

# 《Java 基础入门》课后习题

## 第 1 章 Java 开发入门

### 一、填空题

- 1、Java 的三大体系分别是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- 2、Java 程序的运行环境简称之为\_\_\_\_\_。
- 3、编译 Java 程序需要使用\_\_\_\_\_命令。
- 4、javac.exe 和 java.exe 两个可执行程序存放在 JDK 安装目录的\_\_\_\_\_目录下。
- 5、\_\_\_\_\_环境变量用来存储 Java 的编译和运行工具所在的路径，而\_\_\_\_\_环境变量则用来保存保存 Java 虚拟机要运行的“.class”文件路径。

### 二、选择题

- 1、以下选项中，哪些属于 JDK 工具？（多选）  
A、Java 编译器 B、Java 运行工具 C、Java 文档生成工具 D、Java 打包工具
- 2、Java 属于以下哪种语言？  
A、机器语言 B、汇编语言 C、高级语言 D、以上都不对
- 3、下面哪种类型的文件可以在 Java 虚拟机中运行？  
A、.java B、.jre C、.exe D、.class
- 4、安装好 JDK 后，在其 bin 目录下有许多 exe 可执行文件，其中 java.exe 命令的作用是以下哪一种？  
A、Java 文档制作工具 B、Java 解释器 C、Java 编译器 D、Java 启动器
- 5、如果 jdk 的安装路径为：d:\jdk，若想在命令窗口中任何当前路径下，都可以直接使用 javac 和 java 命令，需要将环境变量 path 设置为以下哪个选项？  
A. d:\jdk; B. d:\jdk\bin; C. d:\jre\bin; D. d:\jre;

### 三、问答题

- 1、简述 Java 的特点。
- 2、简述 JRE 与 JDK 的区别。

### 四、编程题

使用记事本编写一个 HelloWorld 程序，在 dos 命令行窗口编译运行。请按照题目的要求编写程序并给出运行结果。

## 第 2 章 Java 编程基础

### 一、填空题

- 1、Java 中的程序代码都必须在一个类中定义，类使用\_\_\_\_\_关键字来定义。
- 2、布尔常量即布尔类型的两个值，分别是\_\_\_\_\_和\_\_\_\_\_。
- 3、Java 中的注释可分为三种类型，分别是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- 4、Java 中的变量可分为两种数据类型，分别是\_\_\_\_\_和\_\_\_\_\_。
- 5、在 Java 中，byte 类型数据占\_\_\_\_\_个字节，short 类型数据占\_\_\_\_\_个字节，int 类型数据占\_\_\_\_\_个字节，long 类型数据占\_\_\_\_\_个字节。

- 6、在逻辑运算符中，运算符\_\_\_\_\_和\_\_\_\_\_用于表示逻辑与，\_\_\_\_\_和\_\_\_\_\_表示逻辑或。
- 7、若  $x = 2$ ，则表达式  $(x++) / 3$  的值是\_\_\_\_\_。
- 8、若 `int a = 2; a += 3;` 执行后，变量 `a` 的值为\_\_\_\_\_。
- 9、若 `int []a = {12, 45, 34, 46, 23};`，则 `a[2]` = \_\_\_\_\_。
- 10、若 `int a[3][2] = {{123, 345}, {34, 56}, {34, 56}};`，则 `a[2][1]` = \_\_\_\_\_。

## 二、判断题

- 1、Java 语言不区分大小写。
- 2、0xC5 表示的是一个十六进制整数。
- 3、continue 语句只用于循环语句中，它的作用是跳出循环。
- 4、“/\*...\*/”中可以嵌套“//”注释，但不能嵌套“/\*...\*/”注释。
- 5、 $-5\%3$  的运算结果是 2。

## 三、选择题

- 1、以下选项中，哪些属于合法的标识符？（多选）  
A、Hello\_World    B、class    C、123username    D、username123
- 2、关于方法重载的描述，以下选项中哪些是正确的？（多选）  
A、方法名必须一致    B、返回值类型必须不同  
C、参数个数必须一致    D、参数的个数或类型不一致
- 3、以下关于变量的说法错误的是？  
A、变量名必须是一个有效的标识符  
B、变量在定义时可以没有初始值  
C、变量一旦被定义，在程序中的任何位置都可以被访问  
D、在程序中，可以将一个 byte 类型的值赋给一个 int 类型的变量，不需要特殊声明
- 4、以下选项中，switch 语句判断条件可以接收的数据类型有哪些？（多选）  
A、int    B、byte    C、char    D、short
- 5、假设 `int x = 2`，三元表达式 `x > 0 ? x + 1 : 5` 的运行结果是以下哪一个？  
A、0    B、2    C、3    D、5
- 6、下面的运算符中，用于执行除法运算是哪个？  
A、/    B、\    C、%    D、\*
- 7、下列语句哪些属于循环语句？（多选）  
A、for 语句    B、if 语句    C、while 语句    D、switch 语句
- 8、下面哪种写法可以实现访问数组 `arr` 的第 1 个元素？  
A、`arr[0]`    B、`arr(0)`    C、`arr[1]`    D、`arr(1)`
- 9、以下哪个选项可以正确创建一个长度为 3 的二维数组？  
A、`new int [2][3];`    B、`new int [3][];`  
C、`new int [][][3];`    D、以上答案皆不对
- 10、请先阅读下面的代码。

```
int x = 1;
int y = 2;
if (x % 2 == 0) {
    y++;
} else {
    y--;
}
System.out.println("y=" + y);
```

上面一段程序运行结束时，变量 y 的值为下列哪一项？

A、1    B、2    C、3    D、switch 语句

#### 四、程序分析题

阅读下面的程序，分析代码是否能够编译通过，如果能编译通过，请列出运行的结果。否则请说明编译失败的原因。

代码一：

```
public class Test01 {  
    public static void main(String[] args) {  
        byte b = 3;  
        b = b + 4;  
        System.out.println("b=" + b);  
    }  
}
```

代码二：

```
public class Test02 {  
    public static void main(String[] args){  
        int x = 12;  
        {  
            int y = 96;  
            System.out.println("x is " + x);  
            System.out.println("y is " + y);  
        }  
        y = x;  
        System.out.println("x is " + x);  
    }  
}
```

代码三：

```
public class Test03 {  
    public static void main(String args[]) {  
        int x = 4, j = 0;  
        switch (x) {  
            case 1:  
                j++;  
            case 2:  
                j++;  
            case 3:  
                j++;  
            case 4:  
                j++;  
            case 5:  
                j++;  
            default:  
                j++;  
        }  
    }  
}
```

```
        System.out.println(j);  
    }  
}
```

代码四：

```
public class Test04 {  
    public static void main(String args[]) {  
        int n = 9;  
        while (n > 6) {  
            System.out.println(n);  
            n--;  
        }  
    }  
}
```

## 五、简答题

- 1、请列举 Java 语言中的八种基本数据类型，并说明每种数据类型所占用的空间大小。
- 2、简述&&与&的区别并举例说明。
- 3、什么是方法重载？

## 六、编程题

请按照题目的要求编写程序并给出运行结果。

- 1、请编写程序，实现计算“1+3+5+7+...+99”的值。

提示：

- 1) 使用循环语句实现自然数 1~99 的遍历。
- 2) 在遍历过程中，通过条件判断当前遍历的数是否为奇数，如果是就累加，否则不加。

- 2、已知函数

$$y = \begin{cases} x + 3 & (x > 0) \\ 0 & (x = 0) \\ x^2 - 1 & (x < 0) \end{cases}$$

请设计一个方法实现上面的函数，根据传入的值 x 的不同，返回对应的 y 值。

提示：

- 1) 定义一个 static 修饰符修饰的方法，方法接收一个 int 类型的参数 x，返回值为 int 类型。
  - 2) 在方法中使用 if...else if...else 语句针对 x 的值进行三种情况的判断。
  - 3) 根据判断结果分别执行不同的表达式，并将结果赋予变量 y。
  - 4) 在方法的最后返回 y 的值。
  - 5) 在 main 方法中调用设计好的方法，传入一个 int 型的值，将方法的返回值打印。
- 3、请编写程序，实现对数组 {25,24,12,76,101,96,28} 的排序。

提示：使用冒泡排序算法。

## 第 3 章 面向对象（上）

### 一、填空题

- 1、面向对象的三大特征是\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
- 2、在 Java 中，可以使用关键字\_\_\_\_\_来创建类的实例对象。
- 3、定义在类中的变量被称为\_\_\_\_\_，定义在方法中的变量被称为\_\_\_\_\_。
- 4、面向对象程序设计的重点是\_\_\_\_\_的设计，\_\_\_\_\_是用来创建对象的模板。
- 5、在非静态成员方法中，可以使用关键字\_\_\_\_\_访问类的其它非静态成员。
- 6、当一个对象被当成垃圾从内存中释放时，它的\_\_\_\_\_方法会被自动调用。
- 7、被 static 关键字修饰的成员变量被称为\_\_\_\_\_，它可以被该类所有的实例对象共享。
- 8、在一个类中，除了可以定义属性、方法，还可以定义类，这样的类被称为\_\_\_\_\_。
- 9、在 Java 中，提供了一个\_\_\_\_\_命令，用于将程序中的文档注释提取出来，生成 HTML 格式的帮助文档。
- 10、所谓类的封装是指在定义一个类时，将类中的属性私有化，即使用\_\_\_\_\_关键字来修饰。

