# 考试班级:K9503第一次模拟考试

# 选择题部分

|  |  |
| --- | --- |
| 序号 | 1 |
| 题干 | 关于LinkedList类，错误的描述是（）。 |
| 题型 | 单选题 |
| A | LinkedList是以链表形式进行数据存放的 |
| B | LinkedList继承(extends)自List |
| C | LinkedList可以实现栈，堆这样的数据结构 |
| D | LinkedList实现(implements)了Collection |

|  |  |
| --- | --- |
| 序号 | 2 |
| 题干 | 在Java中，已有public int sum(){return 0;}方法，在同一个类中以下不能与此方法构成重载的是（ ）。 |
| 题型 | 单选题 |
| A | public int sum(int start,int end){ return 10;} |
| B | public int sum(int start){ return 10;} |
| C | public double sum(){return 10;} |
| D | public int sum(double start){return 10;} |

|  |  |
| --- | --- |
| 序号 | 3 |
| 题干 | Java中，集合API 中Set接口的特点是（ ）。 |
| 题型 | 单选题 |
| A | 不允许重复元素，元素有顺序 |
| B | 允许重复元素，元素无顺序 |
| C | 允许重复元素，元素有顺序 |
| D | 不允许重复元素，元素无顺序 |

|  |  |
| --- | --- |
| 序号 | 4 |
| 题干 | 在Java程序中，使用（ ）关键字告知编译器所要使用的类位于哪一个包中，这个过程称为导入包。 |
| 题型 | 单选题 |
| A | import |
| B | package |
| C | system |
| D | using |

|  |  |
| --- | --- |
| 序号 | 5 |
| 题干 | 关于Java中static关键字的说法错误的是（ ）。 |
| 题型 | 单选题 |
| A | static可以用来修饰属性、方法和代码块 |
| B | static修饰的属性和方法可称为类属性、类方法 |
| C | 不使用static关键字修饰的属性和方法，通常称为实例属性、实例方法 |
| D | 使用static修饰的变量和方法只能使用类名来访问，不能使用对象名来访问 |

|  |  |
| --- | --- |
| 序号 | 6 |
| 题干 | 在java中，throws的作用是：（） |
| 题型 | 单选题 |
| A | 表示方法可能会抛出异常 |
| B | 表示后面是方法的输出量 |
| C | 方法的标志，每个方法都必须有 |
| D | 没有意义 |

|  |  |
| --- | --- |
| 序号 | 7 |
| 题干 | 在Java中，以下说法错误的是（ ）。 |
| 题型 | 单选题 |
| A | Java中的类文件可以存储在不同的包中，有效防止命名冲突 |
| B | 可以使用package创建一个包，并将package命令作为一个Java源文件的第一条非注释性语句 |
| C | Java的包名通常由小写字母组成，可以以圆点开头或结尾 |
| D | 要使用不在同一个包中的类，可以通过import导入包 |

|  |  |
| --- | --- |
| 序号 | 8 |
| 题干 | Java中，下列说法错误的是（）。 |
| 题型 | 单选题 |
| A | ArrayList是线程不安全的 |
| B | Vector是线程安全的 |
| C | ArrayList比Vector的性能低 |
| D | ArrayList和Vector都是基于数组实现的List类 |

|  |  |
| --- | --- |
| 序号 | 9 |
| 题干 | Java中，关于HashMap和Hashtable的说法错误的是（）。 |
| 题型 | 单选题 |
| A | HashMap和Hashtable都是Map接口的实现类 |
| B | Hashtable是线性安全的 |
| C | HashMap是线性不安全的 |
| D | Hashtable要比HashMap的性能高一点 |

|  |  |
| --- | --- |
| 序号 | 10 |
| 题干 | Java中，创建一个只能存放String的泛型ArrayList的语句是（）。 |
| 题型 | 单选题 |
| A | ArrayList<int> al=new ArrayList<int>()； |
| B | ArrayList<String> al=new ArrayList<String>(); |
| C | ArrayList al=new ArrayList<String>()； |
| D | ArrayList<String> al =new List<String>(); |

|  |  |
| --- | --- |
| 序号 | 11 |
| 题干 | Java中，没有实现toString（）方法的情况下直接用print打印对象，输出的是（ ）。 |
| 题型 | 单选题 |
| A | 一个字符串，该字符串由类名、at 标记符“@”和此对象哈希码的无符号十六进制表示组成。 |
| B | 此对象的属性值 |
| C | 会报错 |
| D | 什么都不输出 |

