Java面向对象程序设计课程设计实验报告四

实验人：王维

学号：320210942931

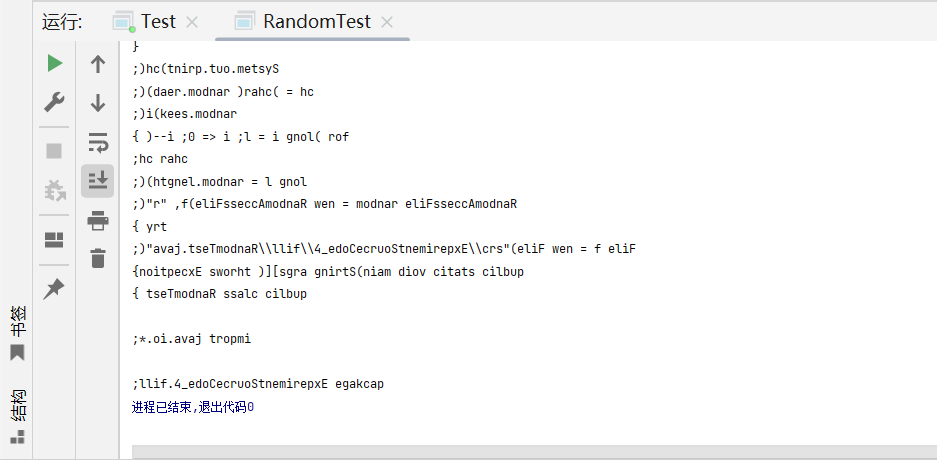
班级：2021级计算机科学与技术二班

一：补全程序题

* 7.3.2（2）
* 源代码

|  |
| --- |
| package ExperimentSourceCode\_4**.**fill**;**  **import** java**.**io**.\*;**  public class RandomTest **{**  public static void main**(**String args**[])** **throws** Exception**{**  File f **=** **new** File**(**"src\\ExperimentSourceCode\_4\\fill\\RandomTest.java"**);**//加上路径，不然会读取失败  **try** **{**  RandomAccessFile random **=** **new** RandomAccessFile**(**f**,** "r"**);** // 创建指向文件f的random对象  long l **=** random**.**length**();**  char ch**;**  **for** **(**long i **=** l**;** i **>=** 0**;** i**--)** **{**  random**.**seek**(**i**);**  ch **=** **(**char**)** random**.**read**();**  System**.**out**.**print**(**ch**);**  **}**  random**.**close**();**  **}** **catch** **(**Exception e**)** **{**  System**.**out**.**println**(**"IOError!"**);**  **}**  **}**  **}** |