## 二、判断题

- 1、在定义一个类的时候，如果类的成员被 private 所修饰，该成员不能在类的外部被直接访问。
- 2、Java 中的每个类都至少有一个构造方法，一个类中如果没有定义构造方法，系统会自动为这个类创建一个默认的构造方法。
- 3、声明构造方法时，不能使用 private 关键字修饰。
- 4、类中 static 修饰的变量或方法，可以使用类名或对象的引用变量访问。
- 5、方法内部类中不能访问外部类的成员变量。

## 三、选择题

- 1、类的定义必须包含在以下哪种符号之间？  
A、方括号[]    B、花括号{}    C、双引号“”    D、圆括号()
- 2、下面哪一个是正确的类的声明？  
A、public void HH {...}    B、public class Move() {...}  
C、public class void number{}    D、public class Car {...}
- 3、在以下什么情况下，构造方法会被调用？  
A、类定义时    B、创建对象时    C、调用对象方法时    D、使用对象的变量时
- 4、下面对于构造方法的描述，正确有哪些？(多选)  
A、方法名必须和类名相同  
B、方法名的前面没有返回值类型的声明  
C、在方法中不能使用 return 语句返回一个值  
D、当定义了带参数的构造方法，系统默认的不带参数的构造方法依然存在
- 5、使用 this 调用类的构造方法，下面的说法正确的是？(多选)  
A、使用 this 调用构造方法的格式为 this([参数 1,参数 2...])  
B、只能在构造方法中使用 this 调用其它的构造方法  
C、使用 this 调用其它构造方法的语句必须放在第一行  
D、不能在一个类的两个构造方法中使用 this 互相调用
- 6、下面哪些可以使用 static 关键字修饰？(多选)  
A、成员变量    B、局部变量    C、成员方法    D、成员内部类
- 7、关于内部类，下面说法正确的是？(多选)  
A、成员内部类是外部类的一个成员，可以访问外部类的其它成员  
B、外部类可以访问成员内部类的成员  
C、方法内部类只能在其定义的当前方法中进行实例化  
D、静态内部类中可以定义静态成员，也可以定义非静态成员
- 8、下面对于单例设计模式的描述，正确的是？(多选)

- A、类中定义一个无参的构造方法，并且声明为私有
- B、在内部创建一个该类的实例对象，使用静态变量引用该实例对象
- C、使用 `private` 修饰静态变量，禁止外界直接访问
- D、定义静态方法返回该类实例

9、请先阅读下面的代码

```
public class Test {  
    public Test() {  
        System.out.println("构造方法一被调用了");  
    }  
    public Test(int x) {  
        this();  
        System.out.println("构造方法二被调用了");  
    }  
    public Test(boolean b) {  
        this(1);  
        System.out.println("构造方法三被调用了");  
    }  
    public static void main(String[] args) {  
        Test test = new Test(true);  
    }  
}
```

上面程序的运行结果为下列哪一项？

- A、构造方法一被调用了
- B、构造方法二被调用了
- C、构造方法三被调用了
- D、以上三个选项之和

10、Outer 类中定义了一个成员内部类 Inner，需要在 `main()` 方法中创建 Inner 类实例对象，以下四种方式哪一种是正确的？

- A、`Inner in = new Inner();`
- B、`Inner in = new Outer.Inner();`
- C、`Outer.Inner in = new Outer.Inner();`
- D、`Outer.Inner in = new Outer().new Inner();`

#### 四、程序分析题

阅读下面的程序，分析代码是否能够编译通过，如果能编译通过，请列出运行的结果。否则请说明编译失败的原因。

代码一：

```
class A {  
    private int secret = 5;  
}  
public class Test1 {  
    public static void main(String[] args) {  
        A a = new A();  
        System.out.println(a.secret++);  
    }  
}
```

代码二：

```
public class Test2 {  
    int x = 50;
```

```
static int y =200 ;
public static void method() {
    System.out.println(x+y);
}
public static void main(String[] args) {
    Test2.method();
}
}
```

代码三:

```
public class Outer {
    public String name="Outer";
    private class Inner {
        String name="inner";
        void showName(){
            System.out.println(name);
        }
    }
    public static void main(String[] args) {
        Inner inner = new Outer().new Inner();
        System.out.println(inner.name);
    }
}
```

## 五、简答题

- 1、构造方法和普通的成员方法有什么区别？
- 2、单例设计模式具备哪些特点？

## 六、编程题

- 1、请按照以下要求设计一个学生类 **Student**，并进行测试。

要求如下：

- 1) **Student** 类中包含姓名、成绩两个属性
- 2) 分别给这两个属性定义两个方法，一个方法用于设置值，另一个方法用于获取值。
- 3) **Student** 类中定义一个无参的构造方法和一个接收两个参数的构造方法，两个参数分别为姓名和成绩属性赋值
- 4) 在测试类中创建两个 **Student** 对象，一个使用无参的构造方法，然后调用方法给姓名和成绩赋值，一个使用有参的构造方法，在构造方法中给姓名和成绩赋值

- 2、定义一个 **Father** 和 **Child** 类，并进行测试。

要求如下：

- 1) **Father** 类为外部类，类中定义一个私有的 **String** 类型的属性 **name**，**name** 的值为“zhangjun”。
- 2) **Child** 类为 **Father** 类的内部类，其中定义一个 **introFather()** 方法，方法中调用 **Father** 类的 **name** 属性。
- 3) 定义一个测试类 **Test**，在 **Test** 类的 **main()** 方法中，创建 **Child** 对象，并调用 **introFather()** 方法。

## 第4章 面向对象（下）

### 一、填空题



- 1、在 Java 语言中，允许使用已存在的类作为基础创建新的类，这种技术我们称为\_\_\_\_\_。
- 2、一个类如果实现一个接口，那么它就需要实现接口中定义的全部\_\_\_\_\_，否则该类就必须定义成\_\_\_\_\_。
- 3、在程序开发中，要想将一个包中的类导入到当前程序中，可以使用\_\_\_\_\_关键字。
- 4、一个类可以从其它的类派生出来，派生出来的类称为\_\_\_\_\_，用于派生的类称为\_\_\_\_\_或者\_\_\_\_\_。
- 5、JDK 中定义了大量的异常类，这些类都是\_\_\_\_\_类的子类或者间接子类。
- 6、定义一个 Java 类时，如果前面加使用\_\_\_\_\_关键字修饰，那么该类不可以被继承。
- 7、如果子类想使用父类中的成员，可以通过关键字\_\_\_\_\_引用父类的成员。
- 8、在 Java 语言中，所有的类都直接或间接继承自\_\_\_\_\_类。
- 9、异常的捕获通常由 try、catch 两部分组成，\_\_\_\_\_代码块用来存放可能发生异常，\_\_\_\_\_代码块用来处理产生的异常。
- 10、在 Java 语言中，要想是将一个已编译的类打包成 jar 文件，可以使用的命令是\_\_\_\_\_，要想在命令行窗口运行 jar 文件，可以使用的命令是\_\_\_\_\_。

## 二、判断题

- 1、抽象方法必须定义在抽象类中，所以抽象类中的方法都是抽象方法。
- 2、Java 中被 final 关键字修饰的变量，不能被重新赋值。
- 3、不存在继承关系的情况下，也可以实现方法重写。
- 4、package 声明语句应当为 Java 源文件中的第一条语句。
- 5、接口中只能定义常量和抽象方法。

## 三、选择题

- 1、在类的继承关系中，需要遵循以下哪个继承原则？  
A、多重      B、单一      C、双重      D、不能继承
- 2、在 Java 语言中，以下哪个关键字用于在方法上声明抛出异常？  
A.try      B.catch      C.throw      D.throws
- 3、关于 super 关键字以下说法哪些是正确的？（多选）  
A、super 关键字可以调用父类的构造方法  
B、super 关键字可以调用父类的普通方法  
C、super 与 this 不能同时存在于同一个构造方法中  
D、super 与 this 可以同时存在于同一个构造方法中
- 4、以下说法哪些是正确的？（多选）  
A、Java 语言中允许一个类实现多个接口  
B、Java 语言中不允许一个类继承多个类  
C、Java 语言中允许一个类同时继承一个类并实现一个接口  
D、Java 语言中允许一个接口继承一个接口
- 5、类中的一个成员方法被下面哪个修饰符修饰，该方法只能在本类被访问？  
A、public      B、protected      C、private      D、default
- 6、关于抽象类的说法哪些是正确的？（多选）  
A、抽象类中可以有非抽象方法。  
B、如果父类是抽象类，则子类必须重写父类所有的抽象方法。  
C、不能用抽象类去创建对象。  
D、接口和抽象类是同一个概念。
- 7、在 Java 中，要想让一个类继承另一个类，可以使用以下哪个关键字？  
A、inherits      B、implements      C、extends      D、modifies
- 8、System 类位于以下哪个包中？



