

第一章 JAVA 语言概论

一、选择题

1. 下列不属于 java 语言鲁棒性特点的是:
 - A. java 能检查程序在变异和运行时的错误
 - B. java 能运行虚拟机实现跨平台
 - C. java 自己操纵内存减少了内存出错的可能性
 - D. java 还实现了真数组, 避免了覆盖数据的可能

2. java 语言的执行模式是:
 - A. 全编译型
 - B. 全解释型
 - C. 半编译和半解释型
 - D. 同脚本语言的解释模式

3. 下列关于虚拟机说法错误的是:
 - A. 虚拟机可以用软件实现
 - B. 虚拟机部可以用硬件实现
 - C. 字节码是虚拟机的机器码
 - D. 虚拟机把代码程序与各操作系统和硬件分开

4. java 语言是 1995 年由 () 公司发布的
 - A. Sun

- B. Microsoft
 - C. Borland
 - D. Fox Software
5. 下列不是虚拟机执行过程特点的是
- A. 多线程
 - B. 多线程
 - C. 动态链接
 - D. 异常处理
6. java 以 JVM 为基础，最下层是移植接口，由适配器和（）组成
- A. 网卡
 - B. Java os
 - C. Java 基本类
 - D. Java 应用程序和 applet 小程序
7. java 程序的执行过程中用到一套 JDK 工具，其中 javac.exe 指
- A. java 语言编译器
 - B. java 字节码解释器
 - C. java 文档生成器
 - D. java 类分解器
8. java 的 API 结构中，不属于类库主要包括的核心包的是

- A. java 包
 - B. javax
 - C. javadoc 包
 - D. org 扩展包
9. 每个 java 的编译单元可包含多个类或界面，但是每个编译单元最多只能有（）类或者界面是公共的
- A. 一个
 - B. 两个
 - C. 四个
 - D. 任意多个
10. 在当前的 java 实现中，每个编译单元就是一个以（）为后缀的文件
- A. java
 - B. class
 - C. doc
 - D. exe

二、填空题

1. 1991 年，SUN 公司的 Jame Gosling 和 Bill Joe 等人，为电视、控制烤面包机等家用电器的交互操作开发了一个____软件，它是 java 的前身。
2. java 是一个网络编程语言，简单易学，利用了____的技术基础，但又独立于硬件结构，具有可移植性、健壮性、安全性、高性能。

3. java 可以跨平台的原因是_____
4. JVM 的执行过程有三个特点：多线程，_____, 异常处理；
5. JAVA 程序的安全性体现在多个层次上，在_____, 有语法检查；在解释层上，有字节码校验器、测试代码段格式和规则检查，访问权限和类型转换和法性检查，操作数堆栈的上溢或下溢，代码参数类型合法性等；在平台层上，通过配置策略，可设定访问资源域，而无需区分本地或远程。
6. JAVA 的产品主流操作系统平台是 Solaris、_____和 Macintosh.
7. JAVA 系统运行时，通过_____机制周期性的释放无用对象所使用的内存，完成对象的清除。
8. 在 JAVA 语言中，将后缀名为_____的源代码文件编译后形成后缀名为.class 的字节码文件。
9. JAVA 语言的执行模式是半编译和_____。
10. JAVA 类库具有_____的特点，保证了软件的可移植性。
11. 在编写执行 JAVA 程序的过程中需要用到一些工具，SUN 公司为我们提供了一套 JDK 工具，它主要包括：javac.exe、java.exe、_____, javap.exe、jkb.exe
12. JAVA 程序包括源代码(.java 文件)、_____,由归档工具 jar 生成的.jar 文件、对象状态序列化.ser 文件。
13. 在 JAVA 语言中，为将源代码翻译成_____文件时产生的错误称为编译错误。而将程序在运行中产生的错误称为运行错误。
14. Java Application 应用程序的编写和执行分 3 步进行：编写源代码、编译源代码、_____。
15. 每个 java 应用程序可以包括许多方法，但必须有且只能有一个_____方法。
16. JAVA 源文件中最多只能有一个_____类，其它类的个数不限。
17. JAVA 的体系结构中，最下层是移植接口，上面一层是虚拟机，虚拟机的上层是_____和基本 API，它们都是具有可扩展性。

18. 类库主要包括核心 JAVA 包、_____和 org 扩展包。
19. JAVA 解释器采用生成与体系结构无关的_____指令的技术, 只需安装 JAVA 运行系统, 就可保证 JAVA 程序可以在网络的任何地方运行。
20. JAVA 的体系结构中, 最下层是_____, 由适配器和 JAVA OS 组成, 保证 JAVA 体系结构可以跨平台。

第二章 Java 语言基础

一 选择题

1. 下列 java 标识符, 错误的是 ()

`_sys_var1`

`$change`

`User_name`

`1_file`

- 下列不属于简单数据类型的是 ()

整数类型

类

符点数类型

布尔类型

- 下列属于 JAVA 关键词的是 ()

`TRUE`

`goto`

float

NULL

下列声明和赋值语句错误的是

double w=3.1415;

String str1="bye" ;

float z=6.74567

boolean truth=true;

java 中，八进制数以_____开头。

0x

0

0X

08

自定义类型转换是由按优先关系从低级数据转换为高级数据，优先次序为（）

char-int-long-float-double

int-long-float-double-char

long-float-int-double-char

以上都不对

在 java 中，Integer.MAX_VALUE 表示

浮点类型最大值

整数类型最大值

长整型最大值

以上说法都不对

JAVA 中，数据类型值中不可能出现的符号是（）

d

f

e

/

下列表示单引号的是（）

‘

\\’

\\\\\\’

‘\\

10. 下列语句片断中，four 得值为：（）

```
int three=3;
```

```
char one= ‘1’
```

```
char four=(char)(three+one);
```

3

1

31

4

11. 下列不属于整型变量的类型是

byte

short

float

long

12. int 类型整型变量在内存中的位数为

8

16

32

64

13. 下列数据类型转换，必须进行强制类型转换的是

byte→int

short→long

float→double

int→char

14. java 中，用 () 关键字定义常量

final

`#define`

`float`

`const`

15. 关于变量的作用范围，下列说法错误的是

异常处理参数作用域为整个类

局部变量作用于声明该变量的方法代码段

类变量作用于声明该变量的类

方法参数作用于传递到方法内代码段

16. 下列属于条件运算符的是

`+`

`?:`

`&&`

`>>`

17. 下列程序 `test` 类中的变量 `c` 的最后结果为

```
public class test
{
    public static void main(String args[])
    {
        int a=10;
```

```
int b;

int c;

if(a>50)

{

    b=9;

}

c=b+a;

}

}

10

0

19

编译出错
```

