且需要使用其中的类，此时需要使用类的全名（包名+ 类名）。

1. **在命令输入行进行编译时是需要 利用 javac –d . XXX.java 其中中间的点. 表示的是当前目录，点可以换位任何一个路径。执行时，需要使用 java 包名.类名 ；必须使用包名，否则执行错误。因为此时类的全名包括包名。**
2. 命令 cd 文件夹的名字，表示打开此文件夹；cd .. 加上两个点，表示的是回到上一级目录。
3. Java中的内存分为栈内存和堆内存，栈内存中存放的是对象的引用，而堆内存中存放的是堆的实体。对象本身存放在堆内存中。
4. 匿名对象：不定义对象的引用名称而直接使用这个对象。这种对象成为匿名对象。如：

new Dog（）.jump();m 匿名对象只能使用一次，因为有一个new就会在堆内存中生成一个对象。由于使用的对象没有名字，使用后再也找不到了。

1. 内部类：就是一个类定义在另外一个类里面，这种类称为内部类。编译会产生两个编译文件，一个是外部类的class文件，另外一个是外部类名$内部类名.class
2. 匿名类：没有名字的类。区别于匿名对象。显然类没有名字，这个有点难于理解，可以通过直接对接口操作产生一个类，而此类没有名字，接口是一个抽象类，不可以直接new的，不可以生成对象，但是在这里后面紧跟着实现此接口的实体。

如： interface A{

public void doSomething();

}

B b = new B();

b.fun(new A（）{ // 注意直接new的是接口，必须后面紧跟着实现此接口的实体

public void doSomething(){

System.out.println(“这是个匿名内部类”);

}

}

);

1. 匿名内部类：使用的比较多。两个特点：没有名字和定义在另外一个类的内部。匿名内部类一般直接使用在参数列表中，如上面例子。 new 接口（）{

实体

}

也就是A这个类完全是在B的fun方法的参数列表里定义的。

八、 如果需要一下子编译很多文件，命令javac \*.java ；利用星号代表所有文件名。确定是如果出现错误，不能及时发现。 \* 是个通配符，代表的就是任意多个任意字符。

1. 在命令输入框内清屏的命令为cls
2. new在对象生成过程中完成了三种事情：
3. 为对象开辟了内存空间；
4. 调用类的构造方法；
5. 将生成的对象的地址返回；
6. 第12、13节课的视频教程把前11节课的视频总结了一下，可以直接查看第12节对前11节的回顾，第12、13节是个复习课。
7. Java中输入1.2，默认为double型的变量，但是Java中不可以把范围大的数据类型赋给范围小的数据类型，如 float a = 1.2；会出现错误，方法在1.2后面加上f或F,或者（float）1.2；强制类型转换。
8. 方法重写（Override）：又叫覆写，子类和父类的方法返回类型、方法名称、参数都是一样的。重写发生在父类和子类之间。
9. 方法重写和方法重载之间的关系：方法重载发生在同一个类内部的两个或者多个方法；重写发生在父类和子类之间。
10. 当两个方法形成重写关系时，可以在子类方法中通过利用super.\*\*\*调用父类中的\*\*\*\*方法，此时super.\*\*\*\*不必放在第一行，因为此时父类对象已经构造完成。
11. 设计模式（design pattern）：面向对象的语言都有自己的设计模式。单例模式（singleton）：表示一个类只会生成单一的对象。
12. Append 附加，贴上appendix 附录，附加物

