

实际上任何数据类型遇见“+”都会变为字符串进行连接处理（如果是引用数据类型会调用 `toString()` 转换后进行连接），但是这样的连接会造成垃圾空间的产生，所以不建议使用。

- 方式二：利用 `String` 类中提供的方法：`public static String valueOf(数据类型变量)`。

```
public class TestDemo {
    public static void main(String args[]) {
        int num = 100;                // 定义int型变量
        String str = String.valueOf(num); // 变为String型
        System.out.println(str.replaceAll("0", "9")); // 调用String类方法
    }
}
```

程序执行结果： 199

本程序使用 `String` 类提供的 `valueOf()` 方法，此方法使用 `static` 声明，可以直接利用类进行调用，而此方法在 `String` 类中也进行了多次重载，可以将任意数据类型变为 `String` 型数据，这样的操作可以避免垃圾的产生，所以在开发中往往使用较多。

## 本章小结

1. 继承可以扩充已有类的功能。通过 `extends` 关键字实现，可将父类（超类）的成员（包含数据成员与方法）继承到子类（派生类），在 Java 中一个类只允许继承一个父类，存在有单继承局限。

2. Java 在执行实例化子类对象前（子类构造方法执行前），会先默认调用父类中无参的构造方法，其目的是对继承自父类的成员做初始化的操作。

3. 父类有多个构造方法时，如果要调用特定的构造方法，则可在子类的构造方法中，通过 `super()` 这个关键字来完成，但是此语句必须放在子类构造方法的首行。

4. `this` 调用属性或方法时，会先从本类查找是否存在指定的属性或方法，如果没有，则会去查找父类中是否存在指定的属性或方法。而 `super` 是子类直接调用父类中的属性或方法，不会查找本类定义。

5. `this()` 与 `super()` 的相似之处：当构造方法有重载时，两者均会根据所给予的参数类型与个数，正确地执行相应的构造方法；二者均必须编写在构造方法内的第一行，也正是这个原因，`this()` 与 `super()` 无法同时存在于同一个构造方法内。

6. “覆写”（`overriding`），它是在子类当中，定义名称、参数个数与类型均与父类相同的方法，但是覆写的方法不能拥有比父类更为严格的访问控制权限。覆写的意义在于：保存父类中的方法名称，但是不同的子类可以有不同的实现。

7. 如果父类的方法不希望被子类覆写，可在父类的方法之前加上“`final`”关键字，这样该方法便不会被覆写。

8. `final` 的另一个功能是把它加在数据成员变量前面，这样该变量就变成了一个常量，便无法在程序代码中再做修改了。使用 `public static final` 可以声明一个全局常量。

9. 所有的类均继承自 `Object` 类。一个完整的简单 Java 类理论上应该覆写 `Object` 类中的 `toString()`、`equals()`、`hashCode()` 3 个方法。所有的数据类型都可以使用 `Object` 类型接收。

10. Java 可以创建抽象类,专门用来当做父类。抽象类的作用类似于“模板”,其目的是依据其格式来修改并创建新的类,在定义抽象类时类中可以不定义抽象方法。

11. 抽象类的方法可分为两种:一种是普通方法,另一种是以 `abstract` 关键字开头的“抽象方法”。其中,“抽象方法”并没有定义方法体,在子类(不是抽象类)继承抽象类时,必须要覆写全部抽象方法。

12. 抽象类不能直接用来产生对象,必须通过对象的多态性进行实例化操作。

13. 接口是方法和全局常量的集合的特殊结构类,使用 `interface` 关键字进行定义,接口必须被子类实现(`implements`),一个接口可以同时继承多个接口,一个子类也可以同时实现多个接口。

14. Java 并不允许类的多重继承,但是允许实现多个接口,即使用接口来实现多继承的概念。

15. 接口与一般类一样,均可通过扩展的技术来派生出新的接口。原来的接口称为基本接口或父接口;派生出的接口称为派生接口或子接口。通过这种机制,派生接口不仅可以保留父接口的成员,同时也可以加入新的成员以满足实际的需要。