18. 下列程序片断的执行，说法正确的是

```
public class test

{

public static void main (String args[])

{

byte b=100;
```

```
int i=b;

int a=2000;

b=a;

System.out.println(b);

}

}
```

b 的值为 100

b 的值为 2000

第六行出错

第八行出错

19. 00101010 (&) 00010111 语句的执行结果为

00000010

11111111

00111111

11000000

20. 已知 i 为整形变量，关于一元运算 ++i 和 i++，下列说法正确的是

A. ++i 运算将出错

B. 在任何情况下运行程序结果都一样

C. 在任何情况下运行程序结果都不一样

D. 在任何情况下变量 i 的值都增 1

21. 在 java 中语句：37.2%10 的运算结果为

A. 7.2

B. 7

C. 3

D. 0.2

22. 在 java 语句中，运算符&&实现

A. 逻辑或

B. 逻辑与

C. 逻辑非

D. 逻辑相等

23. 在 java 语句中，位运算操作数只能为整型或（）数据

A. 实型

B. 字符型

C. 布尔型

D. 字符串型

00101010（|）00010111 语句的执行结果为

A. 00000000

B. 11111111

C. 00111111

D. 11000000

25. \sim 0010101 语句的执行结果为

A. 1101010

B. 0010101

C. 11111111

D. 00000000

26. 复合语句用 () 括起来的一段代码。

小括号 ()

大括号 { }

中括号 []

单引号 ‘ ’

下列不属于条件语句关键字的是

if

else

swich

while

多分支语句 switch (表达式) { } 中, 表达式不可以返回哪种类型的值

整型

实型

接口型

字符型

下面不属于 java 条件分支语句结构的是

if 结构

if-else 结构

if-else if 结构

if-else else 机构

30. 下列方法 method () 执行, 返回值为

```
int method () {  
  
    int num=10;  
  
    if (num>20)  
  
        return num;  
  
    num=30;  
  
}
```

10

20

30

编译出错

三元条件运算符 `ex1?ex2: ex3`，相当于下面（）语句。

`if (ex1) ex2; else ex3;`

`if (ex2) ex1; else ex3;`

`if (ex1) ex3; else ex2;`

`if (ex3) ex2; else ex1;`

java 用（）来实现 `goto` 语句所特有的一些功能。

`break`

`default`

`continue`

`return`

一个循环一般应包括哪几部分内容

初始化部分

循环体部分

迭代部分和终止部分

以上都是

34. 关于 `while` 和 `do-while` 循环，下列说法正确的是

两种循环除了格式不通外，功能完全相同

与 `do-while` 语句不通的是，`while` 语句的循环至少执行一次

`do-while` 语句首先计算终止条件，当条件满足时，才去执行循环体中的语句

以上都不对。

35. 下列程序输出结果为:

```
public class test
{
    public static void main(String args[])
    {
        int a=0;
        outer: for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                if(j>i)
                {
                    continue
                }
                a++;
            }
        }
    }
}
```

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

0

2

3

4
```

36. 下列数组定义及赋值，错误的是

A. `int intArray[];`

B. `intArray=new int[3];`

`intArray[1]=1;`

`intArray[2]=2;`

`intArray[3]=3;`

C. `int a[]={1,2,3,4,5};`

D. `int[][]=new int[2][];`

`a[0]=new int[3];`

`a[1]=new int[3];`

37. 在 java 中，字符串由 `java.lang.String` 和 () 定义

E. `java.lang.StringChar`

F. `java.lang.StringBuffer`

G. `java.io.StringChar`

H. `java.io.StringBuffer`

二、填空题

1. 在 java 语言中, `boolean` 型常量只有 `true` 和 _____ 两个值。

2. 标识符是以____、下划线、美元符号作为首字母的字符串序列。

3. 下面的语句是声明一个变量并赋值:

```
boolean b1=5!=8;
```

`b1` 的值是_____。

4. 在 java 程序中, 用关键字_____修饰的常量对象创建后就不能再修改了。

5. 数据类型包括简单数据类型和复合数据类型。复合数据类型又包括类、数组、_____。

6. 八进制整数 012 表示十进制的_____。

7. 浮点型数据属于实型数据, 分为 `float` 和_____两种类型。

8. 类变量在类中声明, 而不是在类的某个方法中声明, 它的作用域是_____。

9. java 语言中的各种数据类型之间提供自动转换, 如第 1 操作数是 `byte` 类型, 第 2 操作数是 `float` 类型, 其结果是_____类型。

10. 在 java 语言中, 如果数值后没有字母, 计算机默认值为_____类型。

11. 下面程序分别用 while、do-while 和 for 语句实现 1~10 累计求和。请在横线处填入适当内容完成程序。

```
Public class Sum

{

    public static void main(String args[])

    {

        System.out.println(“\n*****while 循环*****”);

        int n=10, sum=0;

        while(_____)

        {

            sum+=n;

            n--;

        }

        System.out.println(“sum is” +sum);

        System.out.println(“\n*****do_while 循环*****”);

        n=0;

        sum=0;

        do

        {
```

```

        sum+=n;

        n++;

    }

while(______);

System.out.println(“sum is” +sum);

System.out.println(“\n*****for 循环*****”);

sum=0;

for(int I=1;____i++)

    {

        sum+=I

    }

System.out.println(“sum is” +sum);

}

}

```

定义 double 型数组

```
double d1[]={1. 2, 2. 3, 3. 4, 4. 5, 5. 6}
```

则 d1[2]=_____

数组是有序数据的集合，数组中的每个元素具有相同的_____

14. 下列程序输出结果为_____

```
public class test

{

    public static void main(String args[])

    {

        String s=" I am a string!" ;

        Int n=s.length();

        Char c=s.charAt(7);

        System.out.println(n);

        System.out.println(c);

    }

}
```

第三章 面向对象程序设计

一、选择题

2. 下列不属于面向对象编程的三个特征的是

A. 封装

- B. 指针操作
 - C. 多态性
 - D. 继承
3. 类所实现的接口以及修饰不可以是
- A. public
 - B. abstract
 - C. final
 - D. void
4. 下列类的定义, 错误的是
- A.

```
public class test extends Object {  
  
    .....  
  
}
```
 - B.

```
final class operators {  
  
    .....  
  
}
```
 - C.

```
class Point {  
  
    .....  
  
}
```
 - D.