A、java.io    B、java.util    C、java.awt    D、java.lang

9、已知类的继承关系如下：

```
class Employee;
class Manager extends Employee;
class Director extends Employee;
则以下语句能通过编译的有哪些？
```

A、 Employee e=new Manager();    B、 Director d=new Manager();  
C、 Director d=new Employee();    D、 Manager m=new Director();

10、编译运行下面的程序，结果是什么？

```
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.print("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，但会产生运行时异常

#### 四、分析题

阅读下面的程序，分析代码是否能编译通过，如果能编译通过，请列出运行的结果。如果不能编译通过，请说明原因。

代码一：

```
public class Test01 {
    public static void main(String[] args) {
        try {
            int x = 2 / 0;
            System.out.println(x);
        } catch (Exception e) {
            System.out.println("进入 catch 代码块");
        } finally {
            System.out.println("进入 finally 代码块");
        }
    }
}
```

代码二：

```
final class Animal {
```

```
        public final void shout() {  
            // 程序代码  
        }  
    }  
    class Dog extends Animal {  
        public void shout() {  
            // 程序代码  
        }  
    }  
    class Test02 {  
        public static void main(String[] args) {  
            Dog dog=new Dog();  
        }  
    }  
}
```

代码三:

```
class Animal {  
    void shout() {  
        System.out.println("动物叫！");  
    }  
}  
class Dog extends Animal {  
    void shout() {  
        super.shout();  
        System.out.println("汪汪……");  
    }  
}  
public class Test03 {  
    public static void main(String[] args) {  
        Animal animal = new Dog();  
        animal.shout();  
    }  
}
```

代码四:

```
interface Animal {  
    void breathe();  
    void run();  
    void eat(){};  
}  
class Dog implements Animal {  
    public void breathe() {  
        System.out.println("I'm breathing")  
    }  
    public void eat() {
```

```
        System.out.println("I'm eathing")
    }
}

public class test04 {
    public static void main(String [] args) {
        Dog dog = new Dog();
        dog.breathe();
        dog.eat();
    }
}
```

## 五、简答题

- 1、什么是方法重写？
- 2、什么是多态？
- 3、抽象类和接口的区别？

## 六、编程题

请按照题目的要求编写程序并给出运行结果。

- 1、设计一个学生类 `Student` 和它的一个子类 `Undergraduate`，要求如下：
  - 1) `Student` 类有 `name`(姓名)、和 `age`(年龄)属性，一个包含两个参数的构造方法，用于给 `name` 和 `age` 属性赋值，一个 `show()`方法打印 `Student` 的属性信息
  - 2) 本科生类 `Undergraduate` 增加一个 `degree`(学位)属性。有一个包含三个参数的构造方法，前两个参数用于给继承的 `name` 和 `age` 属性赋值，第三个参数给 `degree` 专业赋值，一个 `show()`方法用于打印 `Undergraduate` 的属性信息
  - 3) 在测试类中分别创建 `Student` 对象和 `Undergraduate` 对象，调用它们的 `show()`。
- 2、设计一个 `Shape` 接口和它的两个实现类 `Square` 和 `Circle`，要求如下：
  - 1) `Shape` 接口中有一个抽象方法 `area()`，方法接收有一个 `double` 类型的参数，返回一个 `double` 类型的结果
  - 2) `Square` 和 `Circle` 中实现了 `Shape` 接口的 `area()`抽象方法，分别求正方形和圆形的面积并返回在测试类中创建 `Square` 和 `Circle` 对象，计算边长为 2 的正方形面积和半径为 3 的圆形面积
- 3、自定义一个异常类 `NoThisSoundException` 和 `Player` 类，在 `Player` 的 `play()`方法中使用自定义异常，要求如下：
  - 1) `NoThisSongException` 继承 `Exception` 类，类中有一个无参和一个接收一个 `String` 类型参数的构造方法，构造方法中都使用 `super` 关键字调用父类的构造方法。
  - 2) `Player` 类中定义一个 `play(int index)`方法，方法接收一个 `int` 类型的参数，表示播放歌曲的索引，当 `index>10` 时，`play()`方法用 `throw` 关键字抛出 `NoThisSongException` 异常，创建异常对象时，调用有参的构造方法，传入“您播放的歌曲不存在”。
  - 3) 在测试类中创建 `Player` 对象，并调用 `play()`方法测试自定义的 `NoThisSongException` 异常，使用 `try...catch` 语句捕获异常，调用 `NoThisSongException` 的 `getMessage()`方法打印出异常信息。

## 第 5 章 多线程

### 一、填空题

- 1、一个应用程序中有多条并发执行的线索，每条线索都被称作一个\_\_\_\_\_，它们会交替执行，彼此

间可以进行\_\_\_\_\_。

- 2、在实现多线程的程序时有两种方式，一是通过继\_\_\_\_\_承类，二是通过实现\_\_\_\_\_接口。
- 3、yield()方法只能让相同优先级或者更高优先级、处于\_\_\_\_\_状态的线程获得运行的机会。
- 4、在 Java 语言中，同步方法需要用到关键字\_\_\_\_\_，对于同步方法而言无需指定同步锁，它的同步锁是方法所在的\_\_\_\_\_，也就是\_\_\_\_\_ (关键字)。
- 5、在多任务系统中，每个独立执行的程序称之为\_\_\_\_\_，也就是“正在运行的程序”。
- 6、线程的整个生命周期分为五个阶段，分别是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、和\_\_\_\_\_。
- 7、线程的优先级用 1~10 之间的整数来表示，其中\_\_\_\_\_代表优先级最高，\_\_\_\_\_代表优先级最低。
- 8、在 Thread 类中，提供了一个 start()方法，该方法用于\_\_\_\_\_，当新线程启动后，系统会自动调用\_\_\_\_\_方法。
- 9、要想解决线程间的通信问题，可以使用\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_方法。
- 10、要将某个线程设置为后台线程，需要调用该线程的\_\_\_\_\_方法，该方法必须在\_\_\_\_\_方法之前调用。

## 二、判断题

- 1、当我们创建一个线程对象时，该对象表示的线程就立即开始运行。
- 2、如果前台线程全部死亡，后台线程也会自动死亡。
- 3、同步代码块中的锁对象可以是任意类型的对象。
- 4、静态方法不能使用 synchronized 关键字来修饰。
- 5、线程结束等待或者阻塞状态后，会进入运行状态。

## 三、选择题

- 1、Thread 类位于下列哪个包中？  
A、java.io            B、java.lang            C、java.util            D、java.awt
- 2、关于线程的创建过程，下面四种说法正确的有哪些？(多选)  
A、定义 Thread 类的子类，重写 Thread 类的 run()方法，创建该子类的实例对象，调用对象的 start()方法  
B、定义 Thread 类的子类，重写 Thread 类的 run()方法，创建该子类的实例对象，调用对象的 run()方法  
C、定义一个实现 Runnable 接口的类并实现 run()方法，创建该类实例对象，将其作为参数传递给 Thread 类的构造方法来创建 Thread 对象，调用 Thread 对象的 start()方法  
D、定义一个实现 Runnable 接口的类并实现 run()方法，创建该类对象，然后调用 run()方法
- 3、对于通过实现 Runnable 接口创建线程，下面说法正确的有哪些？(多选)  
A、适合多个相同程序代码的线程去处理同一个资源的情况  
B、把线程同程序代码、数据有效的分离，很好的体现了面向对象的设计思想  
C、可以避免由于 Java 的单继承带来的局限性  
D、编写简单，可以不通过 Thread 类直接创建线程
- 4、对于线程的生命周期，下面四种说法正确的有哪些？(多选)  
A、调用了线程的 start()方法，该线程就进入运行状态  
B、线程的 run()方法运行结束或被未 catch 的 InterruptedException 等异常终结，那么该线程进入死亡状态  
C、线程进入死亡状态，但是该线程对象仍然是一个 Thread 对象，在没有被垃圾回收器回收之前仍可以像引用其它对象一样引用它  
D、线程进入死亡状态后，调用它的 start()方法仍然可以重新启动
- 5、对于死锁的描述，下面四种说法正确有哪些？(多选)  
A、当两个线程互相等待对方释放同步锁时会发生死锁

- B、Java 虚拟机没有检测和处理死锁的措施
  - C、一旦出现死锁，程序会发生异常
  - D、处于死锁状态的线程处于阻塞状态，无法继续运行
- 6、下面四个选项中，哪些是线程进入阻塞状态的原因？(多选)
- A、线程试图获取某个对象的同步锁，而该锁被其它线程持有
  - B、线程调用了另一个线程的 `join()` 方法
  - C、当线程调用了一个阻塞式的 IO 方法
  - D、线程调用了 `setDaemon(boolean b)` 方法
- 7、线程调用 `sleep()` 方法后，该线程将进入以下哪种状态？
- A、就绪状态
  - B、运行状态
  - C、阻塞状态
  - D、死亡状态
- 8、在以下哪种情况下，线程进入就绪状态？
- A、线程调用了 `sleep()` 方法时
  - B、线程调用了 `join()` 方法
  - C、线程调用了 `yield()` 方法时
  - D、线程调用了 `notify()` 方法
- 9、下面四个选项中，对线程同步的目的描述正确的有哪些？(多选)
- A、锁定资源，使同一时刻只有一个线程去访问它，防止多个线程操作同一个资源引发错误
  - B、提高线程的执行效率
  - C、让线程独占一个资源
  - D、让多个线程同时使用一个资源
- 10、对于 `wait()` 方法，下面说法正确的是？(多选)
- A、`wait()` 方法的调用者是同步锁对象
  - B、`wait()` 方法使线程进入等待状态
  - C、调用同一锁对象的 `notify()` 或 `notifyAll()` 方法可以唤醒调用 `wait()` 方法等待的线程
  - D、调用 `wait()` 方法的线程会释放同步锁对象

#### 四、程序分析题

阅读下面的程序，分析代码是否能编译通过，如果能编译通过，请列出运行的结果。如果不能编译通过，请说明原因。

##### 1.代码一：