append（）是一个多重载的方法，括号里可以使字符串、证书、布尔值等，实现不同类型的拼接。

加号“+”主要是对字符串的拼接，对于布尔值拼接出错。

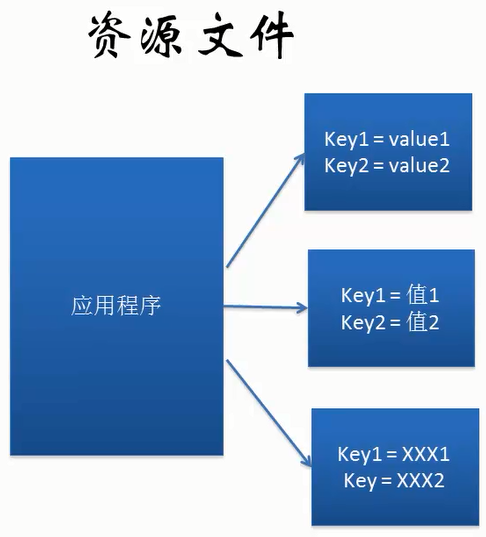
对于字符串的拼接，考试可能常常会考察，但是工程开发利用较少。

1. java.lang.Object类。Java.lang 是个包，编译时自动导入，不需要显式地导入即import。
2. 当打印引用时，实际上会打印出引用所指向对象的toString（）方法的返回值，因为每个类都直接或者间接地继承Object，而Object中定义了toString，所以每个类都有toString这个方法。如object3.toString（）； 。
3. 包装类：wrapped class：针对原生数据类型（primitive）的包装。8个原生数据类型对应8个包装类。目的是使程序完全面象对象。所有的包装类都位于java.lang包下。Java的8个包装类分别是Byte，Short，Long，Integer，Float，Double，Character，Boolean；他们的使用方法都一样，都是实现原生数据类型与包装类型的双向转换。
4. 在静态方法中无法引用分静态方法。
5. java.util.Random 类中有个nextIn（int n）；返回的是【0，n),范围的随机整数，但是不包括n。有很多函数可以返回特定范围的随机数，但是都是左闭右开的范围内的值。如Math类有个方法random返回【0，1);的随机数（double）。Math类在java.lang包下。
6. JAVA 中常量的命名规则是都用大写字母，不同单词之间用下划线连接，如AGE\_OF\_PERSON。对于常量来说，通常定义为public static final int AGE\_OF\_PERSON =20;

定义为static的好处是使类的不同实例使用同一个常量。

1. Eclipse 的重构（refactor）：重构的功能很强大，以后深入研究必须要用到的。
2. 资源文件：键值对（key = value ）。好处：通过键key找到value值，需要修改时，不用再修改源代码了，只修改资源文件中key 相应的value值即可。资源文件的扩展名为properties。 File.separator 代表 文件路径中用的斜杠（正斜杠或者反斜杠（windows））

Properties 是个类。

1. 
2. 得到项目路径的方法：String projectDir = System.get.Property（”项目名字.dir”）；
3. 
4. 接收键盘输入：System.in 是System下的一个成员变量，用于接收用户通过键盘输入。
5. 面向对象：是一种编程方法，是一种思维方式，不是一种编程语言。
6. 学习方法：要习惯于将面向对象与现实世界作比较。
7. 掌握一种面向对象语言的语法；
8. 掌握面向对象的思维方式；
9. 熟悉面向对象设计原则；（面向对象的终极目标就是消除程序中的重复代码）
10. 掌握面向对象设计模式。
11. 思维方式：
12. 首先确定谁来做，其次确定怎么做；
13. 首先考虑整体，其次考虑局部；
14. 首先考虑抽象，其次考虑具体。
15. 类的属性又称成员变量，主要用于描述类的状态；

类的方法又称成员方法（成员函数），主要用于描述类的行为。

1. 如果同一个类中，某个成员方法中定义了一个与成员变量同名的局部变量，则利用此方法调用的是局部变量，而不是成员变量。
2. 类是抽象的，对象是具体的。
3. 堆内存中存放的是对象的实体，栈内存当中存放的是对象的引用。
4. 匿名对象就是没有名字的对象。可以不定义对象的引用名称，而直接调用这个对象的方法，如 new Dog（）.jump();
5. 使用继承也是为了减少重复代码，既然有继承，则允许子类自己进行扩展。关键字extends。子类继承父类的成员变量和成员方法，但是父类的构造方法是不能被继承的。
6. 调用子类的构造方法时，如果没有显式地调用父类的构造方法，则编译器默认地调用父类无参数的构造方法（如果没有则就会报错），如果显式地调用了父类的构造方法（利用super（参数列表）），编译器就不会再默认调用父类的无参数的构造方法了。 就是说调用子类构造方法时必须调用父类的构造方法。
7. 重写（Override）又称复写或覆盖。子类和父类当中的有成员方法的定义（返回值类型，方法名字，参数列表）完全相同。区别去重载，重载是依靠参数列表的参数个数和参数类型及顺序不同而区别为不同的方法的。
8. 抽象方法：只有方法的定义，没有方法体的方法。方法定义：abstract 返回值类型 方法名 （）；方法体指的是 {代码}；
9. 语意上的错误，区别于语法上的错误。
10. Dos下命令 javac –d . 路径 ，就可以自动生成文件夹，. 点表示当前目录，-d 根据相应的包名生成相应的文件夹。
11. 在命令框中输入cls命令就是清屏的意思。
12. **内部类：**
13. **定义：一个类定义在另一个类里面；生成一个内部类的对象方法：**
14. **匿名内部类：**
15. **线程和线程：**
16. **多进程：操作程序中能同时运行多个程序（任务）；**
17. **多线程（Thread）：在同一应用程序中有多个顺序流（同时）执行；**
18. **多线程程序运行模式：**
19. **定义线程的方法：（一个线程本质上就是一个对象）主函数就是个线程。**