```
void class Point {
```

```
.....  
}
```

5. 关键字 `super` 的作用是
- A. 用来访问父类被隐藏的成员变量
 - B. 用来调用父类中被重载的方法
 - C. 用来调用父类的构造函数
 - D. 以上都是
6. 下面程序定义了一个类，关于该类说法正确的是

```
abstract class abstractClass{  
  
.....  
}
```

- A. 该类能调用 `new abstractClass()`，方法实例化为一个对象
 - B. 该类不能被继承
 - C. 该类的方法都不能被重载
 - D. 以上说法都不对
7. 关于对象的删除，下列说法正确的是
- A. 必须由程序员完成对象的清除
 - B. java 把没有引用的对象作为垃圾收集起来并释放
 - C. 只有当程序中调用 `System.gc()` 方法时才能进行垃圾收集

- D. java 中的对象都很小，一般不进行删除操作。
8. 下列说法正确的是
- A. java 中包的主要作用是实现跨平台功能
 - B. package 语句只能放在 import 语句后面
 - C. 包 (package) 由一组类 (class) 和界面 (interface) 组成
 - D. 可以用 #include 关键词来标明来自其它包中的类;
9. 关于构造方法，下列说法错误的是
- A. 构造方法不可以进行方法重写
 - B. 构造方法用来初始化该类的一个新的对象
 - C. 构造方法具有和类名相同的名称
 - D. 构造方法不返回任何数据类型
10. 请将程序补充完整
1. _____ class C
 2. {
 3. abstract void callme()
 4. void metoo
 5. {
 6. System.out.println(“类C的metoo()方法”);
 7. }

```

8.     }

9.     class D_____C

10.    {

11.        void callme()

12.    {

13.        System.out.println(“重载C类的callme()方法”);

14.    }

15. }

16. public class Abstract

17. {

18.     public static void main(String args[])

19. }

20. C c=_____D();

21. c.callme();

22. c.metoo();

23. }

24. }

```

二、填空题

1. 把对象实例化可以生成多个对象，使用_____运算符为对象分配内存空间。

2. java 程序引入接口的概念, 是为了弥补只允许类的_____的缺憾。
3. java 语言以_____为程序的基本单位, 它是具有某些共同特性实体的集合, 是一种抽象的概念。
4. 抽象方法只能存在于抽象类中。抽象方法用关键字_____来修饰。
5. java 语言中_____是所有类的根。
6. 在 java 中有一种叫做_____特殊方法, 在程序中用它来对类成员进行初始化。
7. new 是_____对象的操作符。
8. 继承主要强调子类在父类的基础上取”长”补”短”, 而_____主要强调的是类与类之间的传输
9. 我们在 java 程序中, 把关键字_____加到方法名称的前面, 来实现子类调用父类的方法
10. 定义一个类包括定义类头和定义_____两个部分
11. 在 java 程序里, 同一类中重载的多个方法具有相同的方法名和_____的参数列表. 重载的方法可以有不同的返回值类型
12. java 是面向对象语言, 对象是客观事物的_____, 对象与之是一一对应的, 它是很具体的概念。
13. java 语言通过接口支持_____继承, 使类继承具有更灵活的扩展性
14. 一个对象的生命周期包含三个阶段: 生成, _____ 和清除
15. java 语言中, 调用方法时, 参数传递是_____调用, 而不是地址调用
16. 接口是一种只含有抽象方法或_____的一种特殊抽象类
17. 创建一个类就是创建一个新的数据类型, 而类在 java 中属于_____数据类型。
18. abstract 方法_____ (不能或能) 与 final 并列修饰同一个类

19. 如果一个类中定义了几个名为 method 的方法，这些方法的参数都是整数，则这些方法的_____必须是不同的，这种现象称为方法的重载。

第四章 Java 小应用程序

选择题

关于 Applet 运行过程，下列说法错误的是（）

浏览器家在指定 URL 中的 HTML 文件

浏览器加密 HTML 文件

浏览器加载 HTML 文件中指定的 Applet 类

浏览器中的 java 运行环境运行该 Applet

Applet 的生命周期是指（）

Applet 下载到浏览器，到用户连接到下一个页面的过程（）

用户一次提交或刷新页面的全过程

Applet 下载到浏览器，到用户退出浏览器的过程

打开浏览器到关闭计算机的全过程

Applet 是一个面板容器，它默认使用（）布局管理器

Border

Flow

Grid

Card

与 Applet 生命周期有关的主要方法是 ()

init()

start()

stop()

以上都是

关于 Applet 和 Application, 下列说法错误的是 ()

Applet 自身不能运行

Applet 可以嵌在 Application 中运行

Application 以 main() 方法为入口

Applet 可嵌在浏览器中运行

在 Applet 中画图、画图像、显示字符串用到的方法是 ()

paint()

init()

stop()

draw()

Graphics 类中提供的绘图方法分为两类: 一类是绘制图形, 另一类是绘制 ()

屏幕

文本

颜色

图像

下列不属于 Applet 编写步骤的是 ()

引入需要的包和类

定义一个 Applet 类的子类

实现 Applet 类的某些方法

加密 Applet 程序

要在一个单一的类文件中创建既可以用作 Java Applet,又可以用作 Java 应用程序的 Java 软件代码。

下面说法错误的是 ()

作为 Application 要定义 main() 方法, 并且把 main() 方法所在的类定义为一个 public 类

为使该程序成为一个 Applet, main() 方法所在的这个 public 类必须继承 Applet 类或者 Lapplet 类

在该类中可以像普通 Applet 类一样重写 Applet 类的 init(), start(), paint() 等方法

转换后的程序只能在浏览器中加载执行, 而不能在 Appletviewer 中执行。

关于 Applet 和 Application, 下列说法错误的是 ()

是 java 的两种应用程序形式

Applet 可以用 Appletviewer 或者浏览器加载执行

Applet 不能利用 java 解释器从命令行启动运行

Applet 和 Application 程序入口不一样

填空题

java 程序可以分为 Application 和 Applet 两大类，能在 WWW 浏览器上运行的是_____。

java Applet 应用程序的编写和执行共分为 4 步进行：编写源代码、编写 HTML 文件调用该小程序、_____、解释执行。

Applet 生命周期中的关键方法包括：_____start()、stop()、destroy()。

每个 Applet 必须定义为_____的子类。

一个 Applet 标记中，必须出现的属性项有_____WIDTH、HEIGHT。

_____是 java 的一类特殊应用程序，它嵌入 HTML 中，随主页发布到互联网上，它在网络上的传输不受计算机 CPU 和环境控制。

java 有 2 类应用程序，它嵌入 java application 和_____

java 体系结构中，最上面一层是_____和 Applet 小程序。

传递给实现了 java.awt.event.MouseMotionListener 接口的类中 mouseDragged() 方法的事件对象是_____。

当用户在 TextField 中输入一行文字后，按回车，实现_____接口可以实现对时间的响应。

编写同时具有 Applet 与 Application 的特征的程序。具体的实现方法：作为 Application 要定义 main() 方法，并且把 main() 方法所在的类定义一个_____类。为使该程序成为一个 Applet, main() 方法所在的这个类必须继承 Applet 类或者 JApplet 类

应用程序一定要 main() 方法，小程序一定要继承自_____类。

Applet 生命周期是指从 Applet 下载到_____，到用户退出浏览器，中止 Applet 运行的结果。

paint() 方法带有一个参数，它是 java.awt. _____类的一个实例。这个参数总是建立该 Applet 的 panel 的图形上下文。能用这个上下文在 Applet 中绘图或写入文本。

Applet 程序既可以用_____或浏览器加载执行，也可以用 java 解释器从命令行启动执行。

下面的 Applet 获得了相对于 `getDocumentBase` 方法返回的目录路径为 `graph/java.gif` 的图像文件，
并将它显示在 `appletviewer` 中。

请在横线处填入适当的内容完成该程序。

