

复习

1. 子类通过成员变量的隐藏和方法的重写可以把父类的状态和行为改变为自身的状态和行为。
2. 如果一个 Java Application 程序文件中定义有 3 个类，则使用 JDK 编译器 `javac.exe` 编译该源程序文件将产生 3 个文件名与类名相同而扩展名为 `.class` 的字节码文件。
3. 与类同名的方法是构造方法。
4. 创建一个名为 `MyPackage` 的包的语句是 `package MyPackage`，该语句应该放在程序的位置为应该在程序第一句。
5. 按对象和类的关系，对象是类的实例。
6. `super` 通常用来引用当前对象的父对象。
7. 构造函数的名称必须与类同名，而且不能指定返回值类型。
8. 字符型常量用 `' '` 符号括起来，而字符串型常量用 `" "` 符号括起来。
9. 所有图形绘制的方法是由 `Graphics` 类提供的。
10. 用修饰符 `static` 说明的成员变量是类变量。
11. Java 中类是创建对象的模板。
12. 对象声明的一般格式为类名对象名。
13. 访问权限是公有变量，不但在本类中使用，还可以在任何另外一个类中使用。
14. 当子类中定义的成员变量和父类中的成员变量同名时，子类的成员变量____隐藏了父类的成员变量。
15. `Java.lang` 包是 java 语言的核心类库，它包含了运行 java 程序必不可少的系统类。
16. 变量的名字与成员变量的名字相同，则成员变量被隐藏，该成员变量在这个方法内暂时失效。
17. 在 Java 程序中，通过类的定义只能实现单重继承，但通过接口的定义可以实现多重继承关系。
18. 一个类通过使用 `new` 运算符可以创建多个不同对象，这些对象将分配不同得内存空间。
19. 访问权限是友好变量，只能在本类和与本类同一个包中的其他类使用。
20. 抽象 (`abstract`) 方法是一种仅有方法头，没有具体方法体和操作实现的方法，该方法必须在抽象类之中定义。最终 (`final`) 方法是不能被当前类的子类重新定义的方法。
21. 声明为 `protected` 的类成员可以被此类作用域中的任何对象访问。
22. Java 中不同对象的实例变量将被分配不同的内存空间。
23. 如果一个类的声明中没有使用 `extends` 关键字，这个类被系统默认为是 `Object` 的子类。
24. Java 的源代码中定义几个类，编译结果就生成几个以 `.class` 为后缀的字节码文件。(对)
25. 构造函数用于创建类的实例对象，构造函数名应与类名相同，返回类型为 `void`。(错)
26. 静态初始化器是在其所属的类加载内存时由系统自动调用执行。(对)
27. 在 Java 中对象可以赋值，只要使用赋值号 (等号) 即可，相当于生成了一个各属性与赋值对象相同的新对象。(错)

28. 类的 private 类型的成员变量不可以被继承。(对)
29. 除串结束标志外, Java 的字符串常量"c:\\1234.doc"中共有 11 个字符。(错)
30. Java 允许一个类实现多个接口, 从而达到多重继承。(对)
31. Java 是一种面向对象的程序设计语言。(对)
32. Java 中类的构造函数只能有一个。(对)
33. 类的 public 类型的成员变量不可以被继承。(错)
34. 子类可以继承父类所有的成员变量及成员函数。(错)
35. Java 类中不能存在同名的两个成员函数。(错)
36. Java 源程序文件中是不区分字母的大小写的。(错)
37. applet 可以运行在浏览器中。(对)
38. Java 程序中的起始类名称必须与存放该类的文件名相同。(对)
39. Unicode 是用 16 位来表示一个字的。(对)

1. 什么是对象? 如何创建对象?

对象是类的具体, 是操作的直接对象。创建对象需要借助 new 操作符和构造函数配合使用, 一般格式为: 类名 对象名 = new 类名 ()

2. 字节码?好处?

字节码: 经过 JAVA 编译器编译后, 与特定机器无关, 需要解释器才能执行的中间代码。

好处: 字节码中不包含有硬件信息, 为 JAVA 跨平台的运行提供了基础。

3. 什么是抛出异常? 如何抛出异常?

抛出异常是指在程序过程中通过 throw 语句以 “throw new 异常类名” 的格式抛出异常。异常名可以选择系统异常类名, 也可以使用自定义异常类名。

4. 什么是构造方法?构造方法有那些特点?