16. Java 对象的多态性分为:向上转型(自动)、向下转型(强制)。

17. 通过 `instanceof` 关键字,可以判断对象属于哪个类。

18. 匿名内部类的好处是可利用内部类创建不具有名称的对象,并利用它访问类里的成员。

19. 基本数据类型的包装类可以让基本数据类型以对象的形式进行操作,从 JDK 1.5 开始支持自动装箱与拆箱操作,这样就可以使用 `Object` 接收基本数据类型。

20. 在包装类中提供了将字符串转换为基本数据类型的操作方法,但是要注意字符串的组成是否正确。

## 课后习题

### 一、填空题

1. Java 中通过\_\_\_\_\_关键字实现继承。
2. 一个类只能继承\_\_\_\_\_个父类,但能实现\_\_\_\_\_接口。
3. \_\_\_\_\_类是所有类的父类,该类中判断两个对象是否相等的方法是\_\_\_\_\_,取得对象完整信息的方法是\_\_\_\_\_。
4. `Integer` 类是对\_\_\_\_\_基本数据类型的封装。`Float` 类是对\_\_\_\_\_基本数据类型的封装。`Double` 类是对\_\_\_\_\_基本数据类型的封装。字符类 `Character` 是对\_\_\_\_\_基本数据类型的封装。
5. 当子类中定义的方法与父类方法同名且参数类型及个数、返回值类型相同时,称子类方法父类方法,子类默认使用\_\_\_\_\_方法,使用父类的同名方法,必须使用\_\_\_\_\_关键字说明。
6. 当子类定义的成员变量与父类的成员变量同名时,称子类\_\_\_\_\_父类的成员变量,子类默认使用\_\_\_\_\_属性。使用父类的同名成员变量,必须用\_\_\_\_\_关键字说明。
7. 如果子类定义了构造方法,在创建子类对象时首先默认调用\_\_\_\_\_,然后调用本类的构造方法。

### 二、选择题

1. 不能直接使用 `new` 创建对象的类是( )。
  - A. 静态类
  - B. 抽象类
  - C. 最终类
  - D. 公有类

2. 为类定义多个名称相同,但参数的类型或个数不同的方法的做法称为( )。
- A. 方法重载      B. 方法覆写      C. 方法继承      D. 方法重用
3. 定义接口的关键字是( )。
- A. extends      B. class      C. interface      D. public
4. 现在有两个类 A、B,以下描述中表示 B 继承自 A 的是( )。
- A. class A extends B      B. class B implements A  
C. class A implements      D. class B extends A
5. 下面关于子类调用父类构造方法的描述正确的是( )。
- A. 子类定义了自己的构造方法,就不会调用父类的构造方法  
B. 子类必须通过 super 关键字调用父类有参的构造方法  
C. 如果子类的构造方法没有通过 super 调用父类的构造方法,那么子类会先调用父类中无参数构造方法,再调用子类自己的构造方法  
D. 创建子类对象时,先调用子类自己的构造方法,让后再调用父类的构造方法
6. 假设类 X 是类 Y 的父类,下列声明对象 x 的语句中不正确的是( )。
- A. X x = new X(); B. X x = new Y(); C. Y x = new Y(); D. Y x = new X();
7. 编译并运行下面的程序,结果是( )。

```
public class A {
    public static void main(String args[]) {
        B b = new B();
        b.test();
    }
    void test() {
        System.out.print("A");
    }
}
class B extends A {
    void test() {
        super.test();
        System.out.println("B");
    }
}
```

- A. 产生编译错误      B. 代码可以编译运行,并输出结果: AB  
C. 代码可以编译运行,但没有输出      D. 编译没有错误,但会运行时会产生异常
8. 编译运行下面的程序,结果是( )。

```
public class A {
    public static void main(String args[]) {
        B b = new B();
        b.test();
    }
    public void test() {
        System.out.print("A");
    }
}
```

```

    }
}
class B extends A {
    void test() {
        super.test();
        System.out.println("B");
    }
}