```
import java.awt.*;

import java._____;

public class HwImage_____Applet

{

    Image dusk;

    public void init()

    {

        dusk=getImage(getDocumentBase(), " graph/java.gif" );

    }

    public void paint(Graphics g)

    {

        g._____ (duke, 25, 25, this);

    }

}
```

第五章 异常处理

一、选择题

java 中用来抛出异常的关键字是

try

catch

throw

finally

关于异常，下列说法正确的是

异常是一种对象

一旦程序运行，异常将被创建

为了保证程序运行速度，要尽量避免异常控制

以上说法都不对

类是所有异常类的父类。

Throwable

Error

Exception

AWTError

java 语言中，下列哪一子句是异常处理的出口

try {...} 子句

catch {...} 子句

finally{...}子句

以上说法都不对

下列程序的执行，说法错误的是

```
class MultiCatch
{
    public static void main(String args[])
    {
        try
        {
            int a=args.length;

            int b=42/a;

            int c[]={1};

            c[42]=99;

            System.out.println(“b=” +b);
        }

        catch(ArithmeticException e)
        {

            System.out.println(“除 0 异常: ” +e);
        }
    }
}
```

```
        catch(ArrayIndexOutOfBoundsException e)

            {

                System.out.println(“数组超越边界异常: ”+e);

            }

        }

    }
```

程序将输出第 15 行的异常信息

程序第 10 行出错

程序将输出 “b=42”

程序将输出第 19 行的异常信息

下列程序的执行，说法正确的是

```
class ExMulti

{

    static void procedure()

    {

        try

        {

            int c[]={1};

            c[42]=99;

        }

    }

}
```

```
    }

    catch(ArrayIndexOutOfBoundsException e)

    {

        System.out.println(“数组超越界限异常: ”+e);

    }

}

public static void main(String args[])

{

    try

    {

        procedure();

        int a=args.length;

        int b=42/a;

        System.out.println(“b= ”+b);

    }

    catch(ArithmeticException e)

    {

        System.out.println(“除0异常: ”+e);

    }

}
```

```
    }  
}
```

程序只输出第 12 行的异常信息

程序只输出第 26 行的异常信息

程序将不输出异常信息

程序将输出第 12 行和第 26 行的异常信息

下面程序抛出了一个“异常”并捕捉它。请在横线处填入适当内容完成程序。

```
class ThrowsDemo  
{  
  
    static void procedure() throws IllegalAccessException  
    {  
  
        System.out.println(“inside procedure”);  
  
        throw_____IllegalAccessException(“demo”);  
  
    }  
  
    public static void main(String args[])  
    {  
  
        try  
        {  
  
            procedure();  
  
        }  
  
    }  
}
```

```

    }

    _____

    {

        System.out.println(“捕获: ” +e);

    }

}

```

二、填空题

catch 子句都带一个参数，该参数是某个异常的类及其变量名，catch 用该参数去与_____对象的类进行匹配。

java 虚拟机能自动处理_____异常。

变量属性是描述变量的作用域，按作用域分类，变量有局部变量、类变量、方法参数和_____

同一段程序可能产生不止一种异常。可以放置多个_____子句，其中每一种异常类型都将被检查，第一个与之匹配的就会被执行。

捕获异常要求在程序的方法中预先声明，在调用方法时用 try-catch-_____语句捕获并处理。

java 语言认为那些可预料和不可预料的出错称为_____

按异常处理不同可以分为运行异常、捕获异常、声明异常和_____几种。

抛出异常的代码可以是_____或者是 JDK 中的某个类，还可以是 JVN.

抛出异常、生成异常对象都可以通过_____语句实现。

捕获异常的统一出口通过_____语句实现。

java 语言的类库中提供了一个_____类，所有的异常都必须是它的实例或它子类的实例。

Throwable 类有两个子类：_____类和 Exception 类。

对程序语言而言，一般有编译错误和_____错误两类。

下面程序定义了一个字符串数组，并打印输出，捕获数组超越界限异常。请在横线处填入适当的内容完成程序。

```
public class HelloWorld

{

    int i=0;

    String greetings[]=

        {

            "Hello world!",

            "No, I mean it!",

            "HELLO WORLD!!"

        };

    while(i<4)

    {

        _____

    }

}
```

```

System.out.println(greeting[i]);

}

_____ (ArrayIndexOutOfBoundsException e)

{

System.out.println(“Re-setting Index Value”);

i=-1;

finally

{

    System.out.println(“This is always printed”);

}

i++;

}

}

}

```

第六章 图形与用户界面技术

选择题

Window 是显示屏上独立的本机窗口，它独立于其它容器，Window 的两种形式是 ()

Frame 和 Dialog

Panel 和 Frame

Container 和 Component

LayoutManager 和 Container

框架 (Frame) 的缺省布局管理器就是 ()

流程布局 (Flow Layout)

卡布局 (Card Layout)

边框布局 (Border Layout)

网格布局 (Grid Layout)

java.awt 包提供了基本的 java 程序的 GUI 设计工具, 包含控件、容器和 ()

布局管理器

数据传送器

图形和图像工具

用户界面构件

所有 Swing 构件都实现了 () 接口

ActionListener

Serializable

Accessible

MouseListener

事件处理机制能够让图形界面响应用户的操作, 主要包括 ()

事件

事件处理

事件源

以上都是

Swing 采用的设计规范是 ()

视图----模式----控制

模式----视图---控制

控制-----模式----视图

控制----视图-----模式

抽象窗口工具包()是 java 提供的建立图形用户界面 GUI 的开发包.

AWT

Swing

Java. io

Java. lang

关于使用 Swing 的基本规则, 下列说法正确的是()

Swing 构件可直接添加到顶级容器中

要尽量使用非 Swing 的重要级构件

Swing 的 Jbutton 不能直接放到 Frame 上

以上说法都对

下列不属于 java. event 包中定义的事件适配器的是()

构件适配器

焦点适配器

键盘适配器

标签适配器

() 布局管理器使容器中各个构件呈网格布局，平均占据容器空间。

FlowLayout

BorderLayout

GridLayout

CardLayout

填空题

在需要自定义 Swing 构件的时候，首先要确定使用那种构件类作为所定制构件的____，一般继承 JPanel 类或更具体的 Swing 类。

Swing 的事件处理机制包括____、事件和事件处理者。

Java 事件处理包括建立事件源、____和将事件源注册到监听器。

Java 的图形界面技术经历了两个发展阶段，分别通过提供 AWT 开发包和____开发包来实现。

抽象窗口工具包____提供用于所有 Java applets 及应用程序中的基本 GUI 组件。

Window 有两种形式：Frame(框架)和_____。

容器里的组件的位置和大小是由_____决定的。

可以使用 `setLocation()`, `setSize()` 或_____中的任何一中方法设定组件的大小或位置。

容器 `Java.awt.Container` 是_____类的子类。

框架的缺省布局管理器是_____。

_____包括五个明显的区域：东、南、西、北、中。

_____布局管理器是容器中各个构件呈网格布局，平均占据容器空间

_____组件提供了一个简单的“从列表中选取一个”类型的输入。

在组件中显示时所使用的字体可以用_____方法来设置。

为了保证平台独立性，Swing 是用_____编写。

Swing 采用了一种 MVC 的设计范式，即_____。

Swing GUI 使用两种类型的类，即 GUI 类和_____支持类。

_____由一个玻璃面板、一个内容面板和一个可选择的菜单条组成。

对 Swing 构件可以设置_____边框。

_____对话框在被关闭前将阻塞包括框架在内的其他所有应用程序的输入

读程序后写出结果

