目录

Day15. Java

1 字符编码

- ASC-II
  0到127
  包含英文字符、标点、指令字符

- ISO-8859-1、Latin-1
  西欧字符编码
  160到255

- GB2312
  国标2312
  双字节编码
  最大到 65535
  7k+ 不包含 喆，镕

- GBK
  在GB2312基础上，添加了更多中文字符

- Unicode 编码表
  - 万国码、统一码
  - 100万+
  - 常用字符表，双字节
  - 生僻字符表，三字节、四字节...
- UTF-8
  - Unicode传输格式

Unicode Transfer Format
- 英文，单字节
- 某些字符，双字节
- 中文，三字节
- 特殊符号，四字节

- Java的char类型字符，采用 Unicode编码
  - 'a' 00 61
  - '中' 4e 2d
- GBK
  - a 61
  - 中 d6 d0
- UTF-8
  - a 61
  - 中 e4 b8 ad

1.1 Java字符编码转换

- Unicode --> 其他编码

```
String s = "abc中文";
//转成系统默认编码
byte[] a = s.getBytes();

//转成指定的编码
byte[] a = s.getBytes("UTF-8");
```

- 其他编码 --> Unicode

```
//从默认编码转换
new String(byte[]数组);

//从指定的编码转换
new String(byte[]数组, "UTF-8")
```

编码转换

项目：day1501_编码转换
类：day1501.Test1

```
package day1501;

import java.io.UnsupportedEncodingException;
import java.util.Arrays;

public class Test1 {
   public static void main(String[] args) throws Exception {
      String s = "abc中文喆镕";
      System.out.println(s);

      byte[] a;
      a = f(s, null);
      System.out.println(Arrays.toString(a));
      a = f(s, "GBK");
      System.out.println(Arrays.toString(a));
```

```
        a = f(s, "GB2312");
        System.out.println(Arrays.toString(a));
        a = f(s, "UTF-8");
        System.out.println(Arrays.toString(a));
    }

    /*
     * 字符编码 encoding
     * 字符集 charset
     */
    private static byte[] f(
            String s, String charset) throws Exception {
        if(charset == null) {
            return s.getBytes();
        } else {
            return s.getBytes(charset);
        }
    }
}
```

Test2

```
package day1501;

import java.io.UnsupportedEncodingException;

public class Test2 {
    public static void main(String[] args) throws Exception {
        byte[] a;

        a = new byte[] {97, 98, 99, -42, -48, -50, -60, -122, -76, -23, 70};
        f(a, null);
        a = new byte[] {97, 98, 99, -42, -48, -50, -60, -122, -76, -23, 70};
        f(a, "GBK");
        a = new byte[] {97, 98, 99, -42, -48, -50, -60, 63, 63};
        f(a, "GB2312");
        a = new byte[] {97, 98, 99, -28, -72, -83, -26, -106, -121, -27, -106, -122,
-23, -107, -107};
        f(a, "UTF-8");
    }

    private static void f(byte[] a, String charset) throws Exception {
        String s;

        if(charset == null) {
            s = new String(a);
        } else {
            s = new String(a, charset);
        }

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

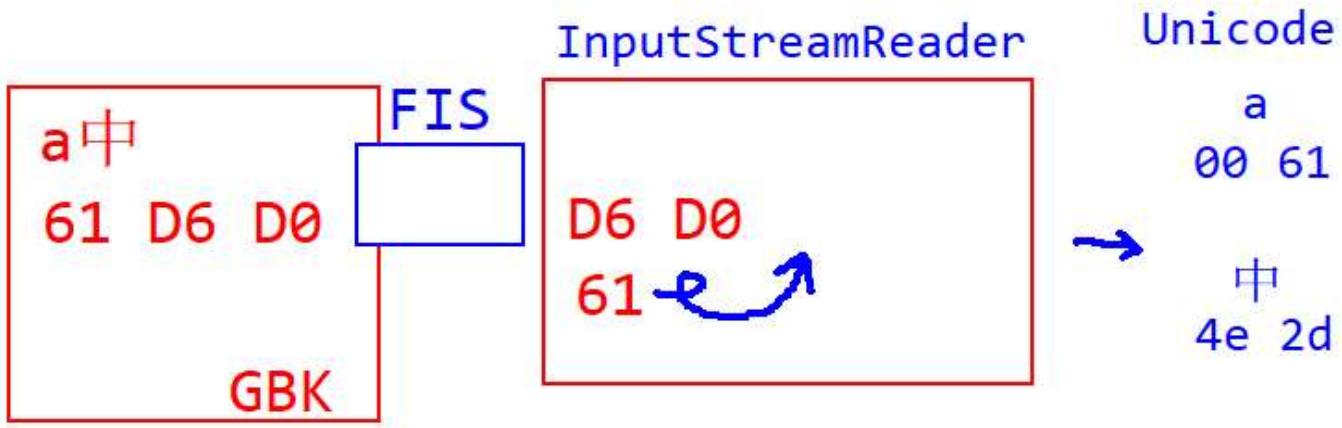
2 Reader / Writer

- 字符流的抽象父类
- 以字符为单位读写数据
- 方法
  - write(int c)
    int末尾的两个字节值，是char类型字符
    输出末尾的两个字节
  - write(char[] buff)
    输出全部
  - write(char[] buff,start,length)
    输出从start开始的length个
  - write(String s)
    输出字符串中全部字符
  - read()
    读取一个字符，补两个0字节，转成int

    读取结束，再读取，返回 -1
  - read(char[] buff)
    批量读取

3 InputStreamReader/OutputStreamWriter

- 字符编码转换流
- InputStreamReader
  - 读取其他编码的字节值，转换成Unicode



- OutputStreamWriter
  - 把 Unicode 转成其他编码字节值，输出

编码转换流

项目：day1502_编码转换流
类：day1502.Test1