```

- A. 产生编译错误, 因为类 B 覆盖类 A 的方法 test() 时, 降低了其访问控制的级别  
 B. 代码可以编译运行, 并输出结果: AB  
 C. 代码可以编译运行, 但没有输出  
 D. 代码可以编译运行, 并输出结果: A
9. ( ) 修饰符所定义的方法必须被子类所覆写。  
 A. final                  B. abstract                  C. static                  D. interface
10. ( ) 修饰符所定义的方法不能被子类所覆写。  
 A. final                  B. abstract                  D. static                  D. interface
11. 下面的程序编译运行的结果是 ( )。

```

public class A implements B {
    public static void main(String args[]) {
        int m, n;
        A a = new A();
        m = a.K;
        n = B.K;
        System.out.println(m + ", " + n);
    }
}
interface B {
    int K = 5;
}

```

- A. 5, 5                                  B. 0, 5  
 C. 0, 0                                  D. 编译程序产生编译结果
12. 下面关于接口的说法中不正确的是 ( )。  
 A. 接口所有的方法都是抽象的  
 B. 接口所有的方法一定都是 public 类型  
 C. 用于定义接口的关键字是 implements  
 D. 接口是 Java 中的特殊类, 包含全局常量和抽象方法
13. 下面关于 Java 的说法不正确的是 ( )。  
 A. abstract 和 final 能同时修饰一个类  
 B. 抽象类不光可以做父类, 也可以做子类  
 C. 抽象方法不一定声明在抽象类中, 也可以在接口中  
 D. 声明为 final 的方法不能在子类中覆写

### 三、判断题

1. final 声明的类可以有子类。 ( )
2. 一个类继承了抽象类, 则抽象类中的抽象方法需要在其子类中覆写。 ( )
3. final 类型的变量是常量, 其内容不可改变。 ( )
4. 一个类不能既是子类又是父类。 ( )
5. 子类只能继承父类的成员, 但不能修改父类成员。 ( )
6. Java 语言只支持单继承, 不支持多继承。 ( )
7. 子类可以继承父类的所有成员。 ( )
8. 一个接口可以继承一个抽象类。 ( )
9. 一个接口可以同时继承多个接口。 ( )
10. 在程序中 this 和 super 调用构造方法时可以同时出现。 ( )

### 四、简答题

1. 简述 this 与 super 关键字的区别。
2. 简述方法的重载与覆写的区别。
3. 在已有类的基础上派生新的类有什么好处?
4. 如何区分子类和父类? 子类可以继承父类的哪些内容?
5. 什么是多态? 实现多态的方法有哪些?
6. 接口有哪些特征? 如何定义和实现接口?
7. 接口和抽象类有哪些区别?
8. 简述基本数据类型的自动装箱及自动拆箱操作。

### 五、编程题

1. 定义一个 `ClassName` 接口, 接口中只有一个抽象方法 `getClassName()`。设计一个类 `Company`, 该类实现接口 `ClassName` 中的方法 `getClassName()`, 功能是获取该类的类名称。编写应用程序使用 `Company` 类。

2. 建立一个人类 (`Person`) 和学生类 (`Student`) 功能要求:

A. `Person` 中包含 4 个保护型的数据成员 `name`、`address`、`sex`、`age` 分别为字符串、字符串、字符及整型。表示: 姓名、地址、性别和年龄。一个四参构造方法, 一个无参构造方法, 及一个输出方法用于显示四种属性。

B. `Student` 继承 `Person`, 并增加输出成员 `math`、`english` 存放数学和英语成绩。一个六参构造方法, 一个两参构造方法, 一个无参构造方法, 重写输出方法用于显示全部六种属性。

3. 定义员工类, 具有姓名、年龄、性别属性, 并具有构造方法, 显示数据方法, 定义管理层类, 继承员工类, 并有自己的属性: 职务、年薪。定义职员类, 继承员工类, 并有自己的属性: 所属部门、月薪。

4. 定义类 `Shape` 表示一般二维图形。`Shape` 具有抽象方法 `area` 和 `perimeter`, 分别计算形状的面积和周长。试定义一些二维形状类 (如矩形、三角形、圆形等), 这些类均为 `Shape` 类的子类。