```
import java.awt.*;

public class ExGui3

{

    private Frame f;

    private Panel p;
```

```
private Button bw, bc;

public static void main(String args[])

{

    ExGui3 gui=new ExGui3();

    Gui.go;

}

public void go()

{

    f=new Panel(“Ex1”);

    bw=new Button(“西”);

    bc=new Button(“空白区域”);

    f.add(bw, BorderLayout.CENTER);

    f.add(bc, BorderLayout.CENTER);

    p=new Panel();

    bfile=new Button(“文件”);

    bhelp=new Button(“帮助”);

    p.add(bfile);

    p.add(bhelp);

    f.add(p, BorderLayout.NORTH);
```

```
f.pack();
```

```
f.setVisible(false);
```

第七章 多线程

选择题

1. 线程调用了 sleep () 方法后，该线程将进入 () 状态。

可运行状态

运行状态

阻塞状态

终止状态

2. 关于 java 线程，下面说法错误的是 ()

线程是以 CPU 为主体的行为

java 利用线程使整个系统成为异步

创建线程的方法有两种：实现 Runnable 接口和继承 Thread 类

新线程一旦被创建，它将自动开始运行

3. 在 java 中的线程模型包含 ()

A. 一个虚拟处理器

B. CPU 执行的代码

C. 代码操作的数据

D. 以上都是

4. 在 java 语言中，临界区可以是一个语句块，或者是一个方法，并用（）关键字标识。

A. synchronized

B. include

C. import

D. Thread

5. 线程控制方法中，yield() 的作用是（）

A. 返回当前线程的引用

B. 使其比低的优先级线程执行

C. 强行终止线程

D. 只让给同优先级线程运行

6. 线程同步中，对象的锁在（）情况下持有线程返回

A. 当 synchronized() 语句块执行完后

B. 当在 synchronized() 语句块执行中出现例外 (exception) 时

C. 当持有锁的线程调用该对象的 wait() 方法时

D. 以上都是

7. 在以下（）情况下，线程就进入可运行状态

A. 线程调用了 sleep() 方法时

B. 线程调用了 join() 方法时

C. 线程调用了 yield() 方法时

D. 以上都是

8. java 用 () 机制实现了进程之间的异步执行

A. 监视器

B. 虚拟机

C. 多个 CPU

D. 异步调用

9. 下列程序实现简单的线程调度，请回答以下问题：

```
1 Thread myThread=new MyThreadClass();

myThread.start();

try

{

    myThread.sleep(10000);

}

catch(InterruptedException e)

{

}

myThread.stop();
```

程序执行完第一行后，线程进入_____状态；

程序执行完第二行后，线程进入_____状态；

程序开始执行第五行时，线程进入_____状态；

程序执行完第五行后，线程进入_____状态；

程序执行完第十行后，线程进入_____状态；

新建状态

可运行状态

阻塞状态

终止状态

10. Thread 类的方法中，toString()方法的作用是（）

- A. 只返回线程的名称
- B. 返回当前线程所属的线程组的名称
- C. 返回当前线程对象
- D. 返回线程的名称

填空题

1. _____是 java 程序的并发机制，它能同步共享数据、处理不同的事件。
2. 线程是程序中的一个执行流，一个执行流是由 CPU 运行程序的代码、_____所形成的，因此，线程被认为是以 CPU 为主体的行为。
3. 线程的终止一般可以通过两种方法实现：自然撤销或者是_____。
4. 线程模型在 java 中是由_____类进行定义和描述的。

5. 线程的创建有两种方法：实现_____接口和继承 Thread 类。
6. 多线程程序设计的含义是可以将程序任务分成几个_____的子任务。
7. 按照线程的模型，一个具体的线程也是由虚拟的 CPU、代码与数据组成，其中代码与数据构成了_____，线程的行为由它决定。
8. java 中，新建的线程调用 start() 方法、如 myThread.start(), 将使线程的状态从 New(新建状态)转换为_____。
9. 多线程是 java 程序的_____机制，它能同步共享数据，处理不同事件。
10. 进程是由代码、数据、内核状态和一组寄存器组成，而线程是表示程序运行状态的_____，如程序计数器、栈指针以及堆栈组成。
11. Thread 类提供了一系列基本线程控制方法，如果我们需要让与当前进程具有相同优先级的线程也有运行的机会则可以调用_____方法。

第八章 多媒体编程

一、编写一个有背景声音的 Java 程序。

二、编写一个实际的动画程序。

第九章 输入与输出流

一、选择题

1. 流的传递方式是 ()

- A. 并行的
- B. 串行的
- C. 并行和串行
- D. 以上都不对

2. 下列不是 java 的输入输出流的是 ()

- A. 文本流
- B. 字节流
- C. 字符流
- D. 文件流

3. 凡是从中央处理器流向外部设备的数据流称为 ()

- A. 文件流
- B. 字符流
- C. 输入流
- D. 输出流

4. 获取一个不包含路径的文件名的方法为 ()

- A. `String getName()`
- B. `String getPath()`
- C. `String getAbsolutePath()`

D. `String getParent()`

5. 下列属于文件输入输出类的是 ()

A. `FileInputStream` 和 `FileOutputStream`

B. `BufferInputStream` 和 `BufferOutputStream`

C. `PipedInputStream` 和 `PipedOutputStream`

D. 以上都是

6. 下列不属于 `FileInputStream` 输入流的 `read()` 成员函数的是 ()

A. `int read();`

B. `int read(byte b[]);`

C. `int read(byte b[], int offset, int len);`

D. `int read(int line);`

7. 当处理的数据量很多, 或向文件些很多次小数据, 一般使用 () 流

A. `DataOutput`

B. `FileOutput`

C. `BufferedOutput`

D. `PipedOutput`

8. 当把一个程序、线程或代码段的输出连接到另一个程序、线程或代码段的输入时, 应使用 () 流

A. `DataOutput`

B. `FileOutput`

C. BufferedOutput

D. PipedOutput

9. 当要将一文本文件当作一个数据库访问，读完一个纪录后，跳到另一个纪录，它们在文件的不同地方时，一般使用（）类访问。

A. FileOutputStream

B. RandomAccessFile

C. PipedOutputStream

D. BufferedOutputStream

10. java 中，实现通过网络使用 URL 访问对象的功能的流是（）

A. URL 输入流

B. Sock 输入流

C. PipedInputStream 输入流

D. BufferedInputStream 输入流

二、填空题

1. java 的输入输出流包括_____、字符流、文件流、对象流以及多线程之间通信的管道。

2. java 中的非字符输出流都是_____抽象类的子类。

3. java 对 I/O 访问还提供了同步处理机制，保证某时刻只有一个线程访问一个 I/O 流，这就是_____。

4. java 中的字符输出流都是抽象类_____的子类。

5. DataOutputStream 数据流向文件里写数据的方法为_____。

6. RandomAccessFile 所实现的接口是_____和 DataOutput 接口.
7. 文件类_____是 java.io 中的一个重要的非流类, 里面封装了对文件系统进行操作的功能.
8. 文件操作中经常需要的是随机访问, java 中的_____类提供了随机访问文件的功能.
9. 下列程序实现从控制台输入并读取输出字符串. 请将程序补充完整