|  |  |
| --- | --- |
| 序号 | 12 |
| 题干 | 关于Java语言中多态的说法错误的是（ ）。 |
| 题型 | 单选题 |
| A | 多态是面向对象三大特性之一 |
| B | 通过多态可以提高代码的可扩展性和可维护性 |
| C | 把子类转换为父类，称为向下转型 |
| D | 使用父类作为方法的形参是使用多态的常用方式 |
| 序号 | 13 |
| 题干 | 以下关于Java中封装描述正确的是（ ）。 |
| 题型 | 单选题 |
| A | 封装是面向对象的三大特性之一，是将类的状态信息隐藏在类的内部，不允许外部程序访问。 |
| B | 封装是修改属性的可见性来限制对属性的访问。为每个属性创建一对取值（setter）方法和赋值（getter）方法，用于对这些属性的存取。 |
| C | 封装的好处主要有：隐藏类的实现细节；限制不合理操作。 |
| D | 进行封装时，必须提供有参数的构造方法，为属性赋初始值。 |

|  |  |
| --- | --- |
| 序号 | 14 |
| 题干 | 在Java中，下列说法中错误的是( )。 |
| 题型 | 单选题 |
| A | continue和break 的作用是一样的 |
| B | continue和break可以出现在同一个循环体中 |
| C | 程序执行到continue时跳出当次循环 |
| D | 程序执行到break时跳出整个循环 |
| 序号 | 15 |
| 题干 | Java中，一个类能被同一包或不同包中的其他类访问的修饰符（ ）。 |
| 题型 | 单选题 |
| A | private |
| B | protected |
| C | public |
| D | 默认访问修饰符 |

|  |  |
| --- | --- |
| 序号 | 16 |
| 题干 | 在Java中，数组下标越界出现的异常是（ ）。 |
| 题型 | 单选题 |
| A | NullPointerException |
| B | ClassCastException |
| C | NumberFormatException |
| D | ArrayIndexOutOfBoundsException |

|  |  |
| --- | --- |
| 序号 | 17 |
| 题干 | 在Java中，以下对final的使用，说法错误的是（ ）。 |
| 题型 | 单选题 |
| A | final用于修饰类，表示类不可被继承 |
| B | final用于修饰变量，不可以被修改 |
| C | 被final修饰的方法，不可以被重写 |
| D | final可同时与abstract同时修饰一个类或方法 |

|  |  |
| --- | --- |
| 序号 | 18 |
| 题干 | 在Java中，抛出异常使用（ ）关键字。 |
| 题型 | 单选题 |
| A | throw |
| B | throwable |
| C | throwing |
| D | catch |

|  |  |
| --- | --- |
| 序号 | 19 |
| 题干 | Java接口中的修饰符可以为（）。 |
| 题型 | 多选题 |
| A | private |
| B | protected |
| C | final |
| D | abstract |
| 序号 | 20 |
| 题干 | 如下代码，代码正确执行后，控制台打印（ ）。  int[] ages = {7,8,9};  for(int i : ages){  System.out.print(i);  } |
| 题型 | 单选题 |
| A | 0 |
| B | 1 |
| C | 2 |
| D | 789 |

|  |  |
| --- | --- |
| 序号 | 21 |
| 题干 | 下面关于方法的说法，不正确的是（）。 |
| 题型 | 单选题 |
| A | Java中的构造方法名必须和类名相同 |
| B | 方法体是对方法的实现，包括变量声明和合法语句 |
| C | 如果一个类定义了构造方法，也可以用该类的默认构造方法 |
| D | 类的私有方法不能被其他类直接访问 |

|  |  |
| --- | --- |
| 序号 | 22 |
| 题干 | 在Java中一个类可同时定义许多同名的方法，这些方法的形式参数个数、类型和顺序各不相同，传回的值也可以不相同这种面向对象程序的特性称为（ ）。 |
| 题型 | 单选题 |
| A | 隐藏 |
| B | 覆盖 |
| C | 重载 |
| D | Java不支持此特性 |

|  |  |
| --- | --- |
| 序号 | 23 |
| 题干 | 在下面的Java代码中，能填入代码横线的选项是（）。  public class Phone {  String phone;  public String getPhone() {  return phone;  }  public void setPhone(String nphone) {  phone = nphone;  }  public void showPhone(){  System.out.println(\_\_\_\_);  }  } |
| 题型 | 单选题 |
| A | this |
| B | this() |
| C | this.phone |
| D | this.phone() |