```
class RunHandler {
    public void run(){
        System.out.println("run");
    }
}

public class Test {
    public static void main(String [] args) {
        Thread t = new Thread(new RunHandler());
        t.start();
    }
}
```

##### 2.代码二：

```
public class A extends Thread{
    protected void run() {
        System.out.println("this is run()");
    }
}
```

```
public static void main(String[] args) {  
    A a = new A();  
    a.start();  
}  
}
```

### 3.代码三:

```
public class Test{  
    private Object obj=new Object();  
    public synchronized void a(){  
        try {  
            obj.wait();  
            System.out.println("waiting");  
        } catch (InterruptedException e) {  
            System.out.println("Exception");  
        }  
    }  
    public static void main(String[] args) {  
        new Test().a();  
    }  
}
```

### 4.为了使下面的程序能够输出“Thread is running”，请在在空格处填上相应的代码。

```
public class B implements Runnable {  
    public static void main(String[] args) {  
        Thread t = new Thread(new B());  
        _____ //此处填空  
    }  
    public void run() {  
        System.out.println("Thread is running");  
    }  
    public void go() {  
        start(1);  
    }  
    public void start(int i) {  
    }  
}
```

## 五、简答题

- 1、java 中有几种方法可以实现一个线程？请举例说明。
- 2、sleep() 和 wait() 有什么区别？

## 六、编程题

请按照题目的要求编写程序并给出运行结果。

- 1、通过继承 Thread 类的方式创建两个线程，在 Thread 构造方法中指定线程的名字，并将这两个线程的名字打印出来。
- 2、通过实现 Runnable 接口的方式创建一个新线程，要求 main 线程打印 100 此次“main”，新线程打印 50 次“new”。

- 3、模拟传智播客的三个老师同时分发 80 份学习笔记，每个老师相当于一个线程。
- 4、编写 10 个线程，第一个线程从 1 加到 10，第二个线程从 11 加到 20...第十个线程从 91 加到 100，最后再把十个线程结果相加。

## 第 6 章 JavaAPI

### 一、填空题

- 1、在 Java 中定义了两个类来封装对字符串的操作，它们分别是\_\_\_\_\_和\_\_\_\_\_。
- 2、Java 中的操作日期的类有\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_等。
- 3、在程序中若想取得一个 Runtime 实例，则可以调用 Runtime 类的静态方法\_\_\_\_\_。
- 4、Math 类中用于计算所传递参数平方根的方法是\_\_\_\_\_。
- 5、Java 中专门用于将日期格式化为字符串的类是\_\_\_\_\_。
- 6、Math 类中有两个静态常量 PI 和 E，分别代表数学常量\_\_\_\_\_和\_\_\_\_\_。
- 7、Java 中的用于产生随机数的类是\_\_\_\_\_，它位于\_\_\_\_\_包中。
- 8、String 类中用于返回字符串字符个数的方法是\_\_\_\_\_。
- 9、System 类中所提供的属性和方法都是\_\_\_\_\_的，想要引用这些属性和方法，直接使用 System 类调用即可。
- 10、已知 sb 为 StringBuffer 的一个实例，且 sb.toString() 的值为 "abcde"，则执行 sb.reverse()后，sb.toString()的值为\_\_\_\_\_。

### 二、判断题

- 1、String 对象和 StringBuffer 对象都是字符串变量，创建后都可以修改。
- 2、用运算符“==”比较字符串对象时，如果两个字符串的值相同，结果为 true。
- 3、System 类中的 currentTimeMillis()方法返回一个 long 类型的值。
- 4、Date、Calendar 以及 DateFormat 类都位于 java.util 包中。
- 5、String 类的方法 replace(CharSequence srt1, CharSequence srt2)返回一个新的字符串，它是通过用 srt2 替换此字符串中出现的所有 srt1 得到的。

### 三、选择题。

- 1、先阅读下面的程序片段：

```
String str = "abccdefcdh";
String[] arr = str.split("c");
System.out.println(arr.length);
```

程序执行后，打印的结果是几？

- A、2 个      B、3 个      C、4 个      D、5 个
- 2、以下都是 Math 类的常用方法，其中用于计算绝对值的方法是哪个？  
A、ceil()      B、floor()      C、abs()      D、random()
- 3、Random 对象能够生成以下哪种类型的随机数？  
A、int      B、string      C、double      D、A 和 C
- 4、String s = "abcdedcba";则 s.substring(3, 4)返回的字符串是以下选项中的哪个？  
A、cd      B、de      C、d      D、e
- 5、假若 indexOf()方法未能找到所指定的子字符串，则返回以下选项中的哪个？  
A、false      B、0      C、-1      D、以上答案都不对
- 6、要产生[20, 999]之间的随机整数可以使用以下哪个表达式？  
A. (int)(20+Math.random()\*97)  
B. 20+(int)(Math.random()\*980)



- C. (int)Math.random()\*999  
D. 20+(int)Math.random()\*980

7、以下 Math 类的方法中，-4.4 通过哪个方法运算后，结果为-5.0?

- A. round() B. min() C. floor() D. ceil()

8、下面的程序段执行后，输出的结果是以下哪个选项?

```
StringBuffer buf=new StringBuffer("Beijing2008");
buf.insert(7,"@");
System.out.println(buf.toString());
```

- A、Beijing@2008 B、@Beijing2008 C、Beijing2008@ D、Beijing#2008

9、阅读下面的程序:

```
public class test {
    public static void main(String args[]) {
        int i;
        float f = 2.3f;
        double d = 2.7;
        i = ((int)Math.ceil(f)) * ((int)Math.round(d));
        System.out.println(i);
    }
}
```

程序执行后，运行结果为以下哪个选项?

- A、9 B、5 C、6 D、6.1

10、先阅读下面的程序片段:

```
String str1=new String("java");
String str2=new String("java");
StringBuffer str3=new StringBuffer("java");
```

对于上述定义的变量，以下表达式的值为 true 的是哪个?

- A、str1==str2; B、str1.equals(str2); C、str1==str3; D、以上都不对

四、程序分析题。

阅读下面的程序，分析代码是否能编译通过，如果能编译通过，请列出运行的结果。如果不能编译通过，请说明原因。

代码一:

```
public class A {
    public static void main(String[] args) {
        System.out.println(Math.abs(-5));
        System.out.println(Math.ceil(6.6));
        System.out.println(Math.floor(-7.8));
        System.out.println(Math.round(-4.9));
        System.out.println(Math.max(8.1, -8.1));
        System.out.println(Math.min(6.1, -6.1));
    }
}
```

代码二:

```
public class B {
    public static void main(String[] args) {
```

```
String s = "dfferghuklmbdfd";
System.out.println("str.length():" + s.length());
System.out.println("str.charAt(0):" + s.charAt(0));
System.out.println("lastIndexOf(m):" + s.lastIndexOf('m'));
System.out.println("substring(2,4):" + s.substring(2, 4));
System.out.println("indexOf(g):" + s.indexOf('g'));
}
}
```

## 五、简答题

- 1、String 和 StringBuffer 有什么区别？
- 2、Date 和 Calender 类有什么区别和联系？

## 六、编程题

请按照题目的要求编写程序并给出运行结果

- 1、编写一个程序，实现字符串大小写的转换并倒序输出，要求如下
  - 1) 使用 for 循环将字符串“HelloWorld”从最后一个字符开始遍历。
  - 2) 遍历的当前字符如果是大写字符，就使用 toLowerCase()方法将其转换为小写字符，反之则使用 toUpperCase() 方法将其转换为大写字符
  - 3) 定义一个 StringBuffer 对象，调用 append()方法依次添加遍历的字符，最后调用 StringBuffer 对象的 toString()方法，并将得到的结果输出。
- 2、计算从今天算起，100 天以后是几月几号，并格式化成为 xxxx 年 x 月 x 日的形式打印出来。  
提示：
  - 1) 调用 Calendar 类的 add()方法计算 100 天后的日期
  - 2) 调用 Calendar 的 getTime() 方法返回 Date 类型对象
  - 2) 使用 FULL 格式的 DateFormat 对象，调用 format() 方法格式化 Date 对象
- 3、利用 Random 类来产生 5 个 20~50 之间的随机整数。  
提示：[n-m](n、m 均为整数，n<m)之间的随机数的公式为 n+(new Random()).nextInt(m-n+1)。

# 第 7 章 集合类

## 一、填空题

- 1、JDK 中提供了一系列可以存储任意对象的类，统称为\_\_\_\_\_。
- 2、在创建 TreeSet 对象时，可以传入自定义的比较器，自定义比较器需要实现\_\_\_\_\_接口。
- 3、Collection 有两个子接口分别是 List 和 Set，List 集合的特点是\_\_\_\_\_，Set 集合的特点是\_\_\_\_\_。
- 4、使用 Iterator 遍历集合时，首先需要调用\_\_\_\_\_方法判断是否存在下一个元素，若存在下一个元素，则调用\_\_\_\_\_方法取出该元素。
- 5、集合按照存储结构的不同可分为单列集合和双列集合，单列集合的根接口是\_\_\_\_\_，双列集合的根接口是\_\_\_\_\_。
- 6、Map 集合中的元素都是成对出现的，并且都是以\_\_\_\_\_、\_\_\_\_\_的映射关系存在。
- 7、Iterator 有一个子类，不仅可以对集合进行从前向后遍历，还可以从后向前遍历，该类是\_\_\_\_\_。
- 8、List 集合的主要实现类有\_\_\_\_\_、\_\_\_\_\_，Set 集合的主要实现类有\_\_\_\_\_、\_\_\_\_\_Map 集合的主要实现类有\_\_\_\_\_、\_\_\_\_\_。

- 9、Map 集合中存储元素需要调用\_\_\_\_\_方法，要想根据该集合的键获取对应的值需要调用\_\_\_\_\_方法。
- 10、java.util 包中提供了一个专门用来操作集合的工具类，这个类是\_\_\_\_\_，还提供了一个专门用于操作数组的工具类，这个类是\_\_\_\_\_。

## 二、判断题

- 1、Set 集合是通过键值对的方式来存储对象的。
- 2、集合中不能存放基本数据类型，而只能存放引用数据类型。
- 3、如果创建的 TreeSet 集合中没有传入比较器，则该集合中存入的元素需要实现 Comparable 接口。
- 4、使用 Iterator 迭代集合元素时，可以调用集合对象的方法增删元素。
- 5、LinkedList 在内部维护了一个双向循环链表，每一个元素节点都包含前一个元素节点和后一个元素节点的引用。

## 三、选择题

- 1、要想保存具有映射关系的数据，可以使用以下哪些集合？（多选）  
A、ArrayList      B、TreeMap      C、HashMap      D、TreeSet
- 2、Java 语言中，集合类都位于哪个包中？  
A、java.util      B、java.lang      C、java.array      D、java.collections
- 3、使用 Iterator 时，判断是否存在下一个元素可以使用以下哪个方法？  
A、next()      B、hash()      C、hasPrevious()      D、hasNext()
- 4、关于 foreach 循环的特点，以下说法哪些是正确的？（多选）  
A、foreach 循环在遍历集合时，无需获得容器的长度  
B、foreach 循环在遍历集合时，无需循环条件，也无需迭代语句  
C、foreach 循环在遍历集合时非常繁琐  
D、foreach 循环的语法格式为：for(容器中元素类型 临时变量 : 容器变量)。
- 5、在程序开发中，经常会使用以下哪个类来存储程序中所需的配置？  
A、HashMap      B、TreeSet      C、Properties      D、TreeMap
- 6、使用 Enumeration 遍历集合时，需要使用以下哪些方法？（多选）  
A、hasMoreElements()      B、nextElement()      C、next()      D、hashNext()
- 7、要想集合中保存的元素没有重复并且按照一定的顺序排列，可以使用以下哪个集合？  
A、LinkedList      B、ArrayList      C、hashSet      D、TreeSet
- 8、下列哪些说法是正确的？（多选）  
A、LinkedList 集合在增删元素时效率较高  
B、ArrayList 集合在查询元素时效率较高  
C、HashMap 不允许出现一对 null 键 null 值  
D、HashSet 集合中元素可重复并且无序
- 9、以下哪些方法是 LinkedList 集合中定义的？（多选）  
A、getLast()      B、getFirst()      C、remove (int index)      D、next()
- 10、获取单列集合中元素的个数可以使用以下哪个方法？  
A、length()      B、size()      C、get(int index)      D、add(Object obj)

## 四、分析题

阅读下面的程序，分析代码是否能编译通过，如果能编译通过，请列出运行的结果。如果不能编译通过，请说明原因。

代码一：

```
import java.util.*;
public class Test01 {
    public static void main(String[] args) {
```

```
TreeSet ts = new TreeSet();
ts.add("b");
ts.add("a");
ts.add("c");
ts.add("c");
Iterator it = ts.iterator();
while(it.hasNext()) {
    System.out.println(it.next());
}
}
```

代码二:

```
import java.util.ArrayList;
public class Test02 {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();
        list.add("a");
        list.add("b");
        list.add("c");
        for (String obj : list) {
            System.out.println(obj);
        }
    }
}
```

代码三:

```
import java.util.*;
public class Test03 {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();
        list.add("demo_1");
        list.add("demo_2");
        list.add("demo_3");
        ListIterator it = list.listIterator();
        while(it.hasPrevious()) {
            Object obj = it.previous();
            System.out.print(obj+" ");
        }
    }
}
```

代码四:

```
import java.util.*;
import java.util.Map.*;
public class Test04 {
    public static void main(String[] args) {
```

