

清华大学出版社

Java 面向对象程序设计第3版-微课版-习题解答

第3版-微课版-习题解答



高等学校Java课程系列教材

Java 面向对象程序设计 (第3版) 微课视频版

◎ 耿祥义 张跃平 编著

50小时
50 HOURS
视频讲解
JDK11版本

《教学大纲》 《教学课件》 《电子教案》
《程序源码》 《习题解答》 《上机指导》

定价：79.80元 ISBN：978-7-302-54052-6

耿祥义 张跃平
2019/11/16

清华大学出版社 2019

《Java 面向对象程序设计》

第 3 版-微课版-习题解答

耿祥义 java 辅助教学微信公众号 java-violin



2019-10-06

目录

习题 1.....	1
习题 2.....	1
习题 3.....	2
习题 4.....	5
习题 5.....	6
习题 6.....	7
习题 7.....	8
习题 8.....	9
习题 9.....	11
习题 10	13
习题 11	17
习题 12	19
习题 13	22
习题 14	23
习题 15	26
习题 16	28



习题 1

1、判断题

(1) (√) ,(2) (×) ,(3) (×) ,(4) (√) ,(5) (×) ,(6) (×) ,(7) (×) ,(8) (√) ,(9) (×) ,(10) (×)

2、单选题

(1). B (2). D (3). A (4). C (5). C

三、挑错题

(1). D (2). A (3). B

4、 A) Speak.java

B) 2 个字节码文件。Speak.class ,Xiti4.class

C) Xiti4

D) 错误: 在类 Speak 中找不到 main 方法

5、

Speak.java

```
public class Speak {  
    void speakHello() {  
        System.out.println("I'm glad to meet you");  
    }  
}
```

Xiti4.java

```
public class Xiti4 {  
    public static void main(String args[]) {  
        Speak sp=new Speak();  
        sp.speakHello();  
    }  
}
```

习题 2

1、判断题

(1) (×) ,(2) (√) ,(3) (√) ,(4) (×) ,(5) (√) ,(6) (√) ,(7) (√) ,(8) (√) ,(9) (×) ,(10) (×)

2、单选题

(1). B (2). A (3). C (4). A (5). D (6). B (7). B (8). D

3、挑错题

(1). D (2). A (3). B (4). C

4.读程序题

(1).127 (2). 200 (3).1 (4).600

5.

```
public class Xiti5{
```

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

```
        char ch1='你',ch2='我',ch3='他';
```

```
        System.out.println("\""+ch1+"\"的位置:"+(int)ch1);
```

```
        System.out.println("\""+ch2+"\"的位置:"+(int)ch2);
```

```
        System.out.println("\""+ch3+"\"的位置:"+ (int)ch3);
    }
}
```

习题 3

1、判断题

(1) (×) , (2) (√) , (3) (√) , (4) (×) , (5) (√) , (6) (√) , (7) (√) , (8) (√) , (9) (×) , (10) (×)

2、单选题

(1). A (2). C (3). B (4). C (5). C

3、挑错题

(1). D (2). B (3). D

4.读程序题

(1). truehello10 (2). truehello100 (3). 321 (4). 5

5. 编程题

(1)

```
public class E {
    public static void main (String args[ ]){
        int startPosition=0,endPosition=0;
        char cStart='a',cEnd='я';
        startPosition=(int)cStart;
        endPosition=(int)cEnd ;
        System.out.println("俄文字母表: ");
        for(int i=startPosition;i<=endPosition;i++){
            char c='\0';
            c=(char)i;    //i 做 char 型转换运算，并将结果赋值给 c
            System.out.print(" "+c);
            if((i-startPosition+1)%10==0)
                System.out.println("");
        }
    }
}
```

(2)

```
public class E {
    public static void main(String args[]){
        double sum=0,a=1;
        int i=1;
        while(i<=20){
            sum=sum+a;
            i++;
            a=a*i;
        }
        System.out.println("sum="+sum);
    }
}
```

(3)

```
public class E {  
    public static void main(String args[]) {  
        int i,j;  
        for(j=2;j<=100;j++) {  
            for(i=2;i<=j/2;i++) {  
                if(j%i==0)  
                    break;  
            }  
            if(i>j/2) {  
                System.out.print(" "+j);  
            }  
        }  
    }  
}
```

(4)

```
public class E {  
    public static void main(String args[]) {  
        double sum=0,a=1,i=1;  
        while(i<=20){  
            sum=sum+a;  
            i++;  
            a=(1.0/i)*a;  
        }  
        System.out.println("使用 while 循环计算的 sum="+sum);  
        for(sum=0,i=1,a=1;i<=20;i++) {  
            a=a*(1.0/i);  
            sum=sum+a;  
        }  
        System.out.println("使用 for 循环计算的 sum="+sum);  
    }  
}
```

(5)

```
public class E {  
    public static void main(String args[]){  
        int sum=0,i,j;  
        for(i=1;i<=1000;i++){  
            for(j=1,sum=0;j<i;j++){  
                if(i%j==0)  
                    sum=sum+j;  
            }  
            if(sum==i)  
                System.out.println("完数:"+i);  
        }  
    }  
}
```

```
}  
}
```

(6)

```
import java.util.Scanner;  
public class E {  
    public static void main (String args[] ){  
        System.out.println("请输入两个非零正整数，每输入一个数回车确认");  
        Scanner reader=new Scanner(System.in);  
        int m=0,n=0,temp=0,gy=0,gb=0,a,b;  
        a=m = reader.nextInt();  
        b=n = reader.nextInt();  
        if(m<n){  
            temp=m;  
            m=n;  
            n=temp;  
        }  
        int r=m%n;  
        while(r!=0) {  
            n=m;  
            m=r;  
            r=m%n;  
        }  
        gy=n;  
        gb=a*b/gy;  
        System.out.println("最大公约数 ： "+gy);  
        System.out.println("最小公倍数 ： "+gb);  
    }  
}
```

