

一. 简答题

- 1
- 2
- 3
- 6

Java语言实习4可重用类

题量: 6 满分: 100

作答时间: 2023-09-25 11:35 至 2023-10-07 12:35

批阅人	***
打分	**

一. 简答题 (共6题)

1. (简答题)

每题必须附上完整代码截图和结果截图，并按要求进行，否则0分。

1. String，StringBuffer 类的使用方法

(1) 利用下面的关键代码编写一个完整的程序，理解 String 类的使用。

String s=new String("This is an demo of the String method.");

```
//String s="This is an demo of the String method.";
System.out.println("Length: "+s.length());
System.out.println("SubString: "+s.substring(11,15));
```

(2) 创建一个程序，实现如下功能

- a) 将字符成“1234”与“5423”转换的整数，并求和，并将结果输出。
- b) 将字符成“87.1”与“45.18”转换的浮点数，并求乘积，并将结果输出。

(3) 利用下面的关键代码编写一个完整的程序，理解 StringBuffer 类的使用。

```
StringBuffer sb=new StringBuffer("Hello World!");
sb.append(" Hello Java!");
sb.insert(12," And");
System.out.println(sb);
System.out.println(sb.charAt(0));
sb.setCharAt(0,"h");
System.out.println(sb.charAt(0));
System.out.println(sb);
```

本次实习要求（每小题5分，3\*5' 共15分，如未按要求进行，0分）：

- (1) 被注释的String s改为s1，去掉注释；第4行改为截取s1的“method”，用字符串的长度方法设置截取参数。
- (2) 必须有数据格式转换。
- (3) 分析代码运行过程，了解StringBuffer特性。

我的答案：

(1) 利用下面的关键代码编写一个完整的程序，理解String类的使用。

代码行：

```
package stringwork;

public class stringwork1 {
    public static void main(String[] args) {
        String s = new String("This is an demo of the String method."); //String s= "This is an demo of the
String method.";
        System.out.println("Length: " + s.length());
```

作业详情

```
}
package stringwork;

public class stringwork1 {
    public static void main(String[] args) {
        String s = new String("This is an demo of the String method."); //String s = "This is an demo of the String method.";
        System.out.println("Length: " + s.length());
        System.out.println("SubString: " + s.substring(11, 15));
    }
}
```

实验结果:

Length: 37  
SubString: demo

被注释的String s改为s1，去掉注释；第4行改为截取s1的"method"，用字符串的长度方法设置截取参数。

```
package stringwork;

public class stringwork1 {
    public static void main(String[] args) {
        String s1 = new String("This is an demo of the String method.");
        System.out.println("Length: " + s1.length());
        System.out.println("SubString: " + s1.substring(11)); // 截取从索引11到末尾的子字符串，11为自己设定数值
    }
}
<terminated> stringwork1 java Application C:\Prog
Length: 37
SubString: demo of the String method.
```

(2)创建一个程序，实现如下功能

- a) 将字符成"1234"与"5423"转换的整数，并求和，并将结果输出。
- b) 将字符成"87.1"与"45.18"转换的浮点数，并求乘积，并将结果输出。

代码行: (已包含数据格式转换)

```
package stringwork;

public class stringwork1to2 {
    public static void main(String[] args) {
        String num1 = "1234";
        String num2 = "5423";
        int intNum1 = Integer.parseInt(num1);
        int intNum2 = Integer.parseInt(num2);
        int sum = intNum1 + intNum2;
        System.out.println("整数和: " + sum); // 将字符串转换为整数并求和
        String float1 = "87.1";
        String float2 = "45.18";
        double doubleNum1 = Double.parseDouble(float1);
        double doubleNum2 = Double.parseDouble(float2);
        double product = doubleNum1 * doubleNum2;
        System.out.println("浮点数积: " + product); // 将字符串转换为浮点数并求乘积
    }
}
```

实验结果:

整数和: 6657  
浮点数积: 3935.178

(3) 利用下面的关键代码编写一个完整的程序，理解StringBuffer 类的使用。

代码行:

```
package stringwork;

public class stringwork1to3 {
    public static void main(String[] args) {
        StringBuffer sb = new StringBuffer("Hello World!");
        sb.append(" Hello Java!"); // 追加字符串
        sb.insert(12, " And"); // 在指定位置插入字符串
        System.out.println(sb); // 打印整个字符串
        System.out.println(sb.charAt(0)); // 获取并打印为0的字符
        sb.setCharAt(0, 'h'); // 修改索引为0的字符为小写'h'
        System.out.println(sb); // 再次打印修改后的字符串
    }
}
```

实验结果:

Hello World! And Hello Java!  
H  
hello World! And Hello Java!