|  |  |
| --- | --- |
| 序号 | 24 |
| 题干 | 在Java中，下面关于ArrayList集合的说法错误的是（ ）。 |
| 题型 | 单选题 |
| A | ArrayList的存储空间是连续的 |
| B | ArrayList可以通过下标访问数据 |
| C | ArrayList遍历和随机访问效率较高 |
| D | ArrayList插入和删除效率较高 |

|  |  |
| --- | --- |
| 序号 | 25 |
| 题干 | 在Java中下列哪些集合是属于线程安全的（ ）。 |
| 题型 | 多选题 |
| A | HashTable |
| B | HashSet |
| C | Vector |
| D | HashMap |
| 序号 | 26 |
| 题干 | 在Java中，以下关于数组的说法不正确的是（ ）。 |
| 题型 | 多选题 |
| A | 数组通过下标访问每个元素，数组下标从0开始 |
| B | 数组中元素的类型可以不相同 |
| C | 数组一旦定义了，大小长度就不能改变了 |
| D | 对于长度大于1的数组arr，获得最后一个元素可以使用arr[arr.length()-1] |

|  |  |
| --- | --- |
| 序号 | 27 |
| 题干 | 下列说法正确的是：（）。 |
| 题型 | 单选题 |
| A | 一个子类可以有多个父类，一个父类也可以有多个子类 |
| B | 一个子类可以有多个父类，但一个父类只可以有一个子类 |
| C | 一个子类只有一个父类，但一个父类可以有多个子类 |
| D | 上述说法都不对 |

|  |  |
| --- | --- |
| 序号 | 28 |
| 题干 | 在子类callFa方法中调用父类方法fly，代码正确的是：（） 。 |
| 题型 | 单选题 |
| A | public static fly（）{super.callFa()} |
| B | public void fly（）{super.fly()} |
| C | public void callFa(){ super.fly()} |
| D | public static void callFa(){super.fly()} |

|  |  |
| --- | --- |
| 序号 | 29 |
| 题干 | 在Java中，以下关于List接口的说法错误的是（ ）。 |
| 题型 | 单选题 |
| A | ArrayList采用了和数组相同的存储方式，在内存中分配连续的空间 |
| B | LinkedList的优点在于插入、删除元素时效率比较高 |
| C | ArrayList的优点在于遍历元素和随机访问元素的效率比较高 |
| D | ArrayList和LinkedList都不允许存储重复数据 |

|  |  |
| --- | --- |
| 序号 | 30 |
| 题干 | Java中，用迭代器Iterator的（）方法遍历集合中的元素，判断集合中是否仍有元素可以迭代。 |
| 题型 | 单选题 |
| A | hasNext() |
| B | next() |
| C | isNext() |
| D | remove() |

|  |  |
| --- | --- |
| 序号 | 31 |
| 题干 | 阅读以下Java程序代码，作出分析，运行的结果是：（）。  public class SubClass {  public static void main(String[] args) {  int a=123;  String b="123";  System.out.println(a==b);  }  } |
| 题型 | 单选题 |
| A | true |
| B | false |
| C | 编译错误 |
| D | 运行错误 |

|  |  |
| --- | --- |
| 序号 | 32 |
| 题干 | 在Java中，关于方法重载的说法错误的是（ ）。 |
| 题型 | 单选题 |
| A | 方法重载要求方法名称必须相同 |
| B | 重载方法的参数列表必须不一致 |
| C | 重载方法的返回类型可以不一致 |
| D | 一个方法在所属的类中只能被重载一次 |

|  |  |
| --- | --- |
| 序号 | 33 |
| 题干 | 下列语句执行后i的值是（ ）。  int i = 8,j = 16;  if(i-1>j){  i--;  }else{  j--;  } |
| 题型 | 单选题 |
| A | 15 |
| B | 16 |
| C | 8 |
| D | 7 |
| 序号 | 34 |
| 题干 | 在Java中，以下接口的定义，正确的为（ ）。 |
| 题型 | 单选题 |
| A | //A1、A2 ,A3为已定义的接口  interface B {  void print();  } |
| B | abstract interface B {  void print(){}  } |
| C | //A1、A2为已定义的接口  interface B implements A1,A2 {  void print();  } |
| D | interface B {  void print(){}  } |
| 序号 | 35 |
| 题干 | 在Java中，以下程序的运行结果是（ ）。  public class Main {  public static void main(String[] args) {  int i = 0, sum = 0;  while (i<= 10) {  sum += i;  i++;  if (i % 2 == 0){  continue;  }  }  System.out.println(sum);  }  } |
| 题型 | 单选题 |
| A | 0 |
| B | 55 |
| C | 50 |
| D | 36 |