```
Map map = new HashMap();
map.put(1, "Tom");
map.put(2, "Lucy");
map.put(3, "Annie");
Set keySet = map.keySet();
Iterator it = keySet.iterator();
while (it.hasNext()) {
    Object key = it.next();
    System.out.println(key);
    map.remove(key);
}
}
```

## 五、简答题

- 1、什么是集合，请列举集合中常用的类和接口？
- 2、集合中的 List、Set、Map 有什么区别？
- 3、请说说 Collection 和 Collections 的有什么区别？

## 六、编程题

请按照题目的要求编写程序并给出运行结果。

- 1、使用 ArrayList 集合，对其添加 10 个不同的元素，并使用 Iterator 遍历该集合。

提示：

- 1) 使用 add()方法将元素添加到 ArrayList 集合中。
- 2) 调用集合的 iterator()方法获得 Iterator 对象，并调用 Iterator 的 hasNext()和 next()方法，迭代出集合中的所有元素。

- 2、在 HashSet 集合中添加三个 Person 对象，把姓名相同的人当做同一个人，禁止重复添加。

提示：

- 1) Person 类中定义 name 和 age 属性，重写 hashCode()方法和 equals()方法，针对 Person 类的 name 属性进行比较，如果 name 相同，hashCode()方法的返回值相同，equals 方法返回 true。

- 3、选择合适的 Map 集合保存 5 位学员的学号和姓名，然后按学号的自然顺序的倒序将这些键值对一一打印出来。

提示：

- 1) 创建 TreeMap 集合。
- 2) 使用 put()方法将学号(“1”、“2”、“3”、“4”、“5”)和姓名(“Lucy”、“John”、“Smith”、“Aimee”、“Amanda”)存储到 Map 中，存的时候可以打乱顺序观察排序后的效果。
- 3) 使用 map.keySet()获取键的 Set 集合。
- 4) 使用 Set 集合的 iterator()方法获得 Iterator 对象用于迭代键。
- 5) 使用 Map 集合的 get()方法获取键所对应的值。

# 第 8 章 IO（输入输出）

## 一、填空题

- 1、Java 中的 IO 流，按照传输数据不同，可分为\_\_\_\_和\_\_\_\_。
- 2、在 Java 中，\_\_\_\_类用于操作磁盘中文件和目录，位于\_\_\_\_包中。

- 3、在 Java 中，\_\_\_类用来把两个或更多的 InputStream 输入流对象合并为单个 InputStream 输入流对象使用。
- 4、Java 中提供了一个类\_\_\_，它不但具有读写文件的功能，并且可以随机地从文件的任何位置开始执行读写数据的操作。
- 5、在 Java 中，能实现线程间通信的流是\_\_\_。
- 6、Java 中提供了一个可以在读文件的同时记录行号的类，这个类是\_\_\_，它是\_\_\_的直接子类，它通过\_\_\_和 \_\_\_方法设置和获取当前行号。
- 7、InputStreamReader 类是用于将\_\_\_转换为\_\_\_。
- 8、System.out 是\_\_\_类的对象，称为标准输出流，调用 System 类的\_\_\_方法可以实现标准输出流的重定向。
- 9、Java 中一个字符占用两个字节，所有字符采用的都是\_\_\_码表。
- 10、BufferedWriter 的\_\_\_方法可以写入一个换行符。

## 二、判断题

- 1、如果一个 File 表示目录下有文件或者子目录，调用 delete()方法也可以将其删除。
- 2、装饰设计模式中，装饰对象应该包含一个被装饰对象的引用。
- 3、使用 ObjectInputStream 与 ObjectOutputStream 类来读取或存储的对象必须要实现 Serializable 接口，否则程序将出现 NotSerializableException 异常。
- 4、InputStream 类的 close()方法是用于关闭流并且释放流所占的系统资源。
- 5、一般来说，把字符转换成计算机识别的字节序列称为解码，而把字节序列转换为普通人能看懂的明文字符称为编码。

## 三、选择题

- 1、下面选项中，哪些是标准输入输出流？（多选）  
A、System.In B、System.Out C、InputStream D、OutputStream
- 2、以下选项中，哪个是 FileOutputStream 的父类？  
A.File B.FileOutput C.OutputStream D.InputStream
- 3、File 类中以字符串形式返回文件绝对路径的方法是？  
A、getParent() B、getName()  
C、getAbsolutePath() D、getPath()
- 4、下面哪些是常用的字符码表？（多选）  
A、ASCII B、UTF-8 C、ISO8859-1 D、GB2312
- 5、以下创建 RandomAccessFile 类实例对象的代码，哪些是正确的？（多选）  
A、RandomAccessFile(new File("D:\\itcast\\dir1\\test.java"),"rw")  
B、RandomAccessFile("D:\\itcast\\dir1\\test.java","r")  
C、RandomAccessFile("D:\\itcast\\dir1\\test.java")  
D、RandomAccessFile("D:\\itcast\\dir1\\test.java","wr")
- 6、以下哪些属于 InputStream 类的方法？（多选）  
A、int read(byte[]) B、void flush() C、void close() D、available()
- 7、以下选项中，哪个流中使用了缓冲区技术？  
A、BufferedOutputStream B、FileInputStream C、DataOutputStream D、FileReader
- 8、以下选项中，哪个是 File 类 delete()方法返回值的类型？  
A、boolean B、int C、String D、Integer
- 9、以下选项中，哪个文件操作类可以实现一次读入多个文件？  
A、FileReader B、BufferedReader C、FileInputStream D、SequenceInputStream
- 10、以下对 File 类的 public boolean isFile()方法的描述，哪个是正确的？

- A、判断该 File 对象所对应的是否是文件。
- B、判断该 File 对象所对应的是否是目录。
- C、返回文件的最后修改时间。
- D、在当前目录下生成指定的目录。

#### 四、程序填空题

1、阅读以下代码，并将空处填写完整。

```
import java.io.*;

public class Test1 {
    public static void main(String args[])throws Exception{
        int a=4;
        BufferedReader br=new BufferedReader(new _____(System.in));
        System.out.println("请输入一个数字");
        String input=_____;
        int b=Integer.parseInt(input);
        if(b>a){
            int sum=b/a;
            System.out.println(sum);
        }else{
            System.out.println("输入错误");
        }
    }
}
```

当输入的数字是 8 时，打印输出的结果是\_\_\_\_\_。

2、以下是使用 FileInputStream 类与 FileOutputStream 类复制文件。阅读以下代码，并将空处填写完整。

```
import java.io.*;

class Test2 {
    public static void main(String[] args) {
        String file1, file2;
        int ch = 0;
        file1 = "readme.txt";
        file2 = "readme.bak";
        try {
            FileInputStream fis = new _____;
            _____ fos=new FileOutputStream(file2);
            int size=_____;
            System.out.println("字节有效数、"+size);
            while ((ch = fis.read()) != -1) {
                System.out.write(ch);
                fos.write(ch);
            }
            fis.close();
            fos.close();
        } catch (IOException e) {
            System.out.println(e.toString());
        }
    }
}
```



```
}  
}  
}
```

## 五、简答题

- 1、简述流的概念。
- 2、Java 流被分为字节流、字符流两大流类，两者有什么区别？
- 3、简要说明管道流。

## 六、编程题

按照题目需求，编写程序并运行

- 1、编写一个程序，分别使用字节流和字符流拷贝一个文本文件。

提示：

- 1) 使用 `FileInputStream`、`FileOutputStream` 和 `FileReader`、`FileWriter` 分别进行拷贝
- 2) 使用字节流拷贝时，定义一个 1024 长度的字节数组作为缓冲区，使用字符流拷贝使用使用 `BufferedReader` 和 `BufferedWriter` 包装流进行包装

- 2、某人在玩游戏的时候输入密码 123456 后成功进入游戏（输错 5 次则被强行退出），要求用程序实现密码验证的过程。

提示：

- 1) 使用 `System.in` 包装为字符流读取键盘输入。
- 2) `BufferedReader` 对字符流进行包装。调用 `BufferedReader` 的 `readLine()` 方法每次读取一行。
- 3) 在 for 循环判中判断输入的密码是否为“123456”，如果是则打印“恭喜你进入游戏”，并跳出循环，否则继续循环读取键盘输入。
- 4) 当循环完毕，密码还不正确，则打印“密码错误，结束游戏”，并调用 `System.exit(0)` 方法结束程序。

# 第 9 章 GUI（图形用户界面）

## 一、填空题

- 1、GUI、java.awt、javax.swing
- 2、适配器
- 3、Graphics
- 4、事件监听器
- 5、JComponent
- 6、窗体事件、键盘事件、鼠标事件、动作事件
- 7、WindowListener、WindowClosing(WindowEvent e)
- 8、模态对话框、非模态对话框
- 9、setLayout(null)
- 10、JMenuBar、JMenu、JMenuItem

## 二、判断题

- 1、对 2、错 3、对 4、错 5、错

## 三、选择题

- 1、D 2、D 3、ABD 4、A 5、D 6、ABCD 7、ABD 8、ABC 9、D 10、C

## 四、程序分析题

- 1、参考答案

```
extends  
Stdno = new JTextField();  
Name = new JTexxtField();
```

## 2、参考答案

```
card.next(cp);  
cp.add(jbt);
```

## 五、简答题

### 1、参考答案

- 通过实现 XxxListener 接口或者继承 XxxAdapter 类实现一个事件监听器类，并对处理监听动作的方法进行重写
- 创建事件源对象和事件监听器对象
- 调用事件源的 addXxxLisntener()方法，为事件源注册事件监听器对象

### 2、参考答案

AWT 是 Abstract Window ToolKit (抽象窗口工具包)的缩写，这个工具包提供了一套与本地图形界面进行交互的接口。AWT 中的图形函数与操作系统所提供的图形函数之间有着一一对应的关系，当我们利用 AWT 来构件图形用户界面的时候，我们实际上是在利用操作系统所提供的图形库。由于不同操作系统的图形库所提供的功能是不一样的，在一个平台上存在的功能在另外一个平台上则可能不存在。为了实现 Java 语言所宣称的"一次编译，到处运行"的概念，AWT 不得不通过牺牲功能来实现其平台无关性，也就是说，AWT 所提供的图形功能是各种通用型操作系统所提供的图形功能的交集。由于 AWT 是依靠本地方法来实现其功能的，我们通常把 AWT 控件称为重量级控件。

Swing 是在 AWT 的基础上构建的一套新的图形界面系统，它提供了 AWT 所能够提供的所有功能，并且用纯粹的 Java 代码对 AWT 的功能进行了大幅度的扩充。由于在 Swing 中没有使用本地方法来实现图形功能，我们通常把 Swing 控件称为轻量级控件。

AWT 和 Swing 之间的基本区别：AWT 是基于本地方法的 C/C++程序，其运行速度比较快；Swing 是基于 AWT 的 Java 程序，其运行速度比较慢。对于一个嵌入式应用来说，目标平台的硬件资源往往非常有限，而应用程序的运行速度又是项目中至关重要的因素。在这种矛盾的情况下，简单而高效的 AWT 当然成了嵌入式 Java 的第一选择。而在普通的基于 PC 或者是工作站的标准 Java 应用中，硬件资源对应用程序所造成的限制往往不是项目中的关键因素，所以在标准版的 Java 中则提倡使用 Swing，也就是通过牺牲速度来实现应用程序的功能。

## 六、编程题

### 1、参考答案

```
import java.awt.*;  
import java.awt.event.*;import javax.swing.*;  
public class MyMouseHandler extends JFrame {  
    public MyMouseHandler() {  
        final JLabel label = new JLabel("此处显示鼠标右键点击的坐标");  
        label.setOpaque(true);  
        label.setBackground(Color.PINK);  
        this.add(label, BorderLayout.NORTH);  
        this.setSize(300, 200);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.addMouseListener(new MouseAdapter() {  
            public void mouseClicked(MouseEvent e) {  
                if (e.getButton() == e.BUTTON1) {
```