**方法一：首先定义一个线程类（继承于类Thread）；生成一个线程类的对象；然后调用对象的start（）；方法，通过复写run（）；方法而实现相应的动作；**

**方法二：首先定义一个Runnable接口的实现类，再生成这个实现类的一个对象，然后生成一个Thread 类的对象，将Runnable接口实现类的对象作为参数传递给该Thread对象（参数在Thread对象生成时的（）里），最后执行Thread的对象的start方法，就会执行重写的run方法的线程体。**

**建议使用第二种方法。（至于要线程干什么，需要重写run（），构成线程体）**

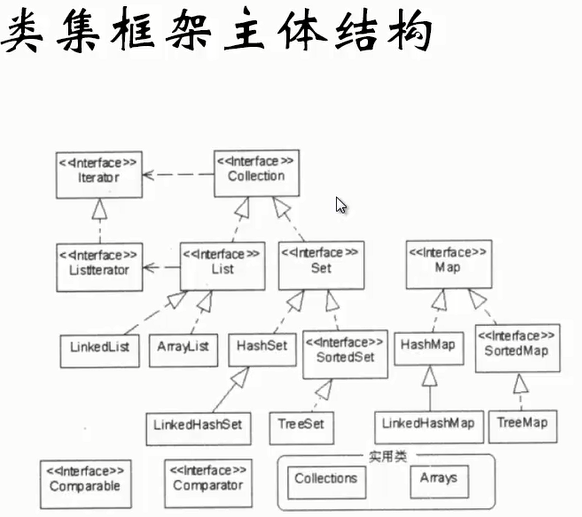
**中断线程 ：① Thread.sleep（int ms）；睡眠多少毫秒② Thread.yield（）;让出CPU重新抢cpu；**

**中断优先级设置： Thread.setPriority(); Thread.getPriority();**

1. **类集框架： 就是一组类和接口；位于java.util包中；用于存储和管理数据；主要分为三大类——集合（set）、列表和映射。**
2. **（1） Set （集合）：集合中的对象不按特定的方式排序，并且没有重复的对象。**

**（2） list （列表）：列表中的对象按照索引位置排序，可以有重复的对象；**

（3） map (映射)：例如键值对。



1. **Set； Iterator 是个接口，Collection；HashSet**

由于equals是定义在Object类里面的，所有的类都继承了equals方法。学习equals要 区 别==与equals的区别。对象的内容相等需要两个条件：①对象的类型相同，可以利用instanceof进行操作符进行比较；② 两个对象的成员变量的值完全相同。

1. Hash算法 toString 也是Object的方法，故所有的类都继承了此方法。
2. 第三种注释方式用来产生Java Doc文档的，/\*\* ………… \*/
3. 抽象类不能实例化，但是可以被继承，也可以拥有自己的构造方法。构造方法是为继承此类的子类所准备的；因为一般情况下，只有在实例化（new 一个新的对象），会调用此类的构造方法，而抽象类不可以实例化，但是继承此抽象类的类不一定是抽象类，这些子类可以实例化，实例化时可以通过super(); 语句来调用父类的构造方法。
4. 抽象方法不需要有函数体，中括号不要添加，只要是包括了中括号{}；就说明了有函数体。
5. 抽象类的作用：包括了抽象方法，那么继承此抽象类的类，必须复写抽象类中的抽象方法，所以起到了提醒的作用。如果不对抽象方法进行复写，则认为是语法错误。把一个方法强制定义成抽象方法，则逼着编程者必须对其进行复写。
6. 获取当前创建的项目的路径的方法： String projectPath = System.getProperty(“user.dir”);这是统一使用的。
7. 在windows下路径名利用的是反斜杠\；在Unix操作系统下使用的是正斜杠/. 利用 File.separator ；就可以在相应系统表现出不同的斜杠。
8. 环境变量：操作系统运行必需的一些参数。环境变量通常就是一系列的键值对。
9. Path 环境变量是操作系统外部命令搜索路径。就是一些搜索目录。
10. Classpath是类文件搜索路径。