Java SE Lesson 3

即定义时必须赋初值

1. 接口中所声明的方法都是抽象方法。接口中的方法都是 public 的。
2. 接口中也可以定义成员变量。**接口中的成员变量都是 public、final、static 的。**
3. 一个类不能既是 final，又是 abstract 的。因为 abstract 的主要目的是定义一种约定，让子类去实现这种约定，而 final 表示该类不能被继承，这样 abstract 希望该类可以被继承而 final 明确说明该类不能被继承，两者矛盾。因此一个类不能既是 final 的，又是 abstract 的。
4. Design Pattern（设计模式）。单例模式（Singleton）：表示一个类只会生成唯一的一个对象。
5. 包（package）。用于将完成不同功能的类分门别类，放在不同的目录（包）下。包的命名规则：将公司域名反转作为包名。www.shengsiyuan.com，com.shengsiyuan（包名），对于包名：每个字母都需要小写。如果定义类的时候没有使用 package，那么 Java 就认为我们所定义的类位于默认包里面（default package）。
6. 编译带有 package 声明的 Java 源文件有两种方式：

a) 直接编译，然后根据类中所定义的包名，逐一手工建立目录结构，最后将生成的 class 文件放到该目录结构中（很少使用，比较麻烦）。

b) 使用编译参数 –d，方式为 javac –d . 源文件.java，这样在编译后，编译器会自动帮助我们建立好包所对应的目录结构。

7. 有两个包名，分别是 aa.bb.cc 与 aa.bb.cc.dd，那么我们称后者为前者的子包。
8. 导入（import），将使用 package 分离的各个类导入回来，让编译器能够找到所需要的类。
9. import 的语法：import　com.shengsiyuan.PackageTest;
10. import com.shengsiyuan.*，表示导入 com.shengsiyuan 包下面的所有类。
11. import aa.bb.*并不会导入 aa.bb.cc 包下面的类。这时需要这样写：
    import aa.bb.*;
    import aa.bb.cc.*;
12. 关于 package、import、class 的顺序问题：

a) 首先需要定义包（package），可选

b) 接下来使用 import 进行导入，可选

c) 然后才是 class 或 interface 的定义。

13. 如果两个类在同一个包下面，那么则不需要导入，直接使用即可。
14. 访问修饰符（access modifier）。

1） public（公共的）：被 public 所修饰的属性和方法可以被所有类访问。

2） protected（受保护的）：被 protected 所修饰的属性和方法可以在类内部、相同包以及该类的子类所访问。

3） private（私有的）：被 private 所修饰的属性和方法只能在该类内部使用

4） 默认的（不加任何访问修饰符）：在类内部以及相同包下面的类所使用。

15. instanceof：判断某个对象是否是某个类的实例。语法形式：引用名 instanceof 类名（接口名），返回一个 boolean 值。
16. People people = new Man();
17. System.out.println(people instanceof People); //结果为 true，因为 Man 是 People 的子类，根据继承，子类就是父类，因此 Man 也可以看作是 People 的实例。
18. 相等性的比较（==）

1） 对于原生数据类型来说，比较的是左右两边的值是否相等。

2） 对于引用类型来说，比较左右两边的引用是否指向同一个对象，或者说左右两边的引用地址是否相同。

19. java.lang.Object 类。java.lang 包在使用的时候无需显式导入，编译时由编译器自动帮助我们导入。
20. API （Application Programming Interface），应用编程接口。
21. 当打印引用时，实际上会打印出引用所指对象的 toString()方法的返回值，因为每个类都直接或间接地继承自 Object，而 Object 类中定义了 toString()，因此每个类都有 toString()这个方法。
22. 关于进制的表示：16 进制，逢 16 进一，16 进制的数字包括：0～9，A,B,C,D,E,F，
23. **equals()方法，该方法定义在 Object 类当中，因此 Java 中的每个类都具有该方法，对于 Object 类的 equals()方法来说，它是判断调用 equals()方法的引用与传进来的引用是否一致，即这两个引用是否指向的是同一个对象。对于 Object 类的 equals()方法来说，它等价于==。**
24. **对于 String 类的 equals()方法来说，它是判断当前字符串与传进来的字符串的内容是否一致。**
25. 对于 String 对象的相等性判断来说，请使用 equals()方法，而不要使用==。
26. String 是常量，其对象一旦创建完毕就无法改变。当使用+拼接字符串时，会生成新的 String 对象，而不是向原有的 String 对象追加内容。
27. String Pool（字符串池）
28. String s = “aaa”;（采用字面值方式赋值）

1) 查找 String Pool 中是否存在“aaa”这个对象，如果不存在，则在 String Pool 中创建一个“aaa”对象，然后将 String Pool 中的这个“aaa”对象的地址返回来，赋给引用变量 s，这样 s 会指向 String Pool 中的这个“aaa”字符串对象

2) 如果存在，则不创建任何对象，直接将 String Pool 中的这个“aaa”对象地址返回来，赋给 s 引用。

29. String s = new String(“aaa”);

1) 首先在 String Pool 中查找有没有“aaa”这个字符串对象，如果有，则不在 String Pool 中再去创建“aaa”这个对象了，直接在堆中（heap）中创建一个“aaa”字符串对象，然后将堆中的这个“aaa”对象的地址返回来，赋给 s 引用，导致 s 指向了堆中创建的这个“aaa”字符串对象。

2) 如果没有，则首先在 String Pool 中创建一个“aaa“对象，然后再在堆中（heap）创建一个”aaa“对象，然后将堆中的这个”aaa“对象的地址返回来，赋给 s 引用，导致 s 指向了堆中所创建的这个”aaa“对象。