构造方法是一个类方法中方法名与类名相同的类方法, 其作用是当使用 new 关键字创建对象时, 自动调用并对新建对象进行初始化操作。

构造方法特点:

- 构造方法与类名相同
- 构造方法没有返回值, 不定义返回类型
- 构造方法可以重载, 针对参数数量和类型定义多个同名构造方法
- 构造方法可以继承, 子类可以继承父类的构造方法
- 构造方法不是由编程人员显式调用, 其在对象创建时由 new 运算符自动调用
- 构造方法只用来对类实例进行初始化, 目的在于简化初始化操作

5. Error 和 Exception 类有什么不同?

Error 类是处理内部系统错误, Exception 类是解决程序本身及其环境所产生的异常。Exception 类可以被捕获和处理, Error 类异常只能在发生时由用户按照系统提示关闭程序。

6. 多态中的重载和覆盖有什么不同？

覆盖是子类对父类同名方法的重定义，其方法名同而方法内容不同，重载是指在类中创建多个方法，它们具有相同的名称但有不同的参数和不同的实现。

7. 何为线程和多线程？如何建立多线程？

线程是一个程序内部的顺序控制流，多线程是在同一个应用程序中有多个顺序控制流同时执行。创建线程有两种方法：继承 Thread 类、通过定义实现新 Runnable 接口。

8. 什么是类的封装？如何对成员变量和方法的访问权限的设置以达到数据封装的目的？

类的封装是指类的设计者为使用者提供类对象可以访问的部分，而对于类中其余的成员变量和方法隐藏起来，用户不能访问。

可以通过 public、protected、private、默认四种控制符来实现数据封装的目的。

9. 静态方法与非静态方法的区别？

静态方法与非静态方法的语法上的区别在于是否被 static 关键字修饰

- a. 非静态方法可以对静态变量和非静态变量操作，而静态方法只能对静态变量进行操作
- b. 静态方法可以有对象调用也可以由类名直接调用，而非静态方法只能由对象调用
- c. 一个类中的方法可以互相调用，但静态方法只能调用静态方法，不能调用非静态方法
- d. 创建对象之前，非静态方法没有入口

10. 线程的生命周期由哪几种状态？各状态之间分别用哪些方法切换？

线程的生命周期有：就绪、可运行、不可运行、终止四种。

就绪->可运行：start()

可运行->可运行：yield()

可运行->不可运行：sleep()、wait()

不可运行->可运行：阻塞解除、睡眠时间终止、唤醒

可运行->终止：run()

11. 编写一个字符界面的 Java Application 程序，接受用户输入的 10 个整数，并输出这 10 个整数的最大值和最小值。

```
import java.io.* ;
public class abc
{
    public static void main(String args[ ])
    { int i , n = 10 , max = 0 , min = 0 , temp = 0;
      try {
          BufferedReader br = new BufferedReader(
```

```

        new InputStreamReader(System.in));
        max = min = Integer.parseInt(br.readLine());
    } catch ( IOException e ) { } ;
for ( i = 2 ; i <= n ; i ++ ) {
    try {
        BufferedReader br = new BufferedReader(
            new InputStreamReader(System.in));
        temp = Integer.parseInt(br.readLine());
        if (temp > max ) max=temp;
        if (temp < min) min=temp;
    } catch ( IOException e ) { } ;
}
System.out.println("max="+max+"\nmin="+min);
}
}

```

12. 创建一个标题为“按钮”的窗框，窗框中显示有“按下我”字样的按钮。

```

import java.awt.*;
public class abc
{ public static void main(String args[])
    { new FrameOut(); }
}
class FrameOut extends Frame // Frame 为系统定
    { Button btn; // 义的
窗框类
FrameOut( )
    {
        super("按钮");
        btn = new Button("按下我");
        setLayout(new FlowLayout());
        add(btn);
        setSize(300,200);
        show();
    }
}
}

```

13. 计算 $1/1+1/2+1/3+\dots+1/100$ 的值。

```

public class Sum
{ public static void main( String args[ ])
    { double sum = 0.0 ;
        for ( int i = 1 ; i <= 100 ; i ++ )
            sum += 1.0/(double) i ;
        System.out.println( "sum="+sum );
    }
}

```

```
}
```

14. 从标准输入(即键盘)读入 10 个整数存入整型数组 a 中, 然后逆序输出这 10 个整数。

```
import java.io.* ;
public class Reverse
{   public static void  main(String  args[ ])
    {   int    i , n =10 ;
        int   a[ ] = new int[10];
        for   ( i = 0 ; i < n ; i ++ )
            try {
                BufferedReader  br = new BufferedReader(
                    new  InputStreamReader(System.in));
                a[i] = Integer.parseInt(br.readLine( ));  // 输入一
个整数
            } catch ( IOException  e ) { } ;
        for   ( i = n-1 ; i >= 0 ; i -- )
            System.out.print(a[i]+" ");
        System.out.println( );
    }
```