|  |  |
| --- | --- |
| 序号 | 36 |
| 题干 | 某Java类的main方法如下所示，则编译运行该类的输出结果是（ ）。  public static void main(String[] args) {  for(int i=0;i<10;i++){  if(i%2!=0){  return;  }  System.out.print(i);  }  } |
| 题型 | 单选题 |
| A | 13578 |
| B | 02468 |
| C | 0123456789 |
| D | 0 |

|  |  |
| --- | --- |
| 序号 | 37 |
| 题干 | Java中，设i、j、k为类x中定义的int型变量名，下列类x的构造函数中错误的是（ ）。 |
| 题型 | 单选题 |
| A | x( int m){ ... } |
| B | void x( int m){ ... } |
| C | x( int m, int n){ ... } |
| D | x( inth,intm,int n){ ... } |

|  |  |
| --- | --- |
| 序号 | 38 |
| 题干 | 下列对Java中使用调试错误的说法是（ ）。 |
| 题型 | 单选题 |
| A | F5键用于跳入某个特定的方法 |
| B | F6键用于单步跳过程序代码 |
| C | 双击代码编辑区左侧设置断点 |
| D | 使用程序调试的方法为观察变量—>设置断点—>单步运行 |

|  |  |
| --- | --- |
| 序号 | 39 |
| 题干 | 在Java中，以下代码的输出结果是（ ）。  public static void add(int num){  num += 5;  }  public static void main(String[] args) {  int age = 20;  add(age);  System.out.println(age);  } |
| 题型 | 单选题 |
| A | 20 |
| B | 25 |
| C | 5 |
| D | 编译错误 |
| 序号 | 40 |
| 题干 | 在Java中，类Person中的方法printA()定义如下，以下语句输出为（）。  public void printA() {  int a =10;  int result=10%3;  System.out.println(result);  }  类Chinese继承了Person，类Chinese里面重写了方法printA(),其定义如下：  public void printA() {  int a=10;  System.out.println(a/3);  }  Demo类中有如下代码：  Person p=new Chinese();  p.printA(); |
| 题型 | 单选题 |
| A | 0 |
| B | 3.3333 |
| C | 1 |
| D | 3 |

|  |  |
| --- | --- |
| 序号 | 41 |
| 题干 | 对以下Java代码的描述正确的是（ ）。  public class Test {  private int count=10;  public static void main(String[] args) {  for(int i=0;i<count;i++){  System.out.println(i);  }  }  } |
| 题型 | 单选题 |
| A | 程序会执行10次，分别输出0到9 |
| B | 程序会执行11次，分别输出0到10 |
| C | 程序可能会出现死循环 |
| D | 程序编译错误 |
| 序号 | 42 |
| 题干 | 分析下面的Java代码，编译运行结果是（ ）。  public class A {  public int getNumber(int a) {  return a + 1;  }  }  public class B extends A {  public int getNumber(int a) { //第7行  return a + 2;  }  public static void main(String args[]) {  A a = new B(); //第11行  System.out.println(a.getNumber(0));  }  } |
| 题型 | 单选题 |
| A | 输出1 |
| B | 输出2 |
| C | 第7行导致编译错误 |
| D | 第11行导致编译错误 |
| 序号 | 43 |
| 题干 | 如下Java代码，程序运行结果是（ ）。  class Super {  public Super (){  System.out.println("Super ()");  }  public Super (String str){  System.out.println("Super (String str)");  }  }  class Sup extends Super {  public Sup (){  super ("");  System.out.println("Sup ()");  }  public Sup (String str){  System.out.println("Sup (String str) ");  }  }  public class Test {  public static void main (String[] args) {  Sup sup = new Sup ();  }  } |
| 题型 | 单选题 |
| A | Sup () |
| B | Super ()  Sup () |
| C | Super (String str)  Sup () |
| D | 编译错误 |