分析代码运行过程，了解StringBuffer特性。

过程: 首先创建StringBuffer对象，在内存中创建了一个字符串对象sb，并初始化为"Hello World!";  
其次追加字符串,使用append方法，将" Hello Java!"追加到字符串的末尾;  
然后插入字符串使用insert方法，在索引12的位置插入" And"。  
接着打印整个字符串，使用"println"方法打印字符串。  
获取并打印索引为0的字符，使用charAt方法，获取索引0处的字符。

一. 简答题

1

2

3

6

作业详情

- 特性：1.提供许多方法来修改字符串的内容，如append、insert、setCharAt等。
- 2.内容可以在同一个对象上多次修改
- 3.对象能够被多次的修改，并且不产生新的未使用对象

一. 简答题

1

2

3

6

\*\*\*: 13分

第一问使用字符串长度方法截取参数错误。

2. (简答题)

每题必须附上完整代码截图和结果截图，并按要求进行，否则0分。

(1) 利用下面的关键代码编写一个完整的程序，理解 Math 类的使用。

```
System.out.println (Math.abs (-5.8)); //5.8
System.out.println (Math.ceil (3.2)); //4
System.out.println (Math.floor (3.8)) //3
System.out.println (Math.round (3.8)); //4
System.out.println (Math.round (3.2)); //3
System.out.println (Math.min (3,2)); //2
System.out.println (Math.max (Math.PI,4)); //4
System.out.println (Math.log (7.0)); //1.94591
System.out.println (Math.pow (7,2)); //72 ---- 49
System.out.println (Math.exp (0.4)); //1.49183
System.out.println (Math.IEEEremainder(10.0,3.0)); //返回 1
angle = 0.785398; //以弧度为单位的角， π /4
System.out.println (Math.tan (angle)); //返回该角的正切
System.out.println (Math.asin(0.707107)); //返回反余弦
System.out.println ("e is:" + Math.e); // e is: 2.71828
System.out.println (" π is:" + Math.PI); // π is: 3.14159
System.out.println(Math.random()); //产生 0 和 1（不含 1）之间的伪随机数
```

- (2) 利用 Math 类中的一些函数编写一个程序实现以下功能：
- 输出半径为 6.5 的圆的面积
  - 输出 1/3PI 的正弦、余弦的值
  - 输出不大于 6.45 的整数

要求：（1）本题考察了Math的多少种方法和多少属性值？分别是什么？  
（2）根据原题进行，注意应为“输出半径为6.5的圆的面积”，输出面积应有单位m<sup>2</sup>。特定参数需用Math属性值，不能自定义变量。

我的答案:

(1)利用下面的关键代码编写一个完整的程序，理解Math类的使用。  
代码行

作业详情

```
public class Stringwork2 {
    public static void main(String[] args) {
        System.out.println(Math.abs(-5.8)); // 输出绝对值: 5.8
        System.out.println(Math.ceil(3.2)); // 输出向上取整值: 4.0
        System.out.println(Math.floor(3.8)); // 输出向下取整值: 3.0
        System.out.println(Math.round(3.8)); // 输出四舍五入值: 4
        System.out.println(Math.round(3.2)); // 输出四舍五入值: 3
        System.out.println(Math.min(3, 2)); // 取两个数中的较小值: 2
        System.out.println(Math.max(Math.PI, 4)); // 取两个数中的较大值: 4.0
        System.out.println(Math.log(7.0)); // 输出自然对数: 1.9459101490553132
        System.out.println(Math.pow(7, 2)); // 输出幂运算: 49.0
        System.out.println(Math.exp(0.4)); // 输出指数函数: 1.4918246976412703
        System.out.println(Math.IEEEremainder(10.0, 3.0)); // 返回余数: 1.0
        double angle = 0.785398; // 以弧度为单位的角,  $\pi/4$ 
        System.out.println(Math.tan(angle)); // 输出正切函数: 0.9999999996829318
        System.out.println(Math.asin(0.707107)); // 输出反正弦函数: 0.785398163338051
        //
        System.out.println("e is: " + Math.E); // 输出常数e: 2.718281828459045
        System.out.println("n is: " + Math.PI); // 输出常数 $\pi$ : 3.141592653589793
        System.out.println(Math.random()); // 输出0.0 到 1.0 之间的随机数
    }
}
```

