Jarrod Parkes

CS 321 – Introduction to Object-Oriented Programming with Java

Programming Assignment #2

Table, Database, and User Interface (DBMenu) Classes – Assignment Description

Problem Statement: Create a text-based user interface that allows users to interact with the program written in the Assignment1 stage. The user interface, which will wrap the functionality of the first assignment, should handle all exceptions and abnormal situations. Newly required behaviors also include Java input/output to text files and Java Serialization to/from binary files.

DBMenu (Text Based User Interface) Class – Capabilities

1. The UI will display all tables stored in a global database.
2. The UI will display a table whose name is equivalent to a name provided by the user by implementing a linear search through the database.
3. The UI will allow for the addition of tables into the global database by basic keyboard inputs, text file inputs, or binary file inputs.
4. The UI will remove tables from the global database.
5. The UI will provide a mechanism for setting an active table to be manipulated by the base table operations.
6. The UI will provide an additional interface for the active table that contains submenus for table operation.
7. The UI will display individual rows, columns, or entries of the active table.
8. The UI will replace or set individual rows, columns, or entries with user entered data values of the active table.
9. The UI will delete individual rows or columns of the active table.
10. The UI will rotate the active table geometrically in increments of 90 degrees where the rows become columns and columns become rows.
11. The UI will swap any two rows or columns of the active table.
12. The UI will find numerical maximums of an individual row or column of the active table.
13. The UI will find the numerical summation if all values are numerical or character concatenation of mixed types for an individual row or column of the active table.
14. The UI will provide an additional interface for writing the entire contents of the global database to a text file or a binary file (through serialization).
15. The UI will provide a way to save changes to tables and a database through the implementation of the writing to files interface.

Specification for General Design and Complex Methods

1. Input/Output File Format

- The basic text file representation of a database is as follows…

1 //the number of tables in the file

Jarrod //the name of the table

3 //number of rows

3 //number of columns

i1 //value located at row 0 column 0

sBobby //value located at row 0 column 1

d23.0 //value located at row 0 column 2

c% //value located at row 1 column 0

n //value located at row 1 column 1

f4 //value located at row 1 column 2

A type character and the associated value denote what will be stored at the appropriate locations. For example the value located at row 0 and column 0 is of Integer type and has a value of 1. Type casting is taken care of within the program; however, if a type and value do not cast into one another then null is stored at that location and the program continues. If there is more than one table in a file, then the values beginning with the next table name begin after the last value of the previous table. The expected types are as follows…

i – Integer

s – String

d – Double

c – Character

n – Null

f – Float

2. Data Hiding and Class Security

- As apparent by the many protected functions of the Table class and the private functions of the DBMenu class, this project begins the process of hiding the details unnecessary to the user and implementer. In the interest of future projects, any graphical representation of the Table or Database classes must be done within the same package as to avoid a complete global access of these classes. Private methods in the DBMenu class restrict the implementer to only the pollOperation() function which subsequently calls any and all functions necessary to modify the values of individual tables or the global database (which could be modified to be an active database). Regardless, the text-based user interface allows a user to interactively use the program written in the first stage of the project.

3. Exception Handling

- Unlike the first program, this stage of the program checks for the exceptions thrown by Java classes. Some table unique exceptions have also been implemented for cases that are specific to the abnormal use of the Table and Database classes. Most every exception in this stage is handled silently and gives a printed statement verifying the cause of the exception and how it was handled. For some exceptions, such as invalid input types, the stream objects that handle input must be reset to compensate for the next inputs without error.

4. Use of ArrayLists instead of Two-Dimensional Arrays

- The first stage of this program ran into the frequent problem of array reallocation to allow the dynamic changing of tables sizes and dimensions; however, a reimplementation of the underlying data storage to ArrayLists has allowed for an easy, smooth transition to creating a dynamic table that can still uphold the functions of the previous program.

5. DBMenu UML Diagram

|  |
| --- |
| DBMenu |
| - input: Scanner  - activeDatabase: Database  - activeTable: Table |
| + DBMenu(Database)  + pollOperation(): int  - setActiveTable(int)  - display()  - pollCreatTable()  - pollActiveTable()  - pollWriteTables()  - displayOperations()  - replaceOperations()  - deleteOperations()  - swapOperations()  - maxOperations()  - sumOperations()  - menuPrintTables()  - menuTablePrint()  - menuTablePrintRow(int)  - menuTablePrintColumn(int)  - menuTablePrintEntry(int, int)  - menuTableSetRow(int, int)  - menuTableSetColumn(int, int)  - menuTableSetEntry(int, int)  - menuTableDeleteRow(int)  - menuTableDeleteColumn(int)  - menuTableSwapRows(int, int)  - menuTableSwapColumns(int, int)  - menuTableMaxRow(int)  - menuTableMaxColumn(int)  - menuTableSumRow(int)  - menuTableSumColumn(int) |

Test Cases

The test cases listed below are multiple executions of the SecondAssignment.jar file in which a user has given subsequent commands to the database menu. To make sure that testing was unbiased, a user who never saw the development of the project was allowed to test the classes for reliability. Several input files were also required, and the contents of these files are attached to this document. Read the output as columns left to right.

Test #1

Database Menu

[Active Table: NULL]

[Number of Tables: 0]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

1

secondassignment.NoTableException:

Exception in printAll(): No Table Exists

Handled: Fails Silently...