(7)

```
public class E {  
    public static void main(String args[]){  
        int n=1;  
        long sum=0,t=1;  
        t=n*t;  
        while(true){  
            sum=sum+t;  
            if(sum>9999)  
                break;  
            n++;  
            t=n*t;  
        }  
        System.out.println("满足条件的最大整数:"+n-1);  
    }  
}
```

习题 4

1、判断题

(1) (√) ,(2) (√) ,(3) (√) ,(4) (×) ,(5) (√) ,(6) (√) ,(7) (√) ,(8) (√) ,(9) (×) ,(10)(×)

2、单选题

(1). D (2). A (3). C (4).A (5).C

3、挑错题

(1).B (2).B (3).C

4.读程序题

(1). 121:111:111 (2). 100:88:188 (3). 10:50 (4). 20:11

5. 编程题

(1) 属于操作题，无解答

(2)

CPU.java

```
public class CPU {  
    int speed;  
    int getSpeed() {  
        return speed;  
    }  
    public void setSpeed(int speed) {  
        this.speed = speed;  
    }  
}
```

HardDisk.java

```
public class HardDisk {  
    int amount;  
    int getAmount() {  
        return amount;  
    }  
    public void setAmount(int amount) {  
        this.amount = amount;  
    }  
}
```

PC.java

```
public class PC {  
    CPU cpu;  
    HardDisk HD;  
    void setCPU(CPU cpu) {  
        this.cpu = cpu;  
    }  
    void setHardDisk(HardDisk HD) {  
        this.HD = HD;  
    }  
    void show(){  
        System.out.println("CPU 速度:"+cpu.getSpeed());  
        System.out.println("硬盘容量:"+HD.getAmount());  
    }  
}
```



```
}
```

Test.java

```
public class Test {  
    public static void main(String args[]) {  
        CPU cpu = new CPU();  
        HardDisk HD=new HardDisk();  
        cpu.setSpeed(2200);  
        HD.setAmount(200);  
        PC pc =new PC();  
        pc.setCPU(cpu);  
        pc.setHardDisk(HD);  
        pc.show();  
    }  
}
```

习题 5

1、判断题

(1) (×) ,(2) (√) ,(3) (×) ,(4) (√) ,(5) (×) ,(6) (√) ,(7) (√) ,(8) (√) ,(9) (×) ,(10)(×)

2、单选题

(1). D (2). A (3). B (4).C (5).D (6).A

3、挑错题

(1).B (2).D (3).C (4).D

4.读程序题

(1). 22 (2). 6:6:100 (3). 18:18 (4). 18:12:7:6

5. 编程题

(1)

```
class A {  
    public final void f(){  
        char cStart='a',cEnd='z';  
        for(char c=cStart;c<=cEnd;c++) {  
            System.out.print(" "+c);  
        }  
    }  
}  
  
class B extends A {  
    public void g() {  
        char cStart='α',cEnd='ω';  
        for(char c=cStart;c<=cEnd;c++){  
            System.out.print(" "+c);  
        }  
    }  
}  
  
public class E {  
    public static void main (String args[ ]) {  
        B b=new B();
```

```
        b.f();
        b.g();
    }
}

(2)
class A {
    public int f(int a,int b){
        if(b<a){
            int temp=0;
            temp=a;
            a=b;
            b=temp;
        }
        int r=b%a;
        while(r!=0) {
            b=a;
            a=r;
            r=b%a;
        }
        return a;
    }
}

class B extends A {
    public int f(int a,int b) {
        int division=super.f(a,b);
        return (a*b)/division;
    }
}

public class E {
    public static void main (String args[ ]) {
        A a=new A();
        System.out.println("最大公约数 : "+a.f(36,24));
        a = new B();
        System.out.println("最小公倍数 : "+a.f(36,24));
    }
}
```

习题 6

1、判断题

(1) (√) ,(2) (√) ,(3) (√) ,(4) (√) ,(5) (√) ,(6) (×) ,(7) (×) ,(8) (√) ,(9) (×) ,(10) (√)

2、单选题

(1). A (2). A (3). C

3、挑错题

(1).C (2).D (3).B

4. 读程序题

(1). welcome! (2). 20 (3). 99 (4). 100:101

5. 编程题

```
import java.util.*;

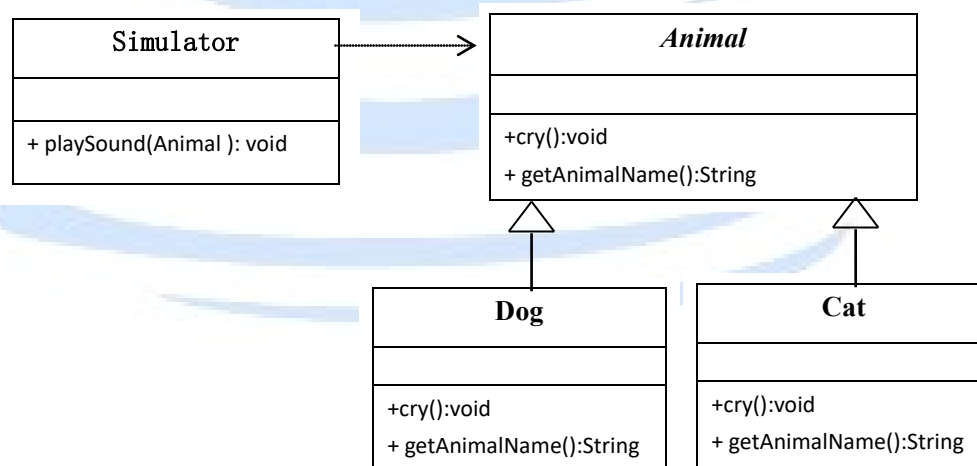
public class E {
    public static void main (String args[ ]){
        Scanner reader = new Scanner(System.in);
        double sum = 0;
        int m = 0;
        while(reader.hasNextDouble()){
            double x = reader.nextDouble();
            assert x< 100&& x>=0 : "数据不合理";
            m = m+1;
            sum = sum+x;
        }
        System.out.printf("%d 个数的和为%f\n",m,sum);
    }
}
```

习题 7

1.