```
        int x = e.getX();
        int y = e.getY();
        String banner = "鼠标当前点击位置的坐标是" + x + "," + y;
        label.setText(banner);
    }
}
});
this.setVisible(true);
}
public static void main(String[] args) {
    new MyMouseHandler();
}
}
```

## 2、参考答案

```
import java.awt.*;
import java.util.*;
import javax.swing.*;
import java.awt.event.*;

public class Information extends JFrame {
    // 窗口 NORTH 部的 JPanel 面板
    private JPanel panel = new JPanel();
    // 爱好标签
    private JLabel lb1 = new JLabel("爱好");
    // 三个表示爱好的 JCheckBox 复选框
    private JCheckBox cb1 = new JCheckBox("羽毛球");
    private JCheckBox cb2 = new JCheckBox("乒乓球");
    private JCheckBox cb3 = new JCheckBox("唱歌");
    // 性别标签
    private JLabel lb2 = new JLabel("性别");
    // 表示性别的 JRadioButton 单选框
    private JRadioButton rb1 = new JRadioButton("男");
    private JRadioButton rb2 = new JRadioButton("女");
    // ButtonGroup 添加 JRadioButton, 实现单选功能
    private ButtonGroup bg = new ButtonGroup();
    // 文本域组件
    private JTextArea area = new JTextArea();
    // 窗口 CENTER 部的 JScrollPane 面板, 其中放置 area 文本域
    private JScrollPane pane = new JScrollPane(area);
    // Set 集合存放选中的兴趣
    private Set<String> hobbies = new HashSet<String>();
    // gender 选中的性别
    private String gender = "";
    // JCheckBox 复选框的事件监听器
    private ActionListener listener1 = new ActionListener() {
```

```
        public void actionPerformed(ActionEvent e) {
            JCheckBox cb = (JCheckBox) e.getSource();
            // 选中的复选框把文本添加到 Set 集合中
            if (cb.isSelected()) {
                hobbies.add(cb.getText());
            } else {
                hobbies.remove(cb.getText());
            }
            print();
        }
    };

    // JRadioButton 单选框的事件监听器
    private ActionListener listener2 = new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JRadioButton jb = (JRadioButton) e.getSource();
            gender = jb.getText();
            print();
        }
    };

    // 打印方法
    private void print() {
        // 清空文本域
        area.setText("");
        // 如果 Set 集合中有元素，打印兴趣
        if (hobbies.size() > 0) {
            area.append("你的兴趣爱好有: ");
            Iterator<String> it = hobbies.iterator();
            while (it.hasNext()) {
                area.append(it.next() + " ");
            }
        }
        // 如果 gender 不为空字符串，打印性别
        if (!"".equals(gender)) {
            area.append("你的性别为: " + gender);
        }
    }

    public Information() {
        // 添加标签、单选和复选按钮
        panel.add(lb1);
        panel.add(cb1);
        panel.add(cb2);
        panel.add(cb3);
        panel.add(lb2);
        panel.add(rb1);
        panel.add(rb2);
    }
}
```

```
bg.add(rb1);
bg.add(rb2);
// 为单选和复选按钮添加事件监听器
cb1.addActionListener(listener1);
cb2.addActionListener(listener1);
cb3.addActionListener(listener1);
rb1.addActionListener(listener2);
rb2.addActionListener(listener2);
// 将 JPanel 面板和 JScrollPane 面板添加到 JFrame 容器中
Container container = this.getContentPane();
container.add(panel, BorderLayout.NORTH);
container.add(pane, BorderLayout.CENTER);
this.pack();
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public static void main(String[] args) {
    new Information();
}
}
```

### 3、参考答案

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class MyMenu extends JFrame implements ActionListener {
    JLabel label = new JLabel("请选择菜单", JLabel.CENTER);
    JMenuItem aaMenuItem, baMenuItem;

    MyMenu() {
        JMenuBar menuBar = new JMenuBar();
        JMenu aMenu = new JMenu("菜单 A");
        JMenu bMenu = new JMenu("菜单 B");
        JMenuItem aaMenuItem = new JMenuItem("菜单项 AA");
        JMenuItem abMenuItem = new JMenuItem("菜单项 AB");
        JMenuItem baMenuItem = new JMenuItem("菜单项 BA");
        menuBar.add(aMenu);
        menuBar.add(bMenu);
        aMenu.add(aaMenuItem);
        aMenu.addSeparator();
        aMenu.add(abMenuItem);
        bMenu.add(baMenuItem);
        aaMenuItem.addActionListener(this);
        abMenuItem.addActionListener(this);
        baMenuItem.addActionListener(this);
        setJMenuBar(menuBar);
    }
}
```

```
        getContentPane().add(label, BorderLayout.CENTER);
    }

    public void actionPerformed(ActionEvent e) {
        JMenuItem source = (JMenuItem) (e.getSource());
        label.setText("选择了菜单:" + source.getText());
        label.setHorizontalAlignment(JLabel.CENTER);
    }

    public static void main(String args[]) {
        JFrame frame = new MyMenu();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);
        frame.setVisible(true);
    }
}
```

### 一、填空题。

- 1、在 Java 中，图形用户界面简称\_\_\_\_\_，它的组件包含在\_\_\_\_\_和\_\_\_\_\_这两个包中。
- 2、为了避免实现监听器中定义的所有方法，造成代码的臃肿，在 JDK 中提供了一些\_\_\_\_\_类，这些类实现了接口所有的方法，但是方法中没有任何代码，属于一种空实现。
- 3、在 Java 中，\_\_\_\_\_类相当于一个抽象的画笔对象，使用它可以在组件上绘制图形。
- 4、\_\_\_\_\_负责监听事件源上发生的事件，并对各种事件做出响应处理。
- 5、大部分的 Swing 组件都是\_\_\_\_\_类的直接或者间接子类，其名称都是在原来 AWT 组件名称前加上字母 J。
- 6、AWT 事件有多种多样，大致可以分为\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_等。
- 7、如果点击 Frame 窗口右上角的关闭按钮能将其关闭，那么这个 Frame 窗口添加了\_\_\_\_\_监听器，并实现了监听器的\_\_\_\_\_方法。
- 8、对话框可以分为\_\_\_\_\_和\_\_\_\_\_两种。
- 9、每一个容器都有一个默认的布局管理器，如果不希望通过布局管理器对容器进行布局，可以调用容器的\_\_\_\_\_方法将其取消。
- 10、创建下拉式菜单需要使用三个组件，分别是\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。

### 二、判断题。

- 1、容器（Container）是一个可以包含基本组件和其它容器的组件。
- 2、可以通过实现 ActionListener 接口或者继承 ActionAdapter 类来实现动作时间监听器。
- 3、CardLayout 布局管理器将界面看做一系列卡片，在任何时候只有其中一张卡片是可见的。
- 4、非模态对话框是指用户需要等到处理完对话框后才能继续与其它窗口进行交互。
- 5、JFrame 的默认布局管理器是 FlowLayout。

### 三、选择题

- 1、下面四个组件中哪一个不是 Component 的子类？  
A、Button      B、Dialog      C、Label      D、MenuBar
- 2、每一个 GUI 程序中必须包含一个什么组件？  
A、按钮      B、标签      C、菜单      D、容器
- 3、下面四个选项中，哪些是事件处理机制中的角色？(多选)  
A、事件      B、事件源  
C、事件接口      D、事件监听器
- 4、当鼠标按键被释放时，会调用以下哪个事件处理器方法？

- A、mouseReleased()                      B、mouseUp()  
C、mouseOff()                            D、mouseLetGo()
- 5、ActionEvent 的对象会被传递给以下哪个事件处理器方法？  
A、addChangeListener()                  B、addActionListener()  
C、stateChanged()                        D、actionPerformed()
- 6、AWT 中，常用的布局管理器包括哪些？(多选)  
A、FlowLayout 布局管理器                B、BorderLayout 布局管理器  
C、CardLayout 布局管理器                D、GridLayout 布局管理器
- 7、下面哪些是 FlowLayout 类中表示对齐方式的常量？(多选)  
A、FlowLayout.LEFT                        B、FlowLayout.CENTER  
C、FlowLayout.VERTICAL                    D、FlowLayout.RIGHT
- 8、下面对与 Swing 的描述，正确的有哪些？(多选)  
A、Swing 是在 AWT 基础上构建的一套新的图形界面系统  
B、Swing 提供了 AWT 所能够提供的功能  
C、Swing 组件是用 Java 代码来实现的  
D、Swing 组件都是重量级组件
- 9、下面四对 AWT 和 Swing 对应组件中，错误的是？  
A、Button 和 JButton  
B、Dialog 和 JDialog  
C、MenuBar 和 JMenuBar  
D、ComboBox 和 JComboBox
- 10、使用下面哪个组件可以接收用户的输入信息？  
A、JButton                                  B、JLabel  
C、JTextField                                D、以上都可以