```
import java.io.*;

public class CharInput

{

    public static void main(String args[]) throws java.io.IOException

}

String s;

InputStreamReader ir;

BufferedReader in;

ir=new _____(System.io);

in=new _____(ir);

while ((s=in. _____) !=null)

{

    System.out.println( "Read;" +s);

}

}
```

```
}
```

10. 下列程序实现从当前目录的 date.ser 文件中读取文件内容并显示出来. 请将程序补充完整.

```
import java.io.*;
```

```
import java.util.Date;
```

```
public class UnSerializeDate
```

```
{
```

```
    UnSerializeDate()
```

```
    {
```

```
        Date d=null;
```

```
        try
```

```
        {
```

```
            FileInputStream f=new
```

```
                _____(“date.ser”);
```

```
            ObjectInputStream s=new
```

```
                ObjectInputStream(_____);
```

```
            d=(Date) s.readObject();
```

```
            f.close();
```

```
        }
```

```
    catch(Exception e)
```

```

    {

        e.printStackTrace();

    }

    System.out.println(“Unserialized Date object from date.set”);

    System.out.println(“Date:” +d);

}

public static void main(String args[])

{

    new UnSerializeDate();

}

}

```

第十章网络通讯与编程

1. 套接口作用是什么？端口有什么意义？
2. 顾客和服务者通过套接口通讯，描述基于有连接和无连接通讯时的流程图
3. 什么是 URL？一个 URL 由哪此部分组成？

第十一章 java 语言的数据库访问技术

一、选择题

1. Java 中，JDBC 是指
 - A. Java 程序与数据库连接的一种机制

- B. Java 程序与浏览器交互的一种机制
 - C. Java 类库名称
 - D. Java 类编译程序
2. 在利用 JDBC 连接数据库时, 为建立实际的网络连接, 不必传递的参数是
- A. URL
 - B. 数据库用户名
 - C. 密码
3. J2ME 是为嵌入式和移动设备提供的 Java 平台, 它的体系结构由()组成。
- A. Profiles
 - B. Configuration
 - C. OptionalPackages
 - D. 以上都是
4. J2EE 包括的服务功能有
- A. 命名服务 JNDI (LDAP) 和事务服务 JTA
 - B. 安全服务和部署服务
 - C. 消息服务 JMS 和邮件服务 JavaMail
 - D. 以上都是
5. JDBC 的模型对开放数据库连接 (ODBC) 进行了改进, 它包含 ()
- A. 一套发出 SQL 语句的类和方法
 - B. 更新表的类和方法
 - C. 调用存储过程的类和方法
 - D. 以上全部都是
6. JDBC 中要显式地关闭连接的命令是 ()
- A. `Connection.close();`
 - B. `RecordSet.close()`

- C. `Connection. stop()` D. `Connection. release()`
7. TCP / IP 系统中的端口号是一个()位的数字，它的范围是 0 到 65535。
- A. 8 B. 16 C. 32 D. 64
8. 在 Java 编程语言中，TCP / IP socket 连接是用 `java. net` 包中的类实现的。其连接步骤和方法是
- A. 服务器分配一个端口号。如果客户请求一个连接，服务器使用 `accept()` 方法打开 socket 连接
- B. 客户在 host 的 port 端口建立连接
- C. 服务器和客户使用 `InputStream` 和 `OutputStream` 进行通信
- D. 以上全部
9. J2ME 中的 Profile 定义应用系统的()特性。
- A. 生命周期模型 B 用户界面 C. 访问设备 D. 以上全部都是
10. J2EE 平台的优势不包括 ()
- A. “一次编程，任意运行”
- B. 应用独立于供应商、服务器、工具，构件的选择自由
- C. 跨平台的解释器作为其核心技术
- D. 提高软件复用

二、填空题

1. Java 提供的类库支持_____协议，应用程序可通过 UR1 地址，在访问网络上任何地方的对象时，如同访问本地文件一样简单。
2. JDBC API 的含义是 Java 应用程序连接_____的编程接口。

3. Socket 的工作步骤分为：创建 Socket、打开连接到 Socket 的输入/输出流、按某个协议对 Socket 进行读 / 写操作、_____。
4. Java 编程语言前台应用程序使用_____来和 JDBC 驱动管理器进行交互。
5. JDBC 驱动管理器使用_____来装载合适的 JDBC 驱动。
6. Java 应用程序通过 JDBC. API 向 JDBC DriverManager 发出请求，指定要装载的 JDBC 驱动程序代码，指定要连接的数据库的具体类型(品牌和版本号)和实例。JDBC. API 主要是定义在_____中的类和方法。
7. JDBC 的类都被汇集在_____包中，在安装 JavaJDK1. 1 或更高版本时会自动安装。
8. 一个 socket 包括两个流：一个输入流和一个输出流。如果一个进程要通过网络向另一个进程发送数据，只需简单地写入与 socket 相关联的_____。
9. J2ME 是为嵌入式和_____提供的 Java 平台，它的体系结构由 Profiles、Configuration 和 OptionalPackages 组成。
10. 查询数据库的 7 个标准步骤是：载人 JDBC 驱动器、定义连接的网址 URL、建立连接、建立声明对象、执行查询或更新、处理结果、_____。
11. _____是指在一个特定编程模型下，进程间通信链路的端点。

参 考 答 案

第一章 JAVA 语言概论

一 选择题

1. B 2.C 3.B 4.A 5.A 6.B 7.A 8.C 9.A 10.A

二 填空题

1. Oak

2. 面向对象
3. java 虚拟机
4. 动态连结
5. 编译层
6. Windows
7. 垃圾回收
8. java
9. 半解释型
10. 跨平台
11. javadoc.exe (或 java 文档生成器)
12. 由编译器生成的类 (.class 文件)
13. 字节码
14. 解释执行
15. main
16. public 或公共
17. java 基本类
18. javax
19. 字节代码
20. 移植接口

第二章 Java 语言基础

一 选择题

1.D 2.B 3.C 4.C 5.B 6.A 7.B 8.D 9.B 10.D 11.C 12.C 13.D 14.A 15.
A 16.B 17.D 18.D 19.A 20.D 21.A 22.B 23.B 24.C 25.A 26.B 27.
D 28.C 29.D 30.D 31.A 32.A 33.D 34.D 35.C 36.B 37.B

二、填空题

1. false

2. 字母

3. true

4. final

5. 接口

6. 10

7. double

8. 整个类

9. float

10. double

11. $n > 0$ $n \leq 10$ $I \leq 10$

12. 3.4

13. 数据类型

14. 14, s

第三章 面向对象程序设计

一 选择题

1. B 2. D 3. D 4. D 5. D 6. B 7. C 8. B 9. abstract extends new

二、填空题

1. new

2. 单继承

3. 类

4. abstract

5. Object

6. 构造函数

7. 创建

8. 多态性

9. supper

10. 类体

11. 不同的

12. 实体

13. 多重
14. 使用
15. 值
16. 常量
17. 不能
18. 复合
19. 参数个数

第四章 Java 小应用程序

一、选择题

1. B 2. C 3. B 4. D 5. B 6. A 7. B 8. D 9. D 10. C

二、填空题

1. Applet
2. 编译过程
3. `init()`
4. Applet 或者 Japplet
5. CODE
6. Applet
7. `java Applet`

8. java 应用程序
9. MouseEvent
10. ActionListener
11. Public
12. Applet 或者 JApplet
13. 浏览器
14. Graphics
15. Appletviewer
16. applet.Applet(或 applet.*)extends drawImage

第五章 异常处理

一、选择题

1. A
2. A
3. C
4. A
5. D
6. new, catch(IllegalAccessException e)

二、填空题

1. 抛出异常
2. 运行
3. 异常处理参数
4. finally
5. 异常

6. 抛出异常
7. java 应用程序
8. throw
9. finally
10. Throwable
11. Error
12. 运行
13. try catch

第六章 图形与用户界面技术

一、选择题

1.A 2.C 3.A 4.C 5.D 6.B 7.A 8.D 9.D 10.C

二、填空题

1. 父类
2. 事件源
3. 建立事件监听器
4. Swing
5. AWT(Abstract Window Toolkit)
6. Dialog(对话框)

7. 布局管理器
8. setBounds()
9. Component
10. Border Layout
11. Border 布局管理器
12. GridLayout
13. 下拉列表
14. setFont()
15. java
16. “模型—视图—控制”
17. 非 GUI
18. 根面板
19. 一个或多个
20. “模式”

三、读程序后写出结果