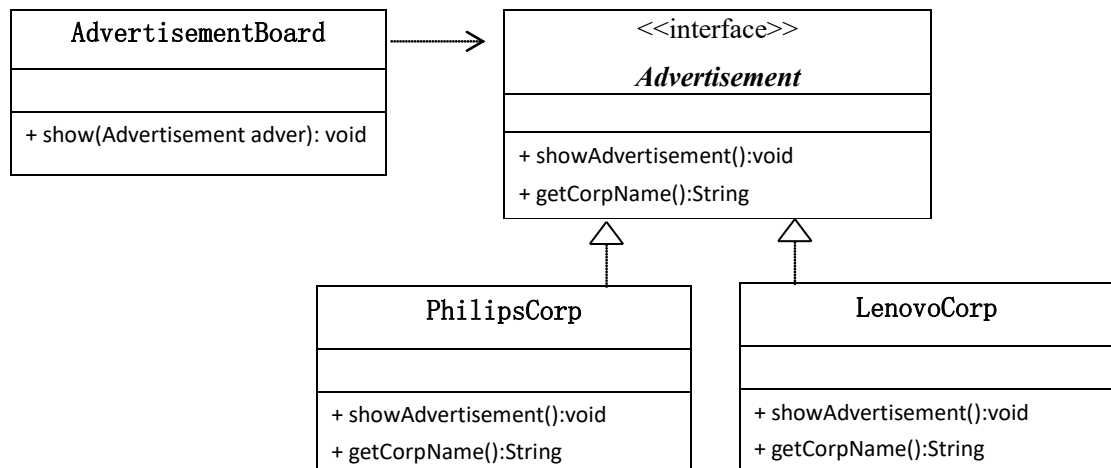
- (1) Strategy 是接口。
- (2) Army 不是抽象类。
- (3) Army 和 Strategy 是关联关系。
- (4) StrategyA, StrategyB、StrategyC 与 Strategy 是实现关系。

2.



第 5 章的例子 13 的 UML 类图

3.



第 5 章的例子 17 的 UML 类图

4. 例子 13 的设计符合开-闭原则。
5. 例子 17 的设计符合开-闭原则。

习题 8

1. 采用新增的策略为选手计算得分。

增加新的具体策略 **StrategyFour**。**StrategyFour** 类将 `double computeScore(double [] a)` 方法实现为去掉数组 **a** 的元素中的一个最大值和一个最小值，然后计算剩余元素的几何平均值。

```

import java.util.Arrays;
public class StrategyFour implements ComputableStrategy {
    public double computeScore(double [] a) {
        if(a.length<=2)
            return 0;
        double score=0,multi=1;
        Arrays.sort(a);
        int n=a.length-2;
        for(int i=1;i<a.length-1;i++) {
            multi=multi*a[i];
        }
        score=Math.pow(multi,1.0/n);
        return score;
    }
}
  
```

- 2.

(1)策略 (Strategy) **PrintCharacter.java**

```

public interface PrintCharacter{
    public abstract void printTable(char [] a,char[] b);
}
  
```

(2) 具体策略

PrintStrategyOne.java

```

public class PrintStrategyOne implements PrintCharacter {
    public void printTable(char [] a,char[] b) {
  
```

```

        for(int i=0;i<a.length;i++) {
            System.out.print(a[i]+",");
        }
        for(int i=0;i<b.length;i++) {
            System.out.print(b[i]+",");
        }
        System.out.println("");
    }
}
PrintStrategyTwo.java
public class PrintStrategyTwo implements PrintCharacter {
    public void printTable(char [] a,char[] b) {

        for(int i=0;i<a.length;i++) {
            System.out.print(b[i]+","+a[i]+",");
        }

    }
}

```

(3) 上下文 **PrintGame.java**

```

public class PrintGame {
    PrintCharacter strategy;
    public void setStrategy(PrintCharacter strategy) {
        this.strategy=strategy;
    }
    public void getPersonScore(char[] a,char[] b){
        if(strategy==null)
            System.out.println("sorry!");
        else
            strategy.printTable(a,b);
    }
}

```

应用以上策略:

```

public class Application {
    public static void main(String args[]) {
        char [] a=new char[26];
        char [] b=new char[26];
        for(int i=0;i<=25;i++){
            a[i]=(char)('a'+i);
        }
        for(int i=0;i<=25;i++){
            b[i]=(char)('A'+i);
        }
        PrintGame game=new PrintGame(); //上下文对象
        game.setStrategy(new PrintStrategyOne()); //上下文对象使用策略一

        System.out.println("方案 1:");
        game.getPersonScore(a,b);

        game.setStrategy(new PrintStrategyTwo()); //上下文对象使用策略二
        System.out.println("方案 2:");
        game.getPersonScore(a,b);
    }
}

```

习题 9

1、判断题

(1) (✓) , (2) (×) , (3) (✓) , (4) (×) , (5) (✓) , (6) (✓) , (7) (✓) , (8) (✓) , (9) (×) , (10) (✓)