```
package day1502;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;

public class Test1 {
   public static void main(String[] args) throws Exception {
      /*
       * Unicode转换成其他编码存到文件
       * OSW--FOS--f3, GBK
       * OSW--FOS--f4, UTF-8
       *
       * Unicode编码表中所有的中文字符：
       * \u4e00到\u9fa5, 20902个中文
       */
      OutputStreamWriter out1 =
       new OutputStreamWriter(
        new FileOutputStream("d:/abc/f3"), "GBK");
      OutputStreamWriter out2 =
       new OutputStreamWriter(
       new FileOutputStream("d:/abc/f4"), "UTF-8");

      out1.write("Unicode编码表中所有的中文字符：\n");
      out2.write("Unicode编码表中所有的中文字符：\n");
      int count = 0;
      for(char c='\u4e00';c<='\u9fa5';c++) {
         out1.write(c);
         out2.write(c);
         count++;
         if(count == 30) {
            out1.write('\n');
            out2.write('\n');
            count = 0;
         }
      }
      out1.close();
      out2.close();
   }

}
```

4 BufferdReader/BufferedWriter

- 字符缓冲流
- readLine()
  - 读取一行字符串，不包含末尾的换行符
  - 读取结束，再读取，返回 null

5 PrintWriter

PrintStream

- 打印输出流
- print()、println()
- PrintStream 只能接字节流
- PrintWriter 能接字节流，也能接字符流

随机点餐

录入经常点的餐
名称(exit结束)：鱼香肉丝
名称(exit结束)：宫保鸡丁
名称(exit结束)：石锅拌饭
名称(exit结束)：拉面
名称(exit结束)：exit

文件foods.txt
鱼香肉丝;2
宫保鸡丁;0
石锅拌饭;1
拉面;0

随机点餐

项目：day1503_随机点餐
类：day1503.Test1

```
package day1503;

import java.io.FileOutputStream;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.net.URLDecoder;
import java.util.ArrayList;
import java.util.Scanner;

public class Test1 {
   public static void main(String[] args) {
      //录入食物列表
      ArrayList<String> list = luRu();
      //列表中的食物，保存到文件
      baoCun(list);
   }
```

```
    private static ArrayList<String> luRu() {
        System.out.println("录入经常点的餐:");
        ArrayList<String> list = new ArrayList<>();
        while(true) {
            System.out.println("名称(exit结束): ");
            String s = new Scanner(System.in).nextLine();
            if("exit".equals(s)) {
                break;
            }
            list.add(s);
        }
        return list;
    }

    private static void baoCun(ArrayList<String> list) {
        //获得文件的路径
        try {
            String path = getPath();
            //PW--OSW--FOS--path
            PrintWriter out =
             new PrintWriter(
             new OutputStreamWriter(
             new FileOutputStream(path), "GBK"));

            for (String s : list) {
                out.println(s+";0");
            }

            out.close();
            System.out.println("文件已保存");
        } catch (Exception e) {
            System.out.println("保存文件失败");
            e.printStackTrace();
        }
    }

    private static String getPath() throws Exception {
        //D:\lesson\1810\ws1810\day1503_随机点餐\src\foods.txt
        /*
         * *)用相对路径的方式，来获取绝对路径
         * *) "/" 表示程序的运行目录，bin目录
         * *) "/foods.txt" 表示 bin/foods.txt
         */

        //文件必须存在，不存在会得到null
        String path =
         Test1.class
         .getResource("/foods.txt")
         .getPath();