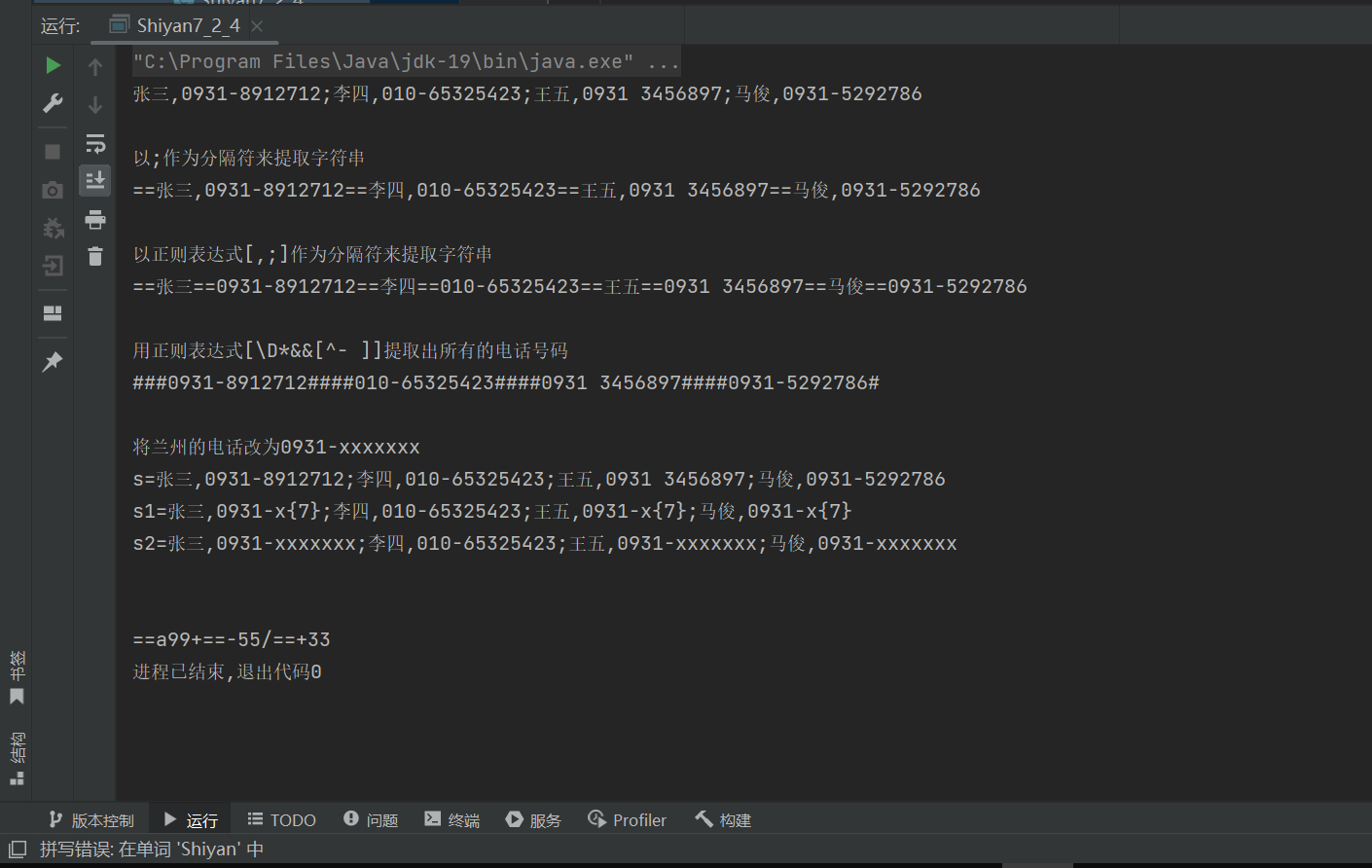
* 运行结果



* 7.3.2（4）
* 源代码：

|  |
| --- |
| package ExperimentSourceCode\_4**.**fill**;**  **import** java**.**io**.\*;**  public class Shiyan7\_2\_4 **{**  public static void main**(**String**[]** args**)** **throws** Exception **{**  FileReader fin**=new** FileReader**(**"src\\ExperimentSourceCode\_4\\FileData\\phone.txt"**);**  BufferedReader bin**=new** BufferedReader**(**fin**);**  String s**=**bin**.**readLine**();**  System**.**out**.**println**(**s**);**  System**.**out**.**print**(**"\n以;作为分隔符来提取字符串\n"**);**  String**[]** sa**=**s**.**split**(**";"**);**  **for(**String ss**:**sa**)** System**.**out**.**print**(**"=="**+**ss**);**//用==来区分数组元素  System**.**out**.**print**(**"\n\n以正则表达式[,;]作为分隔符来提取字符串\n"**);**  sa**=**s**.**split**(**"[,;]"**);**  **for(**String ss**:**sa**)** System**.**out**.**print**(**"=="**+**ss**);**//用==来区分数组元素    System**.**out**.**print**(**"\n\n用正则表达式[\\D\*&&[^- ]]提取出所有的电话号码\n"**);**  sa**=**s**.**split**(**"[\\D\*&&[^- ]]"**);**  **for(**String ss**:**sa**)** System**.**out**.**print**(**ss**+**"#"**);**//用#来区分数组元素    System**.**out**.**println**(**"\n\n将兰州的电话改为0931-xxxxxxx"**);**  String s1**=**s**.**replaceAll**(**"0931[ -]\\d{7}"**,**"0931-x{7}"**);**  String s2**=**s**.**replaceAll**(**"0931[ -]\\d{7}"**,**"0931-xxxxxxx"**);**  System**.**out**.**println**(**"s="**+**s**);**  System**.**out**.**println**(**"s1="**+**s1**);**  System**.**out**.**println**(**"s2="**+**s2**+**"\n\n"**);**    String s3**=**"a99+0X55f-55/0xAF+33"**;**  String**[]** sd**=**s3**.**split**(**"0[Xx]([0-9a-fA-F])+"**);**  **for(**String ss**:**sd**)**System**.**out**.**print**(**"=="**+**ss**);**  **}**  **}** |