2、单选题

(1). A (2). D (3). B (4). A (5). B (6). D (7). D

3、挑错题

(1). A (2). D (3). B

4.读程序题

(1). 20:58:18 (2). 7:hello (3). 苹果 (4). 4:6 日

5. 编程题

(1)

```
import java.util.*;

public class E {
    public static void main(String args[]) {
        String cost = "数学 87 分，物理 76 分，英语 96 分";
        Scanner scanner = new Scanner(cost);
        scanner.useDelimiter("[^0123456789.]+");
        double sum=0;
        int count =0;
        while(scanner.hasNext()){
            try{ double score = scanner.nextDouble();
                count++;
                sum = sum+score;
                System.out.println(score);
            }
            catch(InputMismatchException exp){
                String t = scanner.next();
            }
        }
        System.out.println("总分:"+sum+"分");
        System.out.println("平均分:"+sum/count+"分");
    }
}
```

(2)

```
import java.time.*;
import java.util.Scanner;

class GiveCalendar {
    public LocalDate [] getCalendar(LocalDate date) {
        date = date.withDayOfMonth(1); //确保 data 日期的 day 是 1，即 day 的值是 1
        int days = date.lengthOfMonth(); //得到该月有几天
        LocalDate dataArrays[] = new LocalDate[days];
        for(int i = 0;i<days;i++){
            dataArrays[i] = date.plusDays(i);
        }
    }
}
```

```
        return dataArrays;
    }
}

public class E {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("输入年:");
        int year = scanner.nextInt();
        System.out.print("输入月:");
        int month = scanner.nextInt();
        LocalDate date = LocalDate.of(year, month, 1);
        GiveCalendar giveCalendar = new GiveCalendar();
        LocalDate [] dataArrays = giveCalendar.getCalendar(date);
        printNameHead(date); //输出日历的头
        for(int i = 0; i < dataArrays.length; i++) {
            if(i == 0){
                printSpace(dataArrays[i].getDayOfWeek()); /
                System.out.printf("%4d", dataArrays[i].getDayOfMonth());
            }
            else {
                System.out.printf("%4d", dataArrays[i].getDayOfMonth());
            }
            if(dataArrays[i].getDayOfWeek() == DayOfWeek.SATURDAY)
                System.out.println(); //日历样式中的星期回行
        }
    }

    public static void printSpace(DayOfWeek x) { //输出空格
        switch(x) {
            case SUNDAY: printSpace(0);
                        break;
            case MONDAY: printSpace(1);
                        break;
            case TUESDAY: printSpace(2);
                        break;
            case WEDNESDAY: printSpace(3);
                        break;
            case THURSDAY: printSpace(4);
                        break;
            case FRIDAY: printSpace(5);
                        break;
            case SATURDAY: printSpace(6);
                        break;
        }
    }
}
```

```

public static void printSpace(int n){
    for(int i = 0;i<n;i++){
        System.out.printf("%4s","");//输出 4 个空格
    }
    public static void printNameHead(LocalDate date){ //输出日历的头
        System.out.println(date.getYear()+"年"+date.getMonthValue()+"月日历:");
        String name[] = {"日","一","二","三","四","五","六"};
        for(int i = 0;i<name.length;i++){
            System.out.printf("%3s",name[i]);
        }
        System.out.println();
    }
}

```

(3)

```

import java.time.*;
import java.util.Scanner;
import java.time.temporal.ChronoUnit;
public class E {
    public static void main(String args[] ) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("输入开始年的年份，月份和日期（空格分隔）:");
        int year = scanner.nextInt();
        int month = scanner.nextInt();
        int day = scanner.nextInt();
        LocalDate dateStart = LocalDate.of(year,month,day);
        System.out.print("输入结束年的年份，月份和日期（空格分隔）:");
        year = scanner.nextInt();
        month = scanner.nextInt();
        day = scanner.nextInt();
        LocalDate dateEnd = LocalDate.of(year,month,day);
        long days = dateStart.until(dateEnd,ChronoUnit.DAYS);
        System.out.println(dateStart+"和"+dateEnd+"相隔: ");
        System.out.println(days+"天(不足一天的零头按 0 计算)");
    }
}

```

习题 10**1、判断题**

(1) (√) ,(2) (×) ,(3) (√) ,(4) (×) ,(5) (√) ,(6) (√) ,(7) (√) ,(8) (√) ,(9) (×) ,(10) (√)

2、单选题

(1). D (2). A (3). D (4). B (5). B (6). C

3、挑错题

(1).A (2).D (3).B

4.读程序题

(1). good (2). 81

5. 编程题

(1) 有两个 java 源文件 Computer.java 和 E.java

Computer.java

```
import java.util.regex.Pattern;
import java.util.regex.Matcher;
public class Computer {
    double sum;
    double aver;
    public void setComputer(String input) {
        Pattern pattern;          //模式对象
        Matcher matcher;          //匹配对象
        String regex="-?[0-9][0-9]*[.]?[0-9]*" ;//匹配数字的正则表达式
        pattern = Pattern.compile(regex); //初始化模式对象
        matcher = pattern.matcher(input); //初始化匹配对象,用于检索 input
        sum = 0;
        aver = 0;
        int count = 0;
        while(matcher.find()) {
            String str = matcher.group();
            count ++;
            sum += Double.parseDouble(str);
        }
        aver = sum/count;
    }
    public double getSum(){
        return sum;
    }
    public double getAver(){
        return aver;
    }
}
```

E.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class E {
    public static void main(String args[]) {
        ComputerFrame fr=new ComputerFrame();
        fr.setTitle("计算的窗口");
    }
}
class ComputerFrame extends JFrame {
    JTextArea inputNumber,showResult;
    double sum=0,aver=0;
    Computer computer;
    public ComputerFrame() {
        computer = new Computer();
    }
}
```

