Java SE Lesson 2

1. 多态：父类型的引用可以指向子类型的对象。
2. Parent p = new Child();当使用多态方式调用方法时，首先检查父类中是否有sing()方法，**如果没有则编译错误；如果有，再去调用子类的 sing()方法。**
3. 一共有两种类型的强制类型转换：

a) 向上类型转换（upcast）：比如说将 Cat 类型转换为 Animal 类型，即将子类型转换为父类型。对于向上类型转换，不需要显式指定。

b) 向下类型转换（downcast）：比如将 Animal 类型转换为 Cat 类型。即将父类型转换为子类型。对于向下类型转换，必须要显式指定（必须要使用强制类型转换）。

4. 抽象类（abstract class）：使用了 abstract 关键字所修饰的 类叫做抽象类。抽象类无法实例化，也就是说，不能 new 出来一个抽象类的对象（实例）。
5. 抽象方法（abstract method）：使用 abstract 关键字所修饰的方法叫做抽象方法。**抽象方法需要定义在抽象类中**。相对于抽象方法，之前所定义的方法叫做具体方法（有声明，有实现）。
6. 如果一个类包含了抽象方法，那么这个类一定是抽象类。
7. 如果某个类是抽象类，那么该类可以包含具体方法（有声明、有实现）。
8. 如果一个类中包含了抽象方法，那么这个类一定要声明成 abstract class，也就是说，该类一定是抽象类；反之，如果某个类是抽象类，那么该类既可以包含抽象方法，也可以包含具体方法。
9. 无论何种情况，只要一个类是抽象类，那么这个类就无法实例化。
10. 在子类继承父类（父类是个抽象类）的情况下，那么该子类必须要实现父类中所定义的所有抽象方法；否则，该子类需要声明成一个 abstract class。
11. 接口（interface）：接口的地位等同于 class，**接口中的所有方法都是抽象方法**。在声明接口中的方法时，可以使用 abstract 关键字，也可以不使用。通常情况下，都会省略掉 abstract 关键字。
12. 可以将接口看作是特殊的抽象类（抽象类中可以有具体方法，也可以有抽象方法，而接口中只能有抽象方法，不能有具体方法）。
13. 类可以**实现**接口。实现使用关键字 implements 表示，代表了某个类实现了某个接口。
14. 一个类实现了某个接口，那么该类必须要实现接口中声明的所有方法。如果该类是个抽象类，那么就无需实现接口中的方法了。
15. Java 是单继承的，也就是说某个类只能有唯一一个父类；一个类可以实现多个接口，多个接口之间使用逗号分隔。
16. 多态：所谓多态，就是父类型的引用可以指向子类型的对象，或者接口类型的引用可以指向实现该接口的类的实例。关于接口与实现接口的类之间的强制类型转换方式与父类和子类之间的强制类型转换方式完全一样。
17. static 关键字：可以用于修饰属性，也可以用于修饰方法，还可以用于修饰类（后面的课程讲）
18. static 修饰属性：无论一个类生成了多少个对象，所有这些对象共同使用唯一一份静态的成员变量；一个对象对该静态成员变量进行了修改，其他对象的该静态成员变量的值也会随之发生变化。如果一个成员变量是 static 的，

那么我们可以通过**类名.成员变量名**的方式来使用它（推荐使用这种方式）。

19. static 修饰方法：static 修饰的方法叫做静态方法。对于静态方法来说，可以使用**类名.方法名**的方式来访问。
20. **静态方法只能继承，不能重写（Override）。**
21. final 关键字：final 可以修饰属性、方法、类。
22. final 修饰类：当一个类被 final 所修饰时，表示该类是一个终态类，即不能被继承。
23. final 修饰方法：当一个方法被 final 所修饰时，表示该方法是一个终态方法，即不能被重写（Override）。
24. final 修饰属性：当一个属性被 final 所修饰时，表示该属性不能被改写。
25. **当 final 修饰一个原生数据类型时，表示该原生数据类型的值不能发生变化（比如说不能从 10 变为 20）；如果 final 修饰一个引用类型时，表示该引用类型不能再指向其他对象了，但该引用所指向的对象的内容是可以发生变化的。**
26. 对于 final 类型成员变量，一般来说有两种赋初值方式：

a) 在声明 final 类型的成员变量时就赋上初值
b) 在声明 final 类型的成员变量时不赋初值，但在类的所有构造方法中都为其赋上初值。

27. static 代码块：静态代码块。静态代码块的作用也是完成一些初始化工作。首先执行静态代码块，然后执行构造方法。静态代码块在类被加载的时候执行，而构造方法是在生成对象的时候执行；要想调用某个类来生成对象，首先需要将类加载到 Java 虚拟机上（JVM），然后由 JVM 加载这个类来生成对象。
28. 类的静态代码块只会执行一次，是在类被加载的时候执行的，因为每个类只会被加载一次，所以静态代码块也只会被执行一次；而构造方法则不然，每次生成一个对象的时候都会调用类的构造方法，所以 new 一次就会调用构造方法一次。
29. 如果继承体系中既有构造方法，又有静态代码块，那么首先执行最顶层的类的静态代码块，一直执行到最底层类的静态代码块，然后再去执行最顶层类的构造方法，一直执行到最底层类的构造方法。**注意：静态代码块只会执行一次。**
30. 不能在静态方法中访问非静态成员变量；可以在静态方法中访问静态的成员变量。可以在非静态方法中访问静态的成员变量。
31. **总结：静态的只能访问静态的；非静态的可以访问一切。**
32. **不能在静态方法中使用 this 关键字。**