```
Frame BorderLayout.WEST true
```

第七章 多线程

一、选择题

1. C 2. D 3. D 4. A 5. D 6. D 7. C 8. A 9. ABCBD 10. D

二、填空题

1. 多线程
2. 操纵程序的数据
3. 被停止
4. java.lang.Thread
5. Runnable
6. 并行
7. 线程体
8. 可运行状态
9. 并发
10. 寄存器
11. yield()

第八章 多媒体编程

1、

```
import java.io.*;
```

```
import java.util.*;
```

```
import java.awt.*;
```

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

```
import java.net.*;
```

```

import java.applet.*;

public class multiAudio extends Applet implements ActionListener

{

    String hUr11,hUr12;

    AudioClip audio1, audio2;

    boolean hState1=false;

    boolean hState2=false;

    public void init()

    {

        reize(300,100);

        hUr11= "/test.wav " ;

        hUr12= "/test.mid " ;

        try

        {

            audio1=getAudioClip(new URL(getDocumentBase(),hUr11));

            audio2=getAudioClip(new URL(getDocumentBase(),hUr12));

        }

        catch(Exception e)

        {

```

```
}

this.setBackground(Color. lightGray);

Panel p1=new Panel();

Button myButton1=new Button(" 播放 1 ");

myButton1.addActionListener(this);

p1.add(myButton1);

Button myButton2=new Button(" 停止 1 ");

myButton2.addActionListener(this);

p1.add(myButton2);

Button myButton3=new Button(" 循环 1 ");

myButton3.addActionListener(this);

p1.add(myButton3);

this.add(p1);

Panel p2=new Panel();

Button mybutton4=new Button(" 播放 2 ");

myButton4.addActionListener(this);

p2.add(myButton4);

Button myButton5=new Button(" 停止 2 ");

myButton5.addActionListener(this);
```

```

p2.add(myButton5);

Button myButton6=new Button("循环2");

myButton6.addActionListener(this);

p2.add(myButton6);

this.add(p2);

}

public void actionPerformed(ActionEvent e)

{

    if(e.getActionCommand()=="播放1")

    {

        if(hState1==false)

        {

            audio1.play();

            hState1=true;

        }

    }

    else

    {

        if(e.getActionCommand()=="播放2")

```

```
{  
  
    if(hState2==false)  
  
        {  
  
            audio2.play();  
  
            hState2=true;  
  
        }  
  
    }  
  
else  
  
{  
  
    if(e.getActionCommand()=="停止1")  
  
        {  
  
            if(hState1==true)  
  
                {  
  
                    audio1.stop();  
  
                    hState1=false;  
  
                }  
  
        }  
  
    }  
  
else  
  
{
```

```
        if(e.getActionCommand()==" 停止 2 ")

        {

            if(hState2==true)

            {

                audio2.stop();

                hState2=false;

            }

        }

    else

    {

        if(e.getActionCommand()==" 循环 1 ")

        {

            audio1.loop();

            else audio2.loop();

        }

    }

}

}
```

运行结果如图所示.



图播放多个声音文件的运行结果

单击这些按钮可以产生不同的声音效果。

使用 URL 对象可以播放网络任何位置的声音文件。

二、

```
import java.awt.*;

import java.applet.*;

import java.awt.event.*;

public class ActiveImages extends Applet implements Runnable, ActionListener

{

    Image iImages[]; //图像数组

    Thread aThread;

    int iFrame; //图像数组下
```

标

```

int sleeptime; //参数, 线程 sleep

String graphfile; //图片文件名

int graphcount; //图片张数

AudioClip au; //定义一个声音

Button b1, b2;

public void init()

{

    int i, j;

    iFrame=0;

    aThread=null;

    sleeptime=Integer.parseInt(getParameter("sleeptime"));

    graphfile = getParameter("graphfile");

    graphcount = Integer.parseInt(getParameter("graphcount"));

    iImages = new Image[graphcount];