```

        setLayout(new FlowLayout());
        inputNumber=new TextArea(6,20);
        showResult=new TextArea(6,20);
        add(inputNumber);
        add(showResult);
        showResult.setEditable(false);
        inputNumber.addTextListener((e)->{
            computer.setComputer(inputNumber.getText());
            String s= "和"+computer.getSum()+"\n"+
                "平均"+computer.getAver();
            showResult.setText(s);
        });
        setSize(300,320);
        setVisible(true);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        validate();
    }
}

```

(2) 有 1 个 java 源文件 E.java

E.java

```

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

public class E {
    public static void main(String args[]) {
        ComputerFrame fr=new ComputerFrame();
        fr.setTitle("计算");
    }
}

class ComputerFrame extends JFrame {
    JTextField inputNumber1,inputNumber2,showResult;
    Button buttonMultiAdd,buttonMultiSub,buttonMulti,buttonMultiDiv;
    JLabel showOperator;
    public ComputerFrame() {
        setLayout(new FlowLayout());
        inputNumber1=new JTextField(10);
        inputNumber2=new JTextField(10);
        showResult=new JTextField(10);
        showOperator=new JLabel(" ",showOperator.CENTER);
        showOperator.setBackground(Color.green);
        add(inputNumber1);
        add(showOperator);
        add(inputNumber2);
        add(showResult);
    }
}

```

```
buttonMultiAdd=new Button("加");
buttonMultiSub=new Button("减");
buttonMulti=new Button("乘");
buttonMultiDiv=new Button("除");
add(buttonMultiAdd);
add(buttonMultiSub);
add(buttonMulti);
add(buttonMultiDiv);
buttonMultiAdd.addActionListener((e)->{
    double n1,n2,n;
    try{ n1=Double.parseDouble(inputNumber1.getText());
        n2=Double.parseDouble(inputNumber2.getText());
        n=n1+n2;
        showResult.setText(String.valueOf(n));
        showOperator.setText("+");
    }
    catch(NumberFormatException ee){
        showResult.setText("请输入数字字符");
    }
});
buttonMultiSub.addActionListener((e)->{
    double n1,n2,n;
    try{ n1=Double.parseDouble(inputNumber1.getText());
        n2=Double.parseDouble(inputNumber2.getText());
        n=n1-n2;
        showResult.setText(String.valueOf(n));
        showOperator.setText("-");
    }
    catch(NumberFormatException ee){
        showResult.setText("请输入数字字符");
    }
});
buttonMulti.addActionListener((e)->{
    double n1,n2,n;
    try{ n1=Double.parseDouble(inputNumber1.getText());
        n2=Double.parseDouble(inputNumber2.getText());
        n=n1*n2;
        showResult.setText(String.valueOf(n));
        showOperator.setText("*");
    }
    catch(NumberFormatException ee){
        showResult.setText("请输入数字字符");
    }
});
```

```

        buttonMultiDiv.addActionListener((e)->{
            double n1,n2,n;
            try{ n1=Double.parseDouble(inputNumber1.getText());
                n2=Double.parseDouble(inputNumber2.getText());
                n=n1/n2;
                showResult.setText(String.valueOf(n));
                showOperator.setText("+");
            }
            catch(NumberFormatException ee){
                showResult.setText("请输入数字字符");
            }
        });
        setSize(300,320);
        setVisible(true);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        validate();
    }
}

```

习题 11

1. A
- 2.

```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
class Dwindow extends JFrame implements ActionListener{
    JTextField inputNumber;
    JTextArea save;
    Dwindow(){
        inputNumber=new JTextField(22);
        inputNumber.addActionListener(this);
        save=new JTextArea(12,16);
        setLayout(new FlowLayout());
        add(inputNumber);
        add(save);
        setBounds(60,60,300,300);
        setVisible(true);
        validate();
        setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
    }
    public void actionPerformed(ActionEvent event) {
        String s=inputNumber.getText();
        double n=0;
        try{
            n=Double.parseDouble(s);
            if(n>1000){
                int select=
                    JOptionPane.showConfirmDialog(this,"已经超过 1000 确认正确吗? ","确认对话框",
                                                    JOptionPane.YES_NO_OPTION );
                if(select==JOptionPane.YES_OPTION)
                    save.append("\n"+s);
            }
        }
    }
}

```

```

        else
            inputNumber.setText(null);
    }
    else {
        save.append("\n"+s);
    }
}
catch(NumberFormatException e){
    JOptionPane.showMessageDialog(this,"您输入了非法字符","警告对话框",
        JOptionPane.WARNING_MESSAGE);
    inputNumber.setText(null);
}
}
}
public class E {
    public static void main(String args[]) {
        new Dwindow();
    }
}

```

3.

```

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

public class E {
    public static void main(String args[]) {
        WindowColor win=new WindowColor();
        win.setTitle("带颜色对话框的窗口");
    }
}

class WindowColor extends JFrame {
    JTextArea text;
    JButton button;
    WindowColor() {
        button=new JButton("打开颜色对话框");
        text = new JTextArea(12,20);
        button.addActionListener((e)->{
            Color newColor=JColorChooser.showDialog(this,"调色板",button.getBackground());
            if(newColor!=null) {
                text.setForeground(newColor);
            }
        });
        setLayout(new FlowLayout());
        add(button);
        add(text);
        setBounds(60,60,300,300);
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```

习题 12

1. 使用 `FileInputStream` 流。
2. `FileInputStream` 按字节读取文件，`FileReader` 按字符读取文件。
3. 不能。
4. 使用对象流写入或读入对象时，要保证对象是序列化的。
5. 使用对象流很容易得获取一个序列化对象的克隆，只需将该对象写入到对象输出流，那么用对象输入流读回的对象一定是原对象的一个克隆。