实验结果

```
terminated> Stringwork2.java App
5.8
4.0
3.0
4
3
2
4.0
1.9459101490553132
49.0
1.4918246976412703
1.0
0.9999999996829318
0.785398163338051
e is: 2.718281828459045
n is: 3.141592653589793
0.6191498643096327
```

本题考察了Math的多少种方法和多少属性值？分别是什么？

方法：1.Math.abs(double a) 返回给定数字的绝对值。

2.Math.ceil(double a) 返回不小于参数的最小整数。

3.Math.floor(double a)返回不大于参数的最大整数。

4.Math.round(double a)返回最接近参数的整数，四舍五入。

5.Math.min(double a, double b)返回两个参数中的较小值。

6.Math.max(double a, double b)返回两个参数中的较大值。

7.Math.log(double a)返回给定参数的自然对数。

8.Math.pow(double a, double b)返回a的b次幂。

9.Math.exp(double a)返回自然对数的底e的a次幂。

10.Math.IEEEremainder(double f1, double f2)返回浮点数f1/f2的IEEE余数。

11.Math.tan(double a)返回给定角度的正切值。

12.Math.asin(double a)返回给定值的反正弦值。

属性值：1.Math.E表示自然对数的底，约为2.71828。

2.Math.PI表示圆周率 $\pi$ 的值，约为3.14159。

(2)利用Math类中的一些函数编写一个程序实现以下功能:

输出半径为6.5的圆的面积

输出1/3 $\pi$  的正弦、余弦的值

输出不大于6.45的整数

代码行:

```
package stringwork;

public class stringwork2 {
    public static void main(String[] args) {

        double radius = 6.5;
        double area = Math.PI * Math.pow(radius, 2);
        System.out.println("圆的面积: " + area + " $\pi^2$ "); // 计算半径为6.5的圆的面积

        double angle = Math.PI / 3.0;
        double sin1 = Math.sin(angle);
        double cos1 = Math.cos(angle);
        System.out.println("1/3 $\pi$  的正弦值: " + sin1);
        System.out.println("1/3 $\pi$  的余弦值: " + cos1); // 计算1/3 $\pi$ 的正弦和余弦

        double max = 6.45;
        int putnum = (int) Math.floor(max);
        System.out.println("不大于6.45的整数: " + putnum); // 输出不大于6.45的整数
    }
}
```

实验结果:

一. 简答题

1

2

3

6

作业详情

1/3π 的正弦值：0.8660254037844386  
1/3π 的余弦值：0.5000000000000001  
不大于6.45的整数：6

根据原题进行，注意应为“输出半径为6.5的圆的面积”，输出面积应有单位m²。特定参数需用Math属性值，不能自定义变量。

一. 简答题

1

2

3

6

\*\*\*: 13.3分

Math方法少写了一种

3. (简答题)

每题必须附上完整代码截图和结果截图，并按要求进行，否则0分。

编写一个排序的程序，实现以下功能：

- (1) 从键盘读入 10 个整数存入数组；
- (2) 找到数组最大值；
- (3) 将数组按从大到小顺序排列；
- (4) 输出最大值和排序后结果。

我的答案:

```
package stringwork;
import java.util.Scanner;
import java.util.Arrays;

public class stringwork3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] numbers = new int[10];
        System.out.println("请输入10个整数: ");
        for (int i = 0; i < 10; i++) {
            numbers[i] = scanner.nextInt();
        } // 从键盘读入10个整数存入数组

        int max = numbers[0];
        for (int i = 1; i < 10; i++) {
            if (numbers[i] > max) {
                max = numbers[i];
            }
        } // 找到数组最大值

        Arrays.sort(numbers);
        int[] sortedNumbers = new int[10];
        for (int i = 0; i < 10; i++) {
            sortedNumbers[i] = numbers[9 - i];
        } // 将数组按从大到小顺序排列

        System.out.println("最大值: " + max);
        System.out.println("排序后的数组: ");
        for (int i = 0; i < 10; i++) {
            System.out.print(sortedNumbers[i] + " ");
        } // 输出最大值和排序后结果

        scanner.close();
    }
}
```