        //%e9%9a%8f url解码回成正确的中文：随
        path = URLDecoder.decode(path, "UTF-8");

        return path;
    }
}
```

Test2

```
package day1503;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.net.URLDecoder;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.Comparator;
import java.util.Random;
import java.util.Scanner;

public class Test2 {
   public static void main(String[] args) {
      try {
         //加载foods.txt文件，获得食物列表
         ArrayList<Food> list = load();
         if(list.size() == 0) {
            System.out.println(
             "请先运行Test1，录入食物");
            return;
         }
         System.out.print("随机挑选几个食物：");
         int n = new Scanner(System.in).nextInt();
         //随机选择食物
         Food[] foods = suiJi(list, n);
         //显示挑选的结果
         show(foods);
         //重新保存食物列表
         baoCun(list);
      } catch (Exception e) {
         System.out.println("出现错误");
         e.printStackTrace();
      }
   }

   private static ArrayList<Food> load() throws Exception {
      String path = getPath();
      //BR--ISR--FIS--path
      BufferedReader in =
       new BufferedReader(
       new InputStreamReader(
```

```
         new FileInputStream(path), "GBK"));
      ArrayList<Food> list = new ArrayList<>();
      String line;
      while((line = in.readLine()) != null) {
         //"aaaa;0" --> ["aaaa", "0"]
         String[] a = line.split(";");
         Food f = new Food();
         f.setName(a[0]);
         f.setCount(Integer.parseInt(a[1]));
         list.add(f);
      }
      Collections.sort(list, new Comparator<Food>() {
         @Override
         public int compare(Food o1, Food o2) {
            return o1.getCount()-o2.getCount();
         }
      });

      in.close();
      return list;
   }

   private static String getPath() throws Exception {
      String path =
       Test1.class
       .getResource("/foods.txt")
       .getPath();
      return URLDecoder.decode(path, "UTF-8");
   }

   private static Food[] suiJi(
         ArrayList<Food> list, int n) {
      if(n>list.size()) {
         return list.toArray(new Food[list.size()]);
      }
      if(n<1) {
         n = 1;
      }

      Food[] a = new Food[n];
      for (int i = 0; i < a.length;  ) {
         int mid = list.size()/2;//中间下标
         Food f;
         if(Math.random()<0.7) {
            // [0, mid) 范围随机选取1个
            f = list.get(
             new Random().nextInt(mid));
         } else {
            // mid+[0, list.size()-mid) 范围随机选取1个
            f = list.get(
             mid+new Random().nextInt(list.size()-mid));
         }
         //把a数组，转成List集合
         //再判断集合中是否包含指定的对象
         if(Arrays.asList(a).contains(f)) {
            continue;
         }
         a[i] = f;
         //挑选次数+1
```

```
            a[i].setCount(a[i].getCount()+1);
            i++;
        }
        return a;
    }

    private static void show(Food[] foods) {
        System.out.println("为您随机点餐: ");
        for (Food f : foods) {
            System.out.println(f.getName());
        }
    }

    private static void baoCun(ArrayList<Food> list) throws Exception {
        String path = getPath();
        //PW--OSW--FOS--path
        PrintWriter out =
         new PrintWriter(
         new OutputStreamWriter(
         new FileOutputStream(path), "GBK"));

        for (Food f : list) {
            out.println(f);
        }

        out.close();
        //System.out.println("文件已保存");
    }
}
```

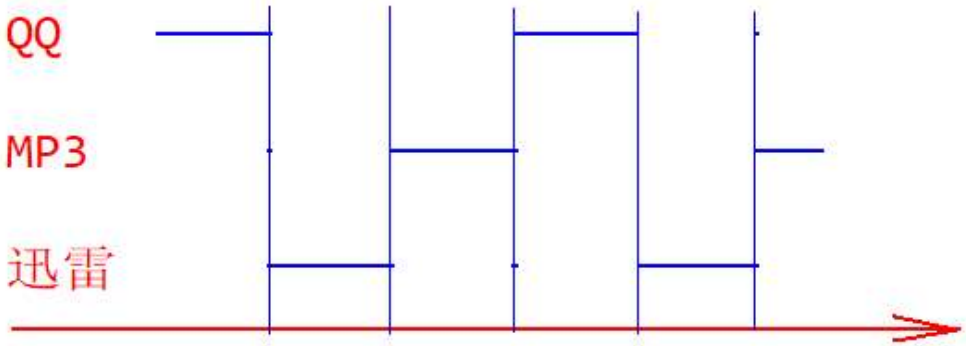
Food