6.

```
import java.io.*;
public class Xiti6 {
    public static void main(String args[]){
        File f=new File("E.java");;
        try{    RandomAccessFile random=new RandomAccessFile(f,"rw");
                random.seek(0);
                long m=random.length();
                while(m>=0){
                    m=m-1;
                    random.seek(m);
                    int c=random.readByte();
                    if(c<=255&&c>=0)
                        System.out.print((char)c);
                    else {
                        m=m-1;
                        random.seek(m);
                        byte cc[]=new byte[2];
                        random.readFully(cc);
                        System.out.print(new String(cc));
                    }
                }
            }
        catch(Exception exp){}
    }
}
```

7.

```
import java.io.*;
public class Xiti7 {
    public static void main(String args[ ]){
        File file=new File("E.java");
        File tempFile=new File("temp.txt");
        try{ FileReader    inOne=new FileReader(file);
                BufferedReader inTwo= new BufferedReader(inOne);
                FileWriter    tofile=new FileWriter(tempFile);
                BufferedWriter out= new BufferedWriter(tofile);
                String s=null;
                int i=0;
                s=inTwo.readLine();
                while(s!=null){
                    i++;
                    out.write(i+" "+s);
                    out.newLine();
                    s=inTwo.readLine();
                }
            }
        }
}
```

```

        inOne.close();
        inTwo.close();
        out.flush();
        out.close();
        tofile.close();
    }
    catch(IOException e){
        System.out.println(e);
    }
}
}
}

```

8. 属于操作题目，无解答。

9. 有 2 个源文件 EWindow.java, ReadProblem.java

EWindow.java

```

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
public class EWindow extends JFrame implements ItemListener{
    JButton start,next;
    JRadioButton checkbox[];    //显示单词选项
    JTextField showEnglishSentence,showScore;//显示英语句子和得分
    int score=0;
    ButtonGroup group;
    ReadProblem readProblem; //负责读取题目
    java.io.File file;
    EWindow(){
        readProblem = new ReadProblem();
        file = new java.io.File("English.txt");
        readProblem.setProblemFile(file);
        setTitle("英语单词学习");
        setLayout(new FlowLayout());
        showScore=new JTextField(10);
        showEnglishSentence = new JTextField(28);
        start=new JButton("重新练习");
        checkbox=new JRadioButton[4];
        group = new ButtonGroup();
        for(int i=0;i<=3;i++) {
            checkbox[i]=new JRadioButton("",false);
            group.add(checkbox[i]);
            checkbox[i].addItemListener(this);
        }
        start.addActionListener((e)->{
            score=0;
            readProblem.setProblemFile(file);
        });
        next=new JButton("下一题目");
        next.addActionListener((e)->{
            readProblem.readAnProbem();
            if(readProblem.englishSentence==null){
                showScore.setText("练习完毕");
                return;
            }
            showEnglishSentence.setText(readProblem.englishSentence);
            checkbox[0].setText(readProblem.word1);
            checkbox[1].setText(readProblem.word2);
            checkbox[2].setText(readProblem.word3);
            checkbox[3].setText(readProblem.word4);
        });
    }
}

```

```

    });
    add(new JLabel("题目"));
    add(showEnglishSentence);
    add(new JLabel("得分:"));
    add(showScore);
    add(new JLabel("选择:"));
    for(int i=0;i<=3;i++) {
        add(checkbox[i]);
    }
    add(next);
    add(start);
    setBounds(20,100,660,300);
    setVisible(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public void itemStateChanged(ItemEvent e){
    JRadioButton check =(JRadioButton)e.getSource();
    if(check.isSelected()){
        group.clearSelection();
        String s= check.getText().trim();
        if(s.equals(readProblem.answer)){
            score ++;
            showScore.setText(""+score);
        }
    }
}
}
public static void main(String args[]){
    EWindow w=new EWindow();
    w.validate();
}
}

```

ReadProblem.java

```

import java.io.*;
public class ReadProblem {
    String englishSentence ; //存放英语需要填空的英语句子
    String word1,word2,word3,word4;//存放四个选择
    String answer;           //存放答案
    FileReader in;
    BufferedReader inBuffer;
    public void setProblemFile(File f){
        try{
            in=new FileReader(f);
            inBuffer=new BufferedReader(in);
        }
        catch(FileNotFoundException exp){
            englishSentence =null;
        }
    }
    public void readAnProbpem(){
        String lineMess = null;
        try{
            lineMess = inBuffer.readLine();
            if(lineMess ==null || lineMess.startsWith("endend")){
                englishSentence =null;
                return;
            }
        }
    }
}

```



```

        lineMess =lineMess.trim();
        String []a=lineMess.split("#");
        englishSentence = a[0];
        word1 = a[1];
        word2 = a[2];
        word3 = a[3];
        word4 = a[4];
        answer = a[5];
    }
    catch(IOException exp){
        englishSentence =null;
    }
}
}

```

习题 13

1. 一个使用链式结构，一个使用顺序结构。
2. 8。
3. ABCD。
4. 选用 `HashMap<K, V>` 来存储。
- 5.

```

import java.util.*;
class UFlashKey implements Comparable {
    double d=0;
    UFlashKey (double d) {
        this.d=d;
    }
    public int compareTo(Object b) {
        UFlashKey st=(UFlashKey)b;
        if((this.d-st.d)==0)
            return -1;
        else
            return (int)((this.d-st.d)*1000);
    }
}
class UFlash {
    String name=null;
    double capacity,price;
    UFlash(String s,double m,double e) {
        name=s;
        capacity=m;
        price=e;
    }
}
public class Xiti5 {
    public static void main(String args[ ]) {
        TreeMap<UFlashKey,UFlash> treemap= new TreeMap<UFlashKey,UFlash>();
        String str[]={"U1","U2","U3","U4","U5","U6","U7","U8","U9","U10"};
        double capacity[]={1,2,2,4,0.5,10,8,4,4,2};
        double price[]={30,66,90,56,50,149,120,80,85,65};
        UFlash UFlash[]=new UFlash[10];
        for(int k=0;k<UFlash.length;k++) {
            UFlash[k]=new UFlash(str[k],capacity[k],price[k]);