|  |  |
| --- | --- |
| 序号 | 44 |
| 题干 | 在Java中，以下代码段输出的结果如图所示，横线处的代码应是（ ）。  public class Test {  public static void main(String[] args) {  for(\_\_\_\_\_\_\_\_\_\_\_\_\_\_){  for(int k=0;k<=2-Math.abs(i);k++){  System.out.print(" ");  }  for(int j=0;j<Math.abs(i)\*2+1;j++){  System.out.print("\*");  }  System.out.println();  }  }  } |
| 题型 | 单选题 |
| 图片 | http://tiku.kgc.cn/resources/questionImages/20140703090851_0.png |
| A | int i=0;i<5;i++ |
| B | int i=4;i>=0;i-- |
| C | int i=-2;i>=2;i++ |
| D | int i=2;i>=-2;i=i-1 |

|  |  |
| --- | --- |
| 序号 | 45 |
| 题干 | 在Java中，异常处理正确的语法是（ ）。 |
| 题型 | 单选题 |
| A | try{}catch(...){} |
| B | try{} |
| C | catch(...){} |
| D | catch(...){}finally{} |

|  |  |
| --- | --- |
| 序号 | 46 |
| 题干 | 下面程序中类ClassDemo中定义了一个静态变量sum，分析程序段的输出结果：（）。  class ClassDemo {  public static int sum = 1;  public ClassDemo() {  sum = sum + 5;  }  }  public class ClassDemoTest {  public static void main(String args[]) {  ClassDemo demo1 = new ClassDemo();  ClassDemo demo2 = new ClassDemo();  System.out.println(demo1.sum);  }  } |
| 题型 | 单选题 |
| A | 0 |
| B | 6 |
| C | 11 |
| D | 2 |

|  |  |
| --- | --- |
| 序号 | 47 |
| 题干 | 阅读下列Java代码，运行后正确的输出结果是（ ）。  public class Test {  public int coutStr(String[] arry,String str){  int count=0;  for(int i=0;i<arry.length;i++){  if(arry[i].indexOf(str)!=-1)  count++;  }  return count;  }  public static void main(String[] args) {  String[]arry={"happy","apple","money","ending"};  String findStr="a";  Test test=new Test();  int count=test.coutStr(arry,findStr);  System.out.println("此Java的数组中有"+count  +"个字符串包含字符"+findStr);  }  } |
| 题型 | 单选题 |
| A | 此Java的数组中有1个字符串包含字符a |
| B | 此Java的数组中有2个字符串包含字符a |
| C | 编译错误 |
| D | 正常编译，运行时报错 |

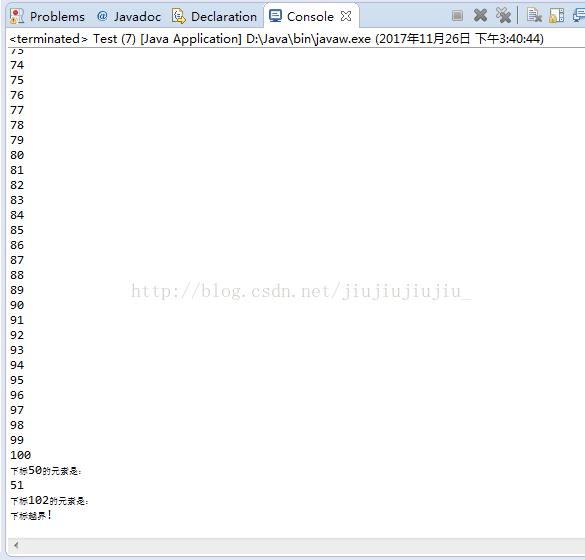
|  |  |
| --- | --- |
| 序号 | 48 |
| 题干 | 下面Java程序的运行结果是（ ）。  int[]arr=new int[4]{0,1,2,3};  for(int i=arr[0];i<=arr[3];i++){  System.out.print(arr[i]);  } |
| 题型 | 单选题 |
| A | 0123 |
| B | 索引越界 |
| C | 1234 |
| D | 编译出错 |
| 序号 | 49 |
| 题干 | 以下Java代码的输出结果为（ ）。  public abstract class Person {  public abstract void study();  }  class Chinese extends Person {  public void study(){  System.out.println("学汉语");  }  }  public class Test {  public static void main(String[] args) {  Person p1=new Chinese();  System.out.println(p1 instanceof Person);  System.out.println(p1 instanceof Chinese);  System.out.println(p1 instanceof Object);  }  } |
| 题型 | 单选题 |
| A | true、true、false |
| B | true、false、false |
| C | true、false、true |
| D | true、true、true |