    String fname = graphfile;

    j = fname.indexOf(".");

    for (i=0;i<graphcount;i++)

```

时间

对象

```

{

    fname = fname.substring(0, j-1)+i+fname.substring(j);

    iImages[i] = getImage(getDocumentBase(), "Images/"+fname);

}

au=getAudioClip(getDocumentBase(), "Wav/Sound.wav");

au.play(); //播放一次声

```

音文件

```

Panel p1 = new Panel();

b1 = new Button("Start");

b2 = new Button("Stop");

p1.add(b1);

p1.add(b2);

b1.addActionListener(this);

b2.addActionListener(this);

setLayout(new BorderLayout());

add(p1, "South");

}

public void start()

{

```

```

        if (aThread == null)

        {

                aThread = new Thread(this);

                aThread.start();                //线程启动

                bl.setEnabled(false);

        }

}

public void stop()

{

        if (aThread != null)

        {

                aThread.interrupt();                //线程中断

                aThread = null;

                au.stop();                //停止播放声

                音文件

        }

}

public void run()

{

```

```

while (true)

{

    iFrame++;

    iFrame %= (iImages.length); //下一幅图像的下标

    repaint();

    try

    {

        Thread.sleep(sleeptime);

    }

    catch (InterruptedException e)

    {

        //

中断时抛出

        break; //退出循

环

    }

}

public void update(Graphics g)

{

```

```

        g.drawImage(iImages[iFrame], 0, 0, this);
    }

    public void actionPerformed(ActionEvent e)
    {
        if ((e.getSource()==b1) && (aThread == null) )
        {
            //
单击 Start 按钮时触发

            aThread = new Thread(this);

            aThread.start();           //线程启动

            b1.setEnabled(false);

            b2.setEnabled(true);

            au.loop();                 //循环播放声
音文件

        }

        if ((e.getSource()==b2) && (aThread != null) )
        {
            //
单击 Stop 按钮时触发

            aThread.interrupt();      //线程中断

            aThread = null;

            b1.setEnabled(true);

```

```
        b2.setEnabled(false);

        au.stop(); //停止播放声

        音文件
    }
}
}
```

第九章 输入与输出流

一、选择题

1. B 2. A 3. D 4. A 5. D 6. D 7. C 8. D 9. B 10. A

二、填空题

1. 字节流
2. OutputStream
3. 过滤流
4. Writer
5. write()
6. DataInput 接口
7. File
8. RandomAccessFile

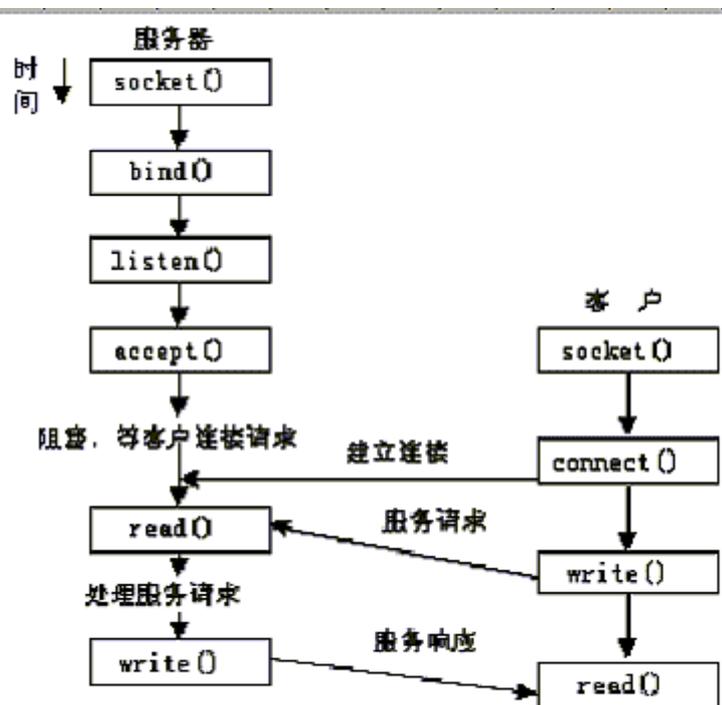
9. InputStreamReader BufferedReader readLine()

10. FileInputStream

第十章 网络通讯与编程

1、 答：套接口（Socket），是传送层提供的网络进程通信接口。通讯的主动方是顾客，它利用 Socket，向服务器（接收方）发送请求，服务器接收顾客请求，返回服务结果。进程之间想通信首先要建立各自的套接口。端口同一台网络计算机的一个特定进程关联，与进程建立的套接口绑定在一起。客户程序必须事先知道自己要求的那个服务进程的 IP 地址和端口号。程序员在创建自己的应用服务程序时一般自己指定一个端口号，也可以由系统分配一个端口号。客户通过这个端口号连接该服务进程。客户端应用进程象服务进程一样，也有自己的端口号，通过该端口客户端应用进程与服务进程通信。

2、 答：TCP（Transport Control Protocol）传输控制协议是面向连接的、可靠的点对点的连接的协议。



图：基于连接的服务者、客户程序流程图。

服务器利用 UDP 协议实现客户和服务器之间通信, 首先建立数据报套接口对象, 自己指定一个端口号, 也可以由系统自动确定端口号。利用无参的构造方法 DatagramSocket () 系统自动确定端口号, 这时必须利用 getLocalPort () 方法获取该数据报端口号。这个端口号及服务器的 IP 地址对客户方发送报文来说是必须知道的。否则, 客户方无法发送报文。

客户方利用 UDP 协议实现客户和服务器之间通信, 首先也要建立数据报套接口对象, 一般使用无参的构造方法 DatagramSocket (), 这时, 系统自动确定一个端口号, 然后利用发送报文缓冲区、服务器的 IP 地址及服务器的端口号建立发送数据包 DatagramPacket 对象。利用建立的数据报套接口发送该 DatagramPacket 对象。

3、 答: URL (Uniform Resource Locator) 是 WWW 资源统一资源定位器的简写。它规范了 WWW 资源网络定位地址的表示方法。一个 URL 的组成部分有: WWW 资源的特征及读其内容的方法。URL 的基本表示格式是: protocol://hostname:/resourcename#anchor; 其中: protocol (使用的协议): 它可以是 http、ftp、news、telnet 等; hostname (主机名): 指定域名服务器 (DNS) 能访问到的 WWW 服务的计算机; port (端口号): 是可选的, 表示所连的端口号, 如缺省, 将连接到协议缺省的端口; Resourcename (资源名): 是主机上能访问到的目录或文件; anchor (标记): 也是可选的, 它指定文件中的有特定标记的位置。

第十一章 java 语言的数据库访问技术

一、选择题

1. A 2. D 3. D 4. D 5. D 6. A 7. B 8. D 9. D 10. C

二、填空题

1. TCP / IP
2. 数据库
3. 关闭 Socket
4. JDBC API
5. JDBC DriverAPI

6. java.Sql. *包

7. java. sql

8. 输出流

9. 移动设备

10. 关闭连接

11. socket