* 实验结果：



二：对实验1设计的矩阵类Matrix进行抽象设计，通过读取文本文件中的数据来创建或初始化矩阵数据，将矩阵加减运算的结果存入文件。

* 思路分析

由题意：在实验一的基础上需要做到两点：

1. 抽象化矩阵类，较为简单，定义抽象类，然后再具体设计中继承，并实现其中的抽象方法即可。
2. 文本创建或初始化矩阵数据：基于上面的抽象化设计，比较于实验一，只需要多一种初始化矩阵的方法即可，通过初始化函数的输入参数不同，进行不同的初始化办法。



* 具体代码：

1. 抽象化矩阵类：

|  |
| --- |
| package ExperimentSourceCode\_4**.**MatrixIO**;**  **import** java**.**io**.**File**;**  **import** java**.**io**.**IOException**;**  **import** java**.**util**.**Arrays**;**  public abstract class Matrix **{**  public Matrix**()** **{**  **}**//构造方法  //输入对应参数的方法，定义为抽象==>创建或初始化矩阵的方法  public abstract void Matrix\_IO**(**File fp**)** **throws** IOException**;**  public abstract void Matrix\_IO**(**int row**,** int col**,**double **[][]** m**);**  public abstract int getCol**();**  public abstract int getRow**();**  public abstract double**[][]** getM**();**  //为整个矩阵赋值  public abstract void setM**(**double**[][]** m**);**  //修改矩阵中某个值  public abstract void setN**(**int a**,** int b**,** int N**);**  @Override  //输出整个矩阵  public abstract String toString**();**  //输出矩阵中的某个数  public abstract double getN**(**int a**,** int b**);**  //加法  public abstract double**[][]** matrixAdd**(**Matrix\_File N**);**  //减法  public abstract double**[][]** matrixSub**(**Matrix\_File N**);**  //判断矩阵是否相等  public abstract boolean equals**(**Matrix\_File matrix**);**  **}** |