实验结果

作业详情

```
2|
3|
4|
5|
6|
9|
8|
0|
7|
最大值：9
排序后的数组：
9 8 7 6 5 4 3 2 1 0
```

一. 简答题

- 1
- 2
- 3
- 6

\*\*\*: 16.6分  
没有错误

4. (简答题)

每题必须附上完整代码截图和结果截图，并按要求进行，否则0分。

在系统盘根目录（如C盘、D盘等）下建立一个文本cypercube.txt，输入以下信息：

立方体的长、宽、高分别为：  
15.5 80.5 20  
圆的半径为：  
40.333

要求：

- (1) 从文本中读取第一行信息，了解几何形状为长方体，长、宽、高存储顺序并显示提示信息；读取第二行的长、宽、高值并分别说明长、宽、高的信息并显示；读取第三行信息，了解几何形状为圆，并显示提示信息，读取第四行得到圆的半径。计算立方体体积和圆周长，输出到屏幕，包括单位。注如涉及到相应数值，必须使用Math属性。
- (2) 新建同目录下的output.txt，将体积和面积保存至output.txt，需有必要说明。
- (3) 用户指定长、宽、高和半径，将新体积和新面积保存至output.txt，需另起一行。

我的答案：

(1) 代码截图

```
package stringwork;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class stringwork4 {
    public static void main(String[] args) {
        try {
            BufferedReader reader = new BufferedReader(new FileReader("cypercube.txt")); // 从文本文件读取信息
            String shapeInfo = reader.readLine();
            System.out.println(shapeInfo); // 读取第一行信息

            String dimensionsInfo = reader.readLine();
            String[] dimensions = dimensionsInfo.split("\\s+");
            double length = Double.parseDouble(dimensions[0]);
            double width = Double.parseDouble(dimensions[1]);
            double height = Double.parseDouble(dimensions[2]);
            System.out.println("长: " + length + " 宽: " + width + " 高: " + height); // 读取第二行信息，解析长、宽、高

            String circleInfo = reader.readLine();
            System.out.println(circleInfo); // 读取第三行信息

            String radiusInfo = reader.readLine();
            double radius = Double.parseDouble(radiusInfo); // 读取第四行信息，得到圆的半径

            double cubeVolume = length * width * height;
            double circlePerimeter = 2 * Math.PI * radius; // 计算立方体的体积和圆的周长

            System.out.println("立方体体积: " + cubeVolume + " 立方米");
            System.out.println("圆的周长: " + circlePerimeter + " 米"); // 输出结果，包括单位

            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

实验结果





每题必须附上完整代码截图和结果截图，并按要求进行，否则0分。

自行制作一个 20M 左右的 test.txt 英文文本，直接读取、使用缓冲流读取、自定义缓冲块大小来读取文本。

- (1) 统计文本中英文字符 “b” 出现的次数；
- (2) 计算各方法运行时间并进行比较；
- (3) 调整缓冲块大小，得到最快运行速度的缓冲块。

本题必须附上test.txt部分内容截图和大小信息截图。

一. 简答题

1

2

3

6

我的答案:

 large\_text\_file.txt

2023/10/6 3:05

文本文档

20,480 KB

large\_text\_file.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

jumps fox lazy lazy dog. lazy jumps The brown fox dog. jumps The dog.  
the dog. brown quick quick quick The  
quick jumps fox quick lazy dog. jumps  
fox over jumps The brown jumps over The lazy the the jumps  
The fox brown The lazy jumps over lazy quick over dog. quick  
The dog. fox dog. brown fox brown dog. brown jumps  
quick The dog. quick dog. lazy dog. the over fox  
The The over The The over  
the the the The The lazy brown fox lazy The over jumps  
brown dog. fox brown The jumps over the brown the  
fox the over quick brown quick over the brown jumps brown lazy  
dog. fox The quick quick fox jumps brown brown over the brown jumps  
The the The over brown the brown dog. The dog. jumps lazy jumps  
brown the lazy The over jumps dog. dog. quick dog. jumps brown  
jumps over over jumps dog.  
dog. jumps quick quick fox dog. jumps  
jumps The quick dog. dog. fox The dog. brown dog. the jumps  
jumps brown quick dog. the The The brown lazy lazy fox  
quick jumps over dog. The lazy brown  
quick jumps quick fox the fox dog. The brown over The jumps  
dog. jumps dog. over lazy lazy fox over brown over The  
brown the the quick dog. brown jumps  
quick The lazy over brown quick lazy  
fox brown jumps jumps quick lazy lazy  
dog. lazy lazy dog. quick over fox  
fox dog. lazy lazy quick fox brown brown  
quick lazy quick over quick lazy over  
The The brown the fox the quick over the over lazy  
fox jumps quick lazy jumps The fox The The  
the jumps over lazy jumps fox brown the the the jumps dog.  
代码行:



作业详情