```

```

    }
    UFlashKey key[]=new UFlashKey[10];
    for(int k=0;k<key.length;k++) {
        key[k]=new UFlashKey(UFlash[k].capacity); //关键字按容量成绩排列大小
    }
    for(int k=0;k<UFlash.length;k++) {
        treemap.put(key[k],UFlash[k]);
    }
    int number=treemap.size();
    System.out.println("树映射中有"+number+"个对象,按容量成绩排序:");
    Collection<UFlash> collection=treemap.values();
    Iterator<UFlash> iter=collection.iterator();
    while(iter.hasNext()) {
        UFlash stu=iter.next();
        System.out.println("U 盘 "+stu.name+" 容量 "+stu.capacity);
    }
    treemap.clear();
    for(int k=0;k<key.length;k++) {
        key[k]=new UFlashKey(UFlash[k].price); //关键字按价格成绩排列大小
    }
    for(int k=0;k<UFlash.length;k++) {
        treemap.put(key[k],UFlash[k]);
    }
    number=treemap.size();
    System.out.println("树映射中有"+number+"个对象:按价格成绩排序:");
    collection=treemap.values();
    iter=collection.iterator();
    while(iter.hasNext()) {
        UFlash stu=(UFlash)iter.next();
        System.out.println("U 盘 "+stu.name+" 价格 "+stu.price);
    }
}
}

```

习题 14

1. (1) 用管理员身份启动命令行窗口, 然后进入 MySQL 安装目录的 bin 子目录下键入 `mysqld` 或 `mysqld -nt`, 回车确认启动 MySQL 数据库服务器

(2) `java -cp jar 文件 1;jar 文件 2; 主类`

在 jar 文件和主类名之间用分号分隔, 而且分号和主类名之间必须留有至少一个空格。

(3) 使用预处理语句不仅减轻了数据库的负担, 而且也提高了访问数据库的速度。

(4) 事务由一组 SQL 语句组成, 所谓事务处理是指: 应用程序保证事务中的 SQL 语句要么全部都执行, 要么一个都不执行。步骤:

①使用 `setAutoCommit(boolean autoCommit)` 方法

con 对象首先调用 `setAutoCommit(boolean autoCommit)` 方法, 将参数 `autoCommit` 取值 `false` 来关闭默认设置:

`con.setAutoCommit(false);`

②使用 `commit()` 方法。con 调用 `commit()` 方法就是让事务中的 SQL 语句全部生效。

③使用 `rollback()` 方法。con 调用 `rollback()` 方法撤消事务中成功执行过的 SQL 语句对数据库数据所做的更新、插入或删除操作, 即撤消引起数据发生变化的 SQL 语句操作, 将数据库中的数据恢复到 `commi()` 方法执行之前的状态。

2. 编程题

1. 参看例子 3 代码。

2. 有 2 个源文件:E.java 和 Query.java。注意加载 Access 数据库连接器, 参见 14.13 。

```
java -cp Access_JDBC30.jar; E
```

注意在 jar 文件和主类名之间用分号分隔, 而且分号和主类名之间留有至少一个空格。

E.java

```
import javax.swing.*;

public class E {
    public static void main(String args[]) {
        String [] tableHead;
        String [][] content;
        JTable table ;
        JFrame win= new JFrame();
        Query findRecord = new Query();
        findRecord.setDatabaseName("Book.accdb");
        findRecord.setSQL("select * from bookList");
        content = findRecord.getRecord(); //返回二维数组, 即查询的全部记录
        tableHead=findRecord.getColumnNames(); //返回全部字段 (列) 名
        table = new JTable(content,tableHead);
        win.add(new JScrollPane(table));
        win.setBounds(12,100,400,200);
        win.setVisible(true); win.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    }
}
```

Query.java

```
import java.sql.*;

public class Query {
    String databaseName=""; //数据库名
    String SQL; //SQL 语句
    String [] columnName; //全部字段 (列) 名
    String [][] record; //查询到的记录
    public Query() {
        try{ Class.forName("com.hxtt.sql.access.AccessDriver");
        }
        catch(Exception e){}
    }
    public void setDatabaseName(String s) {
        databaseName=s.trim();
    }
    public void setSQL(String SQL) {
        this.SQL=SQL.trim();
    }
}
```

```
public String[] getColumnName() {
    if(columnName ==null ){
        System.out.println("先查询记录");
        return null;
    }
    return columnName;
}

public String[][] getRecord() {
    startQuery();
    return record;
}

private void startQuery() {
    Connection con;
    Statement sql;
    ResultSet rs;
    String uri = "jdbc:Access://" + databaseName;
    try {
        con=DriverManager.getConnection(uri,"","");
        sql=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                                ResultSet.CONCUR_READ_ONLY);

        rs=sql.executeQuery(SQL);
        ResultSetMetaData metaData = rs.getMetaData();
        int columnCount = metaData.getColumnCount();//字段数目
        columnName=new String[columnCount];
        for(int i=1;i<=columnCount;i++){
            columnName[i-1]=metaData.getColumnName(i);
        }
        rs.last();
        int recordAmount =rs.getRow(); //结果集中的记录数目
        record = new String[recordAmount][columnCount];
        int i=0;
        rs.beforeFirst();
        while(rs.next()) {
            for(int j=1;j<=columnCount;j++){
                record[i][j-1]=rs.getString(j);//第 i 条记录放入二维数组第 i 行
            }
            i++;
        }
        con.close();
    }
    catch(SQLException e) {
        System.out.println("请输入正确的表名"+e);
    }
}
```