1. 继承实现矩阵类及测试：

|  |
| --- |
| package ExperimentSourceCode\_4**.**MatrixIO**;**  **import** java**.**io**.\*;**  **import** java**.**util**.**Arrays**;**  **import** java**.**util**.**Scanner**;**  public class Matrix\_File **extends** Matrix **{**  private int row**;**//行  private int col**;**//列  private double**[][]** M**;**  //文件初始化  @Override  public void Matrix\_IO**(**File fp**)** **throws** IOException **{**  Scanner read **=** **null;**  **try{**  read **=** **new** Scanner**(**fp**);**  row **=** read**.**nextInt**();**  col **=** read**.**nextInt**();**  M **=** **new** double**[**row**][**col**];**  **for(**int i**=**0**;**i**<**row**;**i**++){**  **for** **(**int j**=**0**;**j**<**col**;**j**++){**  M**[**i**][**j**]** **=** read**.**nextDouble**();**  **}**  **}**  **}catch** **(**FileNotFoundException e**){**  System**.**out**.**println**(**"未找到该文件！"**);**  **}catch** **(**ArrayIndexOutOfBoundsException e**){**  System**.**out**.**println**(**"文件内矩阵数据数量有误！"**);**  **}catch** **(**ArithmeticException e**){**  System**.**out**.**println**(**"文件内矩阵数据内容有误！"**);**  **}finally** **{**  read**.**close**();**  **}**//关闭文件流  **}**  //直接初始化  @Override  public void Matrix\_IO**(**int row**,** int col**,**double**[][]** m**)** **{**  **this.**col**=**col**;**  **this.**row**=**row**;**  M **=** **new** double**[**row**][**col**];**  M **=** m**;**  **}**  public int getCol**()** **{**  **return** col**;**  **}**  public int getRow**()** **{**  **return** row**;**  **}**  public double**[][]** getM**()** **{**  **return** M**;**  **}**  //为整个矩阵赋值  public void setM**(**double**[][]** m**)** **{**  M **=** m**;**  **}**  //修改矩阵中某个值  public void setN**(**int a**,** int b**,** int N**)** **{**  M**[**a**][**b**]** **=** N**;**  **}**  @Override  //输出整个矩阵  public String toString**()** **{**  **return** "NO1.Matrix{" **+**  Arrays**.**deepToString**(**M**)** **+**  '}'**;**  **}**  //输出矩阵中的某个数  public double getN**(**int a**,** int b**)** **{**  **return** M**[**a **-** 1**][**b **-** 1**];**  **}**  //加法  public double**[][]** matrixAdd**(**Matrix\_File N**)** **{**  double**[][]** K **=** **new** double**[**row**][**col**];**  **try** **{**  **for** **(**int i **=** 0**;** i **<** row**;** i**++)** **{**  **for** **(**int j **=** 0**;** j **<** col**;** j**++)** **{**  K**[**i**][**j**]** **=** M**[**i**][**j**]** **+** N**.**getM**()[**i**][**j**];**  **}**  **}**  **}** **catch** **(**ArrayIndexOutOfBoundsException e**)** **{**  System**.**out**.**println**(**"矩阵运算规则错误！"**);**  **}**//两数组行列数不同->会发生数组下标越界的错误  **return** K**;**  **}**  //减法  public double**[][]** matrixSub**(**Matrix\_File N**)** **{**  double**[][]** K **=** **new** double**[**row**][**col**];**  **try** **{**  **for** **(**int i **=** 0**;** i **<** row**;** i**++)** **{**  **for** **(**int j **=** 0**;** j **<** col**;** j**++)** **{**  K**[**i**][**j**]** **=** M**[**i**][**j**]** **-** N**.**getM**()[**i**][**j**];**  **}**  **}**  **}** **catch** **(**ArrayIndexOutOfBoundsException e**)** **{**  System**.**out**.**println**(**"矩阵运算规则错误！"**);**  **}**  **return** K**;**  **}**  //判断矩阵是否相等  public boolean equals**(**Matrix\_File matrix**)** **{**  **if** **(**matrix**.**getCol**()** **==** **this.**col **&&** matrix**.**getRow**()** **==** **this.**row **&&** Arrays**.**deepEquals**(**matrix**.**getM**(),** **this.**M**))**  **return** **true;**  **else** **return** **false;**  **}**  **}**  class MatrixIOTest**{**  public static void main**(**String**[]** args**)** **{**  double**[][]** C **=** **{{**1**,** 1**},** **{**1**,** 1**}};**  double**[][]** D **=** **{{**1**,** 2**},** **{**1**,** 2**}};**  double**[][]** E **=** **{{**1**,**2**,**6**},{**1**,**5**,**7**},{**0**,**9**,**7**}};**  double**[][]** F **=** **{{**6**,**7**,**8**},{-**1**,**2**,**5**},{**9**,**7**,**6**}};**  Matrix\_File A **=** **new** Matrix\_File**();**  A**.**Matrix\_IO**(**2**,**2**,**C**);**  Matrix\_File B **=** **new** Matrix\_File**();**  B**.**Matrix\_IO**(**2**,**2**,**C**);**  Matrix\_File K **=** **new** Matrix\_File**();**  K**.**Matrix\_IO**(**2**,**2**,**D**);**  Matrix\_File T **=** **new** Matrix\_File**();**  T**.