```
public class StringWork6 {
    public static void main(String[] args) {
        String fileName = "text.txt";

        // 统计字符'b'出现的次数并计算运行时间
        long startTime, endTime;

        // 1. 直接读取文本文件
        startTime = System.currentTimeMillis();
        int count1 = countCharacterDirectly(fileName, 'b');
        endTime = System.currentTimeMillis();
        long time1 = endTime - startTime;

        // 2. 使用缓冲流读取文本文件
        startTime = System.currentTimeMillis();
        int count2 = countCharacterWithBufferedReader(fileName, 'b');
        endTime = System.currentTimeMillis();
        long time2 = endTime - startTime;

        // 输出结果
        System.out.println("直接读取字符'b'出现次数: " + count1 + ", 运行时间: " + time1 + " 毫秒");
        System.out.println("使用缓冲流读取字符'b'出现次数: " + count2 + ", 运行时间: " + time2 + " 毫秒");

        // 3. 调整缓冲块大小, 找到最快速度
        int[] bufferSizes = {1024, 2048, 4096, 8192}; // 不同的缓冲块大小
        long[] times = new long[bufferSizes.length];
        int[] counts = new int[bufferSizes.length];

        for (int i = 0; i < bufferSizes.length; i++) {
            startTime = System.currentTimeMillis();
            counts[i] = countCharacterWithCustomBufferSize(fileName, 'b', bufferSizes[i]);
            endTime = System.currentTimeMillis();
            times[i] = endTime - startTime;
        }

        // 找到最快速度的缓冲块大小
        int fastestIndex = 0;
        long fastestTime = times[0];
        for (int i = 1; i < times.length; i++) {
            if (times[i] < fastestTime) {
                fastestTime = times[i];
                fastestIndex = i;
            }
        }

        System.out.println("最快速度的缓冲块大小: " + bufferSizes[fastestIndex]);
        System.out.println("使用缓冲块大小 " + bufferSizes[fastestIndex] + " 读取字符'b'出现次数: " + counts[fastestIndex] + ", 运行时间: " + fastestTime + " 毫秒");
    }

    // 直接读取文本文件并统计字符出现次数
    private static int countCharacterDirectly(String fileName, char targetChar) {
        int count = 0;
        try {
            FileReader reader = new FileReader(fileName);
            int c;
            while ((c = reader.read()) != -1) {
                if (c == targetChar) {
                    count++;
                }
            }
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return count;
    }

    // 使用缓冲流读取文本文件并统计字符出现次数
    private static int countCharacterWithBufferedReader(String fileName, char targetChar) {
        int count = 0;
        try {
            BufferedReader reader = new BufferedReader(new FileReader(fileName));
            int c;
            while ((c = reader.read()) != -1) {
                if (c == targetChar) {
                    count++;
                }
            }
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return count;
    }

    // 使用自定义缓冲块大小读取文本文件并统计字符出现次数
    private static int countCharacterWithCustomBufferSize(String fileName, char targetChar, int bufferSize) {
        int count = 0;
        try {
            BufferedReader reader = new BufferedReader(new FileReader(fileName), bufferSize);
            int c;
            while ((c = reader.read()) != -1) {
                if (c == targetChar) {
                    count++;
                }
            }
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return count;
    }
}
```

结果:

直接读取字符'b'出现次数: 455223, 运行时间: 374 毫秒  
使用缓冲流读取字符'b'出现次数: 455223, 运行时间: 139 毫秒  
最快速度的缓冲块大小: 8192  
使用缓冲块大小 8192 读取字符'b'出现次数: 455223, 运行时间: 133 毫秒

一. 简答题

1

2

3

6

\*\*\*: 16.6分

没有问题

作业详情

StringBuffer 类的 toUpperCase() 方法将文件所有字符变为大写, 换行后添加至原文件, 记录程序执行的时间。  
应有文本部分内容截图和大小信息截图。

一. 简答题

1

2

3

6

我的答案:

english.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

It is a truth universally acknowledged, that a single man in possession of a good fortune must be in want of a wife.

However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed

"My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield park is let at last?"

Mr. Bennet replied that he had not.

"But it is," returned she; "for Mrs. Long has just been here, and she told me all about it."

Mr. Bennet made no answer.

"Do not you want to know who has taken it?" cried his wife impatiently.

"You want to tell me, and I have no objection to hearing it."

This was invitation enough.

"Why, my dear, you must know, Mrs. Long says that Netherfield is taken by a young man of large fortune from the north of Eng

"What is his name?"

"Bingley."

"Is he married or single?"

"Oh! single, my dear, to be sure! A single man of large fortune; four or five thousand a year. What a fine thing for our girls!"

cybercube.txt	2023/10/6 2:10	文本文件	1 KB
english.txt	2023/10/6 2:53	文本文档	15 KB

代码行:

```
package stringwork;  
import java.io.*;  
import java.util.Date;  
  
public class stringwork5 {  
    public static void main(String[] args) {  
        try {  
  
            long startTime = System.currentTimeMillis();// 记录程序开始时间  
  
            File originalFile = new File("english.txt");  
            File tempFile = new File("temp.txt"); // 打开原文件和临时文件  
  
            BufferedReader reader = new BufferedReader(new FileReader(originalFile));  
            BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile));  
  
            String line;  
            while ((line = reader.readLine()) != null) {  
  
                StringBuffer stringBuffer = new StringBuffer(line);  
                stringBuffer = stringBuffer.append("\n");  
                writer.write(stringBuffer.toString().toUpperCase());// 使用StringBuffer的toUpperCase()  
            }  
  
            reader.close();  
            writer.close(); // 关闭文件流  
  
            originalFile.delete();  
            tempFile.renameTo(originalFile); // 删除原文件并将临时文件重命名为原文件名  
  
            long endTime = System.currentTimeMillis();  
            long executionTime = endTime - startTime;// 计算并输出程序执行时间  
            System.out.println("程序执行时间: " + executionTime + " 毫秒");  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

实验结果:

Console  
<terminated> stringwork5 [Java  
程序执行时间: 6 毫秒

作业详情

HOWEVER LITTLE KNOWN THE FEELINGS OR VIEWS OF SUCH A MAN MAY BE ON HIS FIRST ENTERING A NEIGHBOURHOOD, THIS TRUTH IS SO WELL FIXED IN THE MIND OF MR. BENNET THAT HE HAD NO DOUBT OF IT. "MY DEAR MR. BENNET," SAID HIS LADY TO HIM ONE DAY, "HAVE YOU HEARD THAT NETHERFIELD PARK IS LET AT LAST?" MR. BENNET REPLIED THAT HE HAD NOT. "BUT IT IS," RETURNED SHE; "FOR MRS. LONG HAS JUST BEEN HERE, AND SHE TOLD ME ALL ABOUT IT." MR. BENNET MADE NO ANSWER. "DO NOT YOU WANT TO KNOW WHO HAS TAKEN IT?" CRIED HIS WIFE IMPATIENTLY. "YOU WANT TO TELL ME, AND I HAVE NO OBJECTION TO HEARING IT." THIS WAS INVITATION ENOUGH. "WHY, MY DEAR, YOU MUST KNOW, MRS. LONG SAYS THAT NETHERFIELD IS TAKEN BY A YOUNG MAN OF LARGE FORTUNE FROM THE NORTH OF ENGLAND; THAT HE "WHAT IS HIS NAME?" "BINGLEY." "IS HE MARRIED OR SINGLE?" "OH! SINGLE, MY DEAR, TO BE SURE! A SINGLE MAN OF LARGE FORTUNE; FOUR OR FIVE THOUSAND A YEAR. WHAT A FINE THING FOR OUR GIRLS!" "HOW SO? HOW CAN IT AFFECT THEM?" "MY DEAR MR. BENNET," REPLIED HIS WIFE, "HOW CAN YOU BE SO TIRESOME! YOU MUST KNOW THAT I AM THINKING OF HIS MARRYING ONE OF THEM." "IS THAT HIS DESIGN IN SETTLING HERE?"

一. 简答题

- 1
- 2
- 3
- 6

\*\*\*: 13分

转换后并不是换行添加至原文件保存而是到新文件中存放。