```
package day1503;

public class Food {
    private String name;
    private int count;

    public Food() {
        super();
    }
    public Food(String name, int count) {
        super();
        this.name = name;
        this.count = count;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getCount() {
        return count;
    }
    public void setCount(int count) {
        this.count = count;
    }
```

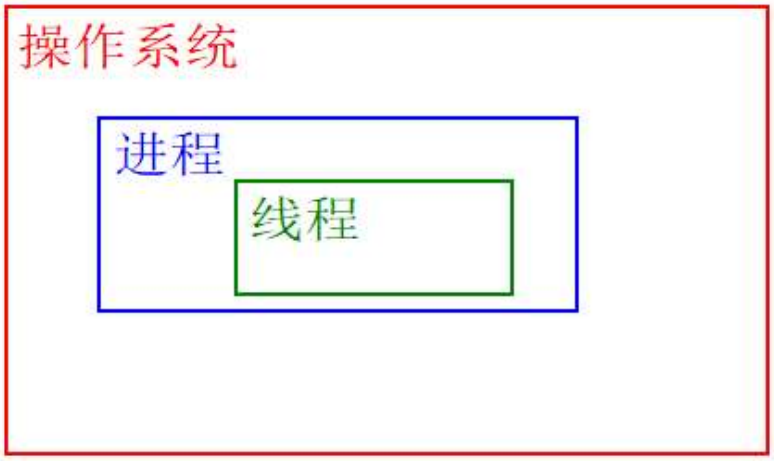
```
    @Override
    public String toString() {
        return name+";"+count;
    }
}
```

6 线程

- 进程
  - 操作系统中，并行执行的任务
  - cpu时间被切分成一个一个小的时间片
  - 一个时间片上，只能处理一项任务



- 线程
  - 进程内部并行执行的任务



6.1 创建线程（两种方式）

- 继承Thread
- 实现Runnable

6.1.1 继承Thread

- 继承Thread，定义一个线程子类
- 子类中，重写run()方法
- 包含在run()方法中的代码，是与其他线程并行执行的代码
- 线程启动后，自动执行run()方法

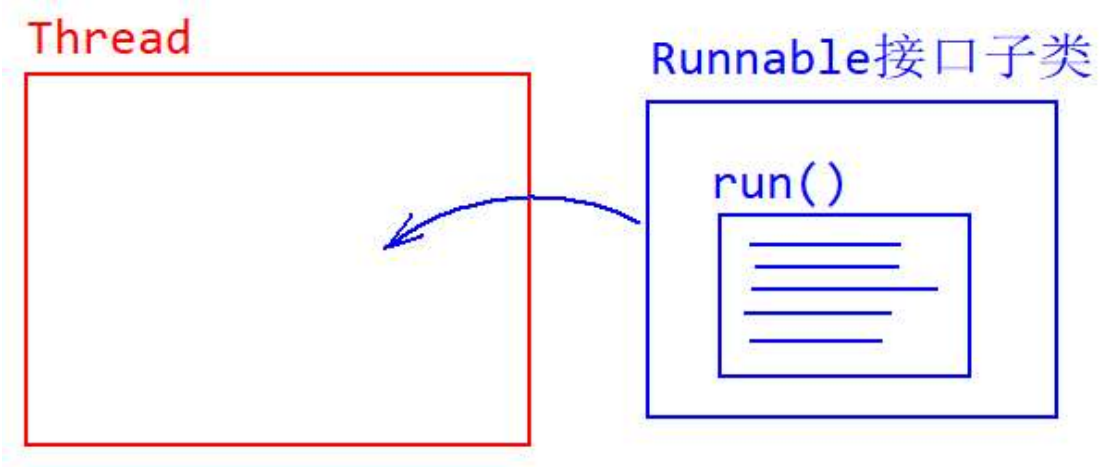
线程

项目：day1504_线程
类：day1504.Test1

```
package day1504;

public class Test1 {
   public static void main(String[] args) {
      T1 t1 = new T1();
      T1 t2 = new T1();
      //启动线程
      //线程启动后，自动执行run()方法
      t1.start();
      t2.start();

      //虚拟机启动后，会自动创建一个main线程
      //执行main()方法中的代码
      System.out.println("main");
   }


   static class T1 extends Thread {
      @Override
      public void run() {
         //获取线程名
         String n = getName();
         //打印1到1000
         for(int i=1;i<=1000;i++) {
            System.out.println(n+" - "+i);
         }
      }
   }

}
```

6.1.2 实现Runnable

- 继承 Runnable 接口
- 实现 run() 方法
- 把 Runnable 对象，放入线程对象，并启动
- 线程启动后，执行 Runnable对象的run()方法



Runnable

Test2

```
package day1504;

public class Test2 {
   public static void main(String[] args) {
      R1 r1 = new R1();
      Thread t1 = new Thread(r1);
      Thread t2 = new Thread(r1);
      t1.start();
      t2.start();
   }

   static class R1 implements Runnable {
      @Override
      public void run() {
         //获得正在执行的线程对象
         Thread t = Thread.currentThread();
         //获得线程名
         String n = t.getName();
         for(int i=1;i<=1000;i++) {
            System.out.println(n+" - "+i);
         }
      }
   }
}
```

7 作业

- 重写
  - day1503_随机点餐
    Food
    Test2
- 手写双向链表