#### 四、程序分析题

阅读下面的程序以及注释的要求，在空格处填上相应的代码  
代码一：

```
import java.awt.*;  
import javax.swing.*;  
public class MyLayout _____/*此处填空*/ JFrame{  
    JLabel labelNo,labelName,labelGender;  
    JTextField stdno,name,gender;  
    int x=0,y=0,w,h;  
    Container cp=getContentPane();  
    public MyLayout() {  
        setLayout(null);  
        学号=new JLabel("labelNo、",JLabel.CENTER);  
        姓名=new JLabel("labelName、",JLabel.CENTER);  
        性别=new JLabel("labelGender、",JLabel.CENTER);  
        _____ // 此处填空  
        _____ // 此处填空  
        gender=new JTextField();  
        x=80;y=30;  
        w=100;h=30;
```



```
cp.add(labelNo); cp.add(labelName);cp.add(labelGender);
cp.add(stdno);cp.add(name);cp.add(gender);
labelNo.setBounds(0,y,w,h); stdno.setBounds(x,y,w,h);
labelName.setBounds(0,2*y,w,h);name.setBounds(x,2*y,w,h);
labelGender.setBounds(0,3*y,w,h);gender.setBounds(x,3*y,w,h);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(280,200);
setVisible(true);
}
public static void main(String args[]){
    MyLayout obj=new MyLayout();
}
}
```

代码二:

```
import java.awt.CardLayout;
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
public class MyCardLayout {
    public static void main(String args[]) {
        final JFrame jframe = new JFrame("一个滚动列表的例子");
        final Container panelcp = jframe.getContentPane();
        final JPanel panel = new JPanel();
        final CardLayout card = new CardLayout (20, 20);
        panelcp.setLayout(card);
        jframe.add(panel);
        for (int i = 0; i < 5; i++) {
            JButton jbt = new JButton("jbt" + i);
            jbt.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    // 点击的时候显示下一个按钮
                    _____ // 此处填空
                }
            });
            _____ // 此处填空
        }
        jframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jframe.setSize(150, 200);
        jframe.setVisible(true);
    }
}
```

## 五、简答题。

- 1、请简述 GUI 中实现事件监听的步骤。
- 2、AWT 和 Swing 的区别

## 六、编程题

- 1、编写一个 JFrame 窗口，要求如下：
  - 1) 在窗口的最上方放置一个 JLabel 标签，标签中默认的文本是“此处显示鼠标右键点击的坐标”
  - 2) 为 JFrame 窗口添加一个鼠标事件，当鼠标右键点击窗口时，鼠标的坐标在 JLabel 标签中显示
- 2、编写一个 JFrame 窗口，要求如下：
  - 1) 窗口中的 NORTH 区域中放置一个 JPanel 面板
  - 2) JPanel 面板中从左到右依次放置如下组件：
    - JLabel 标签，标签的文本为“兴趣”
    - 三个 JCheckBox 多选按钮，文本分别为“羽毛球”、“乒乓球”、“唱歌”
    - JLabel 标签，标签的文本为“性别”
    - 两个 JRadioButton 按钮，文本分别为“男”、“女”
  - 3) 窗口的 CENTER 区域放置一个 JScrollPane 容器，容器中放置一个 JTextArea 文本域
  - 4) 当点击多选按钮和单选按钮时，会把选中按钮的文本显示在 JTextArea 文本域中。
- 3、编写一个 JFrame 窗口，要求如下：
  - 1) 窗口包含一个菜单栏和一个 JLabel 标签。
  - 2) 菜单栏中有两个菜单，第一个菜单有两个菜单项，它们之间用分隔符分开，第二个菜单有一个菜单项。
  - 3) JLabel 标签放置在窗口的中间(即 BorderLayout.CENTER)，当点击菜单项的时候，菜单项中的文本显示的 JLabel 标签中。

## 第 10 章 网络编程

## 一、填空题

- 1、TCP 协议的特点是\_\_\_\_\_，即在传输数据前先在\_\_\_\_\_和\_\_\_\_\_建立逻辑连接。
- 2、在计算机中，端口号是用\_\_\_\_\_字节，也就是 16 位的二进制数表示，它的取值范围是\_\_\_\_\_。
- 3、TCP/IP 协议被分为四个层，分别是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- 4、在 JDK 中，IP 地址用\_\_\_\_\_类来表示的，该类提供了许多和 IP 地址相关的操作。。
- 5、使用 UDP 协议开发网络程序时，需要使用两个类，分别是\_\_\_\_\_和\_\_\_\_\_。

## 二、判断题

- 1、由于 UDP 是面向无连接的协议，可以保证数据的完整性，因此在传输重要数据时建议使用 UDP 协议。
- 2、在网络通信中，对数据传输格式、传输速率、传输步骤等作了统一规定，只有通信双方共同遵守这个规定才能完成数据的交互，这种规定称为网络传输协议。
- 3、在创建发送端的 DatagramPacket 对象时，需要指定发送端的目标 IP 地址和端口号。
- 4、IPv4 版本的 IP 地址使用 4 个字节来表示，IPv6 版本的 IP 地址使用 8 个字节来表示。
- 5、使用 TCP 协议通信时，通信的两端以 IO 的方式进行数据的交互。

## 三、选择题

- 1、使用 UDP 协议通信时，需要使用哪个类把要发送的数据打包？  
A、Socket    B、DatagramSocket    C、DatagramPacket    D、ServerSocket
- 2、以下哪个是 serverSocket 类用于接收来自客户端请求的方法？

- A、accept()    B、getOutputStream()    C、receive()    D、get()
- 3、以下说法哪些是正确的？（多选）
- A、TCP 连接中必须要明确客户端与服务端。
- B、TCP 协议是面向连接的通信协议，它提供了两台计算机之间可靠无差错的数据传输。
- C、UDP 协议是面向无连接的协议，可以保证数据的完整性。
- D、UDP 协议消耗资源小，通信效率高，通常被用于音频、视频和普通数据的传输。
- 4、以下哪个类用于实现 TCP 通信的客户端程序？
- A、ServerSocket    B、Socket    C、Client    D、Server
- 5、进行 UDP 通信时，在接收端若要获得发送端的 IP 地址，可以使用 DatagramPacket 的哪个方法？
- A、getAddress()    B、getPort()    C、getName()    D、getData()
- 6、以下哪个方法是 DatagramSocket 类用于发送数据的方法？
- A、receive()    B、accept()    C、set()    D、send()
- 7、在程序运行时，DatagramSocket 的哪个方法会发生阻塞？
- A、send()    B、receive()    C、close()    D、connect()
- 8、TCP 协议的“三次握手”中，第一次握手指的是什么？
- A、客户端再次向服务器端发送确认信息，确认连接。
- B、服务器端向客户端回送一个响应，通知客户端收到了连接请求。
- C、客户端向服务器端发出连接请求，等待服务器确认。
- D、以上答案全部错误

#### 四、简答题

- 1、网络通信协议是什么？
- 2、TCP 协议和 UDP 协议有什么区别？
- 3、Socket 类和 ServerSocket 类各有什么作用？

#### 五、编程题

请按照题目的要求编写程序并给出运行结果。

- 1、使用 InetAddress 类获取本地计算机的 IP 地址和主机名，甲骨文公司(www.oracle.com)主机的 IP 地址。  
提示：
  - 1) 通过 InetAddress.getLocalHost(); 获取本地计算机的 InetAddress 对象
  - 2) 通过 InetAddress.getByName("www.oracle.com"); 获取 Oracle 公司的 InetAddress 对象
- 2、使用 UDP 协议编写一个网络程序，设置接收端程序监听端口为 8001，发送端发送的数据是“hello world”。  
提示：
  - 1) 使用 new DatagramSocket(8001) 构造方法创建接收端的 DatagramSocket 对象，调用 receive() 方法接收数据。
  - 2) 发送端和接收端使用 DatagramPacket 封装数据，在创建发送端的 DatagramPacket 对象时需要指定目标 IP 地址和端口号，端口号要和接收端监听的端口号一致。
  - 3) 发送端使用 send() 方法发送数据。
  - 4) 使用 close() 方法释放 Socket 资源。
- 3、使用 TCP 协议编写一个网络程序，设置服务器程序监听端口为 8002，当与客户端建立连接后，向客户端发送“hello world”，客户端负责将信息输出。  
提示：
  - 1) 使用 ServerSocket 创建服务器端对象，监听 8002 端口，调用 accept() 方法等待客户端连接，当与客户端连接后，调用 Socket 的 getOutputStream() 方法获得输出流对象，输出“hello world”。

- 2) 使用 Socket 创建客户端对象，指定服务器的 IP 地址和监听端口号，与服务器端建立连接后，调用 Socket 的 `getInputStream()` 方法获得输入流对象，读取数据，并打印出来
- 3) 在服务器端和客户端都调用 `close()` 方法释放 socket 资源。