}

习题 15

1. 4 种状态：新建、运行、中断和死亡。
2. 有 4 种原因的中断：(1) JVM 将 CPU 资源从当前线程切换给其他线程，使本线程让出 CPU 的使用权处于中断状态。(2) 线程使用 CPU 资源期间，执行了 `sleep(int millisecond)` 方法，使当前线程进入休眠状态。经过参数 `millisecond` 指定的毫秒数之后，该线程就重新进到线程队列中排队等待 CPU 资源，以便从中断处继续运行。(3) 线程使用 CPU 资源期间，执行了 `wait()` 方法，使得当前线程进入等待状态。等待状态的线程不会主动进到线程队列中排队等待 CPU 资源，必须由其他线程调用 `notify()` 方法通知它，使得它重新进到线程队列中排队等待 CPU 资源，以便从中断处继续运行。(4) 线程使用 CPU 资源期间，执行某个操作进入阻塞状态，比如执行读/写操作引起阻塞。进入阻塞状态时线程不能进入排队队列，只有当引起阻塞的原因消除时，线程才重新进到线程队列中排队等待 CPU 资源，以便从原来中断处开始继续运行。
3. 死亡状态，不能再调用 `start()` 方法。
4. 新建和死亡状态。
5. 两种方法：用 `Thread` 类或其子类。
6. 使用 `setPriority(int grade)` 方法。
7. Java 使我们可以创建多个线程，在处理多线程问题时，我们必须注意这样一个问题：当两个或多个线程同时访问同一个变量，并且一个线程需要修改这个变量。我们应对这样的问题作出处理，否则可能发生混乱。
8. 当一个线程使用的同步方法中用到某个变量，而此变量又需要其它线程修改后才能符合本线程的需要，那么可以在同步方法中使用 `wait()` 方法。使用 `wait` 方法可以中断方法的执行，使本线程等待，暂时让出 CPU 的使用权，并允许其它线程使用这个同步方法。其它线程如果在使用这个同步方法时不需要等待，那么它使用完这个同步方法的同时，应当用 `notifyAll()` 方法通知所有的由于使用这个同步方法而处于等待的线程结束等待。
9. 不合理。
10. “吵醒”休眠的线程。一个占有 CPU 资源的线程可以让休眠的线程调用 `interrupt` 方法“吵醒”自己，即导致休眠的线程发生 `InterruptedException` 异常，从而结束休眠，重新排队等待 CPU 资源。

```

11.
public class Xiti11 {
    public static void main(String args[]){
        Cinema a=new Cinema();
        a.zhang.start();
        a.sun.start();
        a.zhao.start();
    }
}

class TicketSeller {    //负责卖票的类。
    int fiveNumber=3,tenNumber=0,twentyNumber=0;
    public synchronized void    sellTicket(int receiveMoney) {
        if(receiveMoney==5) {
            fiveNumber=fiveNumber+1;

```

```

        System.out.println(Thread.currentThread().getName()+
            "给我 5 元钱, 这是您的 1 张入场卷");
    }
    else if(receiveMoney==10) {
        while(fiveNumber<1){
            try { System.out.println(Thread.currentThread().getName()+"靠边等");
                wait();
                System.out.println(Thread.currentThread().getName()+"结束等待");
            }
            catch(InterruptedException e) {}
        }
        fiveNumber=fiveNumber-1;
        tenNumber=tenNumber+1;
        System.out.println(Thread.currentThread().getName()+
            "给我 10 元钱,找您 5 元,这是您的 1 张入场卷");
    }
    else if(receiveMoney==20) {
        while(fiveNumber<1 || tenNumber<1){
            try { System.out.println(Thread.currentThread().getName()+"靠边等");
                wait();
                System.out.println(Thread.currentThread().getName()+"结束等待");
            }
            catch(InterruptedException e) {}
        }
        fiveNumber=fiveNumber-1;
        tenNumber=tenNumber-1;
        twentyNumber=twentyNumber+1;
        System.out.println(Thread.currentThread().getName()+
            "给 20 元钱, 找您一张 5 元和一张 10 元, 这是您的 1 张入场卷");
    }
    notifyAll();
}
}

class Cinema implements Runnable {
    Thread zhang,sun,zhao;
    TicketSeller seller;
    Cinema(){
        zhang=new Thread(this);
        sun=new Thread(this);
        zhao=new Thread(this);
        zhang.setName("张小有");
        sun.setName("孙大名");
        zhao.setName("赵中堂");
        seller=new TicketSeller();
    }
    public void run(){
        if(Thread.currentThread()==zhang)
            seller.sellTicket(20);
        else if(Thread.currentThread()==sun)
            seller.sellTicket(10);
        else if(Thread.currentThread()==zhao)
            seller.sellTicket(5);
    }
}

```

12. 参照本章例子 9。

13. BA

习题 16

1. URL 对象调用 `InputStream openStream()` 方法可以返回一个输入流。
2. 客户端的程序使用 `Socket` 类建立负责连接到服务器的套接字对象称为 `socket` 对象。使用 `Socket` 的构造方法 `Socket(String host,int port)`，建立连接到服务器的套接字对象。参考 16.3.2
3. `JEditorPane`
4. 会返回一个和客户端 `Socket` 对象相连接的 `Socket` 对象。
5. 域名/IP 地址 例如，`www.sina.com.cn/202.108.35.210`
6. 参照例子 16.6，只需将例子中的园更改为三角形即可。
7. 参照本章例子 16.6 的步骤和代码，再结合输入流即可。

