

考试预测及知识点总结：

Java一次编写处处运行依靠的是JVM，java源文件后缀.java，字节码文件后缀.class。

java程序：Java应用程序和Java小程序。

编译文件javac xxx.java，运行文件java xxx

如这样的：

Java语言**不允许**在两个嵌套的复合语句内声明两个同名的变量。

- 1 >>>是逻辑右移，移位时高位补0
- 2 移位运算的移位数，会根据变量字长做调整，解释见PPT

类的成员变量有默认值，方法局部变量没有默认值，必须初始化才能引用，否则编译错误。

&& ||是简洁运算，& |非简洁运算

区分类变量、实例变量

类变量和类方法可以直接用“类名.”调用

final实例变量必须在**每个构造方法**结束之前赋初值，以保证使用之前会被初始化。

类方法不能访问实例变量，否则会发生**编译错误**。

包的命名名称必须是“独一无二”的，Java中包名使用小写字母表示

构造方法：方法名与类名相同，没有返回类型（修饰符void也不能有），通常被声明为公有的，在生成一个对象时，系统会自动调用该类的构造方法为新生成的对象初始化。

可以使用this关键字在一个构造方法中调用另外的构造方法。

枚举类型是类，枚举值是类的对象，枚举值是public，static，final的

第2章PPT——BankAccount那段大代码可能考，记得看一下。

错误分为错误和异常。

异常又可以分为非检查型异常（在运行时出现，所以又叫RuntimeException）和检查型异常。

错误：致命性的，**用户程序无法处理**，Error类是所有错误类的父类。

异常的处理try-catch-finally要会写（可能考代码填空）

方法签名：方法名和参数类型一起，构成方法签名（**注意不包括返回值类型**）

方法重载：方法签名不同。**方法重载**又叫做**静态多态**或者**编译时多态**。

（可能会参数类型一样，返回类型不一样，问是不是重载（不是））

静态联编（又叫早期联编）：编译时根据变量类型和方法名称，决定被调方法的签名。

动态联编（又叫晚期联编）：编译时无法确定被调用的方法体，运行时根据对象类型决定。

（Java的方法覆盖就是最好的例子）

Java只支持类的单继承

属性的隐藏：子类声明了与父类同名的变量名，则从父类继承的变量将被隐藏（可以是不同的变量类型）。子类拥有了两个相同名字的变量，一个继承自父类，另一个由自己声明。当子类执行继承自父类操作是，处理的是继承自父类的变量，而当子类执行它自己声明的方法时，所操作的就是它自己声明的变量。

如何访问被隐藏的父亲属性：调用从父类继承的方法，则操作的是从父类继承的属性；使用super.属性

子类**不能继承**父类的静态属性。

覆盖方法的返回类型，方法名称，参数的个数及类型必须和被覆盖的方法一模一样。（注意与方法重载对比）

方法覆盖实现运行时多态。运行时多态是通过动态联编实现的。

要会重写equals方法

```
1 public boolean equals(Object obj){
2     if(obj instanceof xxx){
3         xxx a = (xxx)obj;
4         return ...;
5     }
6     return false;
7 }
```

终结类与方法：被final修饰符修饰的类和方法。

终结类不能被继承。

终结方法不能被当前类的子类重写。（子类重写的话，声明会报错）

抽象类不能new出来

抽象方法，仅有方法头，而没有方法体和操作实现。（**后面没有大括号**）

如果一个抽象类的子类不是抽象类，那它必须重写抽象类中的抽象方法。

只有抽象类才有抽象方法。（抽象类也可以包括非抽象方法）

“?”代表一种类型，它被称为通配符。

面向对象设计：尽可能做到高内聚，低耦合。

接口中也可以包含基本数据类型的数据成员，但它们都默认为static和final。

接口数据一定要赋初值，不能在更改，允许使用final关键字

接口中方法必须是“抽象方法”，不能有方法体，允许省略public及abstract关键字。

实现接口：必须实现接口中的所有方法，来自接口的方法必须声明成public

对象可以转型为其所属接口类型，调用接口声明过的方法。（但转型后，类中声明的一些方法没法用。）

接口的扩展（extends）（可以从extends 接口1，接口2，...）

塑型：任何一个父类类型，对象所属的类实现的一个接口，被塑型为父类或接口后，再被塑型会其本身所在的类。

实例方法查找：从对象创建时的类开始，沿类层次向上查找。

类方法的查找：总是在引用变量申明时所属的类中进行查找（编译时确定）。

绑定：将一个方法调用同一个方法主题连接到一起。

运行前执行绑定：早期绑定。运行时：动态绑定或运行期绑定。

多态的应用：向上塑型技术；动态绑定技术

构造方法的调用顺序：首先调用基类构造方法，然后在进入派生类构造方法前，初始化能够访问的成员（实际上编译器把成员初始化放到构造方法里最前面的地方）