**Matrix\_IO**(**3**,**3**,**E**);**  Matrix\_File U **=** **new** Matrix\_File**();**  U**.**Matrix\_IO**(**3**,**3**,**F**);**  System**.**out**.**println**(**"A="**+**A**.**toString**());**  System**.**out**.**println**(**"B="**+**B**.**toString**());**  System**.**out**.**println**(**"K="**+**K**.**toString**());**  System**.**out**.**println**(**"T="**+**T**.**toString**());**  System**.**out**.**println**(**"U="**+**U**.**toString**());**  //这里的矩阵用数组进行表示，直接调用数组的toString方法  System**.**out**.**print**(**"A+B="**);**  **if(**A**.**matrixAdd**(**B**)!=null)** System**.**out**.**println**(**Arrays**.**deepToString**(**A**.**matrixAdd**(**B**)));**  System**.**out**.**print**(**"A+U="**);**  **if(**A**.**matrixAdd**(**U**)!=null)** System**.**out**.**println**(**Arrays**.**deepToString**(**A**.**matrixAdd**(**U**)));**  System**.**out**.**print**(**"T+U="**);**  **if(**T**.**matrixAdd**(**U**)!=null)** System**.**out**.**println**(**Arrays**.**deepToString**(**T**.**matrixAdd**(**U**)));**  System**.**out**.**print**(**"A-B="**);**  **if(**A**.**matrixSub**(**B**)!=null)** System**.**out**.**println**(**Arrays**.**deepToString**(**A**.**matrixSub**(**B**)));**  System**.**out**.**print**(**"T-U="**);**  **if(**T**.**matrixSub**(**U**)!=null)** System**.**out**.**println**(**Arrays**.**deepToString**(**T**.**matrixSub**(**U**)));**  System**.**out**.**println**(**"T、U是否相等："**+**T**.**equals**(**U**));**  System**.**out**.**println**(**"A、B是否相等：" **+** A**.**equals**(**B**));**  System**.**out**.**println**(**"A、K是否相等：" **+** A**.**equals**(**K**));**  **}**  **}**  class MatrixIOTest\_File**{**  public static void main**(**String**[]** args**)** **throws** IOException **{**  File fp **=null;**  Matrix\_File A **=** **new** Matrix\_File**();**  fp **=** **new** File**(**"src\\ExperimentSourceCode\_4\\FileData\\MatrixDataC.txt"**);**  A**.**Matrix\_IO**(**fp**);**  Matrix\_File B **=** **new** Matrix\_File**();**  fp **=** **new** File**(**"src\\ExperimentSourceCode\_4\\FileData\\MatrixDataC.txt"**);**  B**.**Matrix\_IO**(**fp**);**  Matrix\_File K **=** **new** Matrix\_File**();**  fp **=** **new** File**(**"src\\ExperimentSourceCode\_4\\FileData\\MatrixDataD.txt"**);**  K**.**Matrix\_IO**(**fp**);**  Matrix\_File T **=** **new** Matrix\_File**();**  fp **=** **new** File**(**"src\\ExperimentSourceCode\_4\\FileData\\MatrixDataE.txt"**);**  T**.**Matrix\_IO**(**fp**);**  Matrix\_File U **=** **new** Matrix\_File**();**  fp **=** **new** File**(**"src\\ExperimentSourceCode\_4\\FileData\\MatrixDataF.txt"**);**  U**.**Matrix\_IO**(**fp**);**  System**.**out**.**println**(**"A="**+**A**.**toString**());**  System**.**out**.**println**(**"B="**+**B**.**toString**());**  System**.**out**.**println**(**"K="**+**K**.**toString**());**  System**.**out**.**println**(**"T="**+**T**.**toString**());**  System**.**out**.**println**(**"U="**+**U**.**toString**());**  //这里的矩阵用数组进行表示，直接调用数组的toString方法  System**.**out**.**print**(**"A+B="**);**  **if(**A**.**matrixAdd**(**B**)!=null)** System**.**out**.**println**(**Arrays**.**deepToString**(**A**.**matrixAdd**(**B**)));**  System**.**out**.**print**(**"A+U="**);**  **if(**A**.**matrixAdd**(**U**)!=null)** System**.**out**.**println**(**Arrays**.**deepToString**(**A**.**matrixAdd**(**U**)));**  System**.**out**.**print**(**"T+U="**);**  **if(**T**.**matrixAdd**(**U**)!=null)** System**.**out**.**println**(**Arrays**.**deepToString**(**T**.**matrixAdd**(**U**)));**  System**.**out**.**print**(**"A-B="**);**  **if(**A**.**matrixSub**(**B**)!=null)** System**.**out**.**println**(**Arrays**.**deepToString**(**A**.**matrixSub**(**B**)));**  System**.**out**.**print**(**"T-U="**);**  **if(**T**.**matrixSub**(**U**)!=null)** System**.**out**.**println**(**Arrays**.**deepToString**(**T**.**matrixSub**(**U**)));**  System**.**out**.**println**(**"T、U是否相等："**+**T**.**equals**(**U**));**  System**.**out**.**println**(**"A、B是否相等：" **+** A**.**equals**(**B**));**  System**.**out**.**println**(**"A、K是否相等：" **+** A**.**equals**(**K**));**  **}**  **}** |

* 实验结果：使用文件初始化矩阵内容，结果与实验一致：