|  |  |
| --- | --- |
| 序号 | 50 |
| 题干 | 在Java中，如下代码段的输出结果为（）。  public class Test {  int i = 10;  public void methodOne(int i){  System.out.println(i);  }  public void methodTwo(){  int i = 12;  System.out.println(i);  i++;  }  public void methodThree(){  System.out.println(i);  }  public static void main(String args[]){  Test test = new Test();  test.methodOne(11);  test.methodTwo();  test.methodThree();  }  } |
| 题型 | 单选题 |
| A | 10  11  12 |
| B | 11  12  10 |
| C | 不能正常运行，编译时出错 |
| D | 以上都不正确 |

# 上机题部分

## 一．List集合

使用ArrayList集合，对其添加100个不同的元素：  
 1.使用add()方法将元素添加到ArrayList集合对象中；  
 2.调用集合的iterator()方法获得Iterator对象，并调用Iterator的hasNext()和next()方法，迭代的读取集合中的每个元素；  
 3.调用get()方法先后读取索引位置为50和102的元素，要求使用try-catch结构处理下标越界异常；



## 二．日期

编写一个工具类DateUtil,该类有两个方法，分别实现如下功能：(5分)

1）用来实现字符串转换成Date

2）Date转换成字符串。

要求：面向对象方式实现，其他方式不得分

* 实现日期格式的字符串转换为java.util.Date类型。例如：“2017-10-20”转换为日期格式；
* 实现java.util.Date类型的值转换为字符串格式；

## 三．面向对象

以面向对象的思想来实现下面的需求

已知三个类分别是人类,老师类,学生类,三个类中的成员如下

1.人类:

属性:

姓名,年龄

行为:

吃饭方法(方法中打印:人都要吃饭)

2.老师类:是人类的一种

属性:

含有和人相同的属性:姓名,年龄

行为:

教课方法(方法中打印:"33岁的老王在教java",

注意:教课方法中打印的年龄和姓名需要获取,不能直接打印这句话:"33岁的老王在教java")

3.学生类:是人类的一种

属性:

含有和人相同的属性:姓名,年龄

行为:

学习方法(方法中打印:"18岁的小王在努力敲代码",

注意:学习方法中打印的年龄和姓名需要获取,不能直接打印这句话:"18岁的小王在努力敲代码")

要求如下:

a.按照上面的描述定义出人类,老师类,学生类(4分)

b.利用继承来简化代码(2分)

c.所有的属性必须用私有修饰,利用Eclipse生成每个属性对应的get和set方法(4分)

d.利用Eclipse生成每个类的空参构造函数

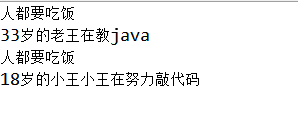
利用Eclipse生成每个类的含有姓名,年龄的满参构造函数

在老师类和学生类的满参构造方法中必须用上super语句(5分)

e.定义一个测试类名称为Demo02,在测试类中(5分)

利用老师类的满参构造函数,创建一个老师对象,这个老师的姓名为老王,年龄为33,调用吃饭方法和教课方法

利用学生类的满参构造函数,创建一个学生对象,这个学生的姓名为小王,年龄为18,调用吃饭方法和学习方法

最终控制台打印结果  


## 四.io流

根据需求完成以下功能

a.已知在当前工程的根目录下有一个a.txt,a.txt中的内容为:(可以在a.txt中手动输入以下内容)

I love you

I miss you

I hate you

I want you

1.利用 字符数组,FileReader,FileWriter将a.txt拷贝到当前工程下的b.txt中,异常不需要处理,抛出即可(throws)

2.利用 BufferedReader和BufferedWriter将a.txt拷贝到当前工程下的c.txt中,异常不需要处理,抛出即可(throws)

## 5、集合

需求说明：

已知用户购买的商品名称如下:

铅笔

本子

橡皮擦

铅笔

铅笔

本子

本子

本子

a.请使用一个集合(使用泛型)存储用户购买的上述商品名称

b.利用迭代器方式遍历出上述集合

c.利用HashMap统计出每件商品对应的购买数量

d.利用增强for循环遍历上述的HashMap

|  |
| --- |
| 商品名称 数量  铅笔 3  本子 4  橡皮擦 1 |