Database Menu

[Active Table: NULL]

[Number of Tables: 0]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

3

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

1

Name of Table: testTable

Number of Rows: 3

Number of Columns: 3

Table testTable Added

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

4

Database Menu

[Active Table: NULL]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

5

Enter Index: -1

java.lang.ArrayIndexOutOfBoundsException: -1

Database Menu

[Active Table: NULL]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

5

Enter Index: 0

Database Menu

[Active Table: testTable]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

6

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

1

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

3

testTable

0 1 2

0 [ ][ ][ ]

1 [ ][ ][ ]

2 [ ][ ][ ]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

5

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

2

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

1

Enter Row Index: 0

Enter Type Then Value (Ex: sJames)

Placing Value At Index 0

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: sCS321

Enter Type Then Value (Ex: sJames)

Placing Value At Index 1

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i42

Enter Type Then Value (Ex: sJames)

Placing Value At Index 2

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: c%

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

1

Enter Row Index: 1

Enter Type Then Value (Ex: sJames)

Placing Value At Index 0

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: n

Enter Type Then Value (Ex: sJames)

Placing Value At Index 1

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: d

java.lang.NumberFormatException: empty String

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

4

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

1

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

3

testTable

0 1 2

0 [ CS321][ 42.0][ %]

1 [ ][ ][ ]

2 [ ][ ][ ]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

5

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

2

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

1

Enter Row Index: 1

Enter Type Then Value (Ex: sJames)

Placing Value At Index 0

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: d30

Enter Type Then Value (Ex: sJames)

Placing Value At Index 1

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: n

Enter Type Then Value (Ex: sJames)

Placing Value At Index 2

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: f14

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

1

Enter Row Index: 2

Enter Type Then Value (Ex: sJames)

Placing Value At Index 0

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i1

Enter Type Then Value (Ex: sJames)

Placing Value At Index 1

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i2

Enter Type Then Value (Ex: sJames)

Placing Value At Index 2

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i3

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

4

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

1

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

3

testTable

0 1 2

0 [ CS321][ 42.0][ %]

1 [ 30.0][ ][ 14.0]

2 [ 1.00][ 2.00][ 3.00]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

5

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

4

Table testTable Rotated 90 Degrees

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

1

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

3

testTable

0 1 2

0 [ %][ 14.0][ 3.00]

1 [ 42.0][ ][ 2.00]

2 [ CS321][ 30.0][ 1.00]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

4

Enter Row Index: 0

Enter Column Index: 2

Table Entry (0,2)

2

0 [ 3.00]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

5

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

5

1) Swap Two Rows

2) Swap Two Columns

3) Back To Table Operations

2

Enter Column Index 1: 0

Enter Column Index 2: 1

1) Swap Two Rows

2) Swap Two Columns

3) Back To Table Operations

3

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

1

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

3

testTable

0 1 2

0 [ 14.0][ %][ 3.00]

1 [ ][ 42.0][ 2.00]

2 [ 30.0][ CS321][ 1.00]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

5

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

2

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

2

Enter Column Index: 0

Enter Type Then Value (Ex: sJames)

Placing Value At Index 0

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i1

Enter Type Then Value (Ex: sJames)

Placing Value At Index 1

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i2

Enter Type Then Value (Ex: sJames)

Placing Value At Index 2

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i3

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

4

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

1

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

3

testTable

0 1 2

0 [ 1.00][ %][ 3.00]

1 [ 2.00][ 42.0][ 2.00]

2 [ 3.00][ CS321][ 1.00]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

1

Enter Row Index: 2

Row 2

0 1 2

2 [ 3.00][ CS321][ 1.00]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

2

Enter Column Index: 4

secondassignment.InvalidDimensionsException:

Exception in printColumn(int): Invalid Column

Handled: Fails Silently...

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

2

Enter Column Index: 0

Column 0

0

0 [ 1.00]

1 [ 2.00]

2 [ 3.00]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

Functionality tested:

1. Basic menu maneuverability
2. Exception handling
3. Creating a table from keyboard input
4. Replacing (Setting) rows and columns
5. Printing the active table, rows, columns, and entries
6. Rotating a table
7. Swapping columns

Test #2

Database Menu

[Active Table: NULL]

[Number of Tables: 0]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

3

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

2

Filename (Ex: file.txt): tables1.txt

Table jarrod Added

Table james Added

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

4

Database Menu

[Active Table: NULL]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

5

Enter Index: 1

Database Menu

[Active Table: james]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

1

jarrod

0 1 2

0 [ James][ 2.00][ 41.3]

1 [ s][ ][ 6.00]

2 [ 7.00][ 8.00][ 9.00]

james

0 1

0 [ 1.00][ 2.00]

1 [ 3.00][ 4.00]

Database Menu

[Active Table: james]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

6

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

3

1) Delete A Row

2) Delete A Column

3) Back To Table Operations

2

Enter Column Index: 0

Column 0 Removed

1) Delete A Row

2) Delete A Column

3) Back To Table Operations

3

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

1

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

3

james

0

0 [ 2.00]

1 [ 4.00]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

5

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

8

Database Menu

[Active Table: james]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

5

Enter Index: 0

Database Menu

[Active Table: jarrod]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

2

Enter Table Name: jarrod

jarrod

0 1 2

0 [ James][ 2.00][ 41.3]

1 [ s][ ][ 6.00]

2 [ 7.00][ 8.00][ 9.00]

Database Menu

[Active Table: jarrod]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

6

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

6

1) Find Maximum In A Row

2) Find Maximum In A Column

3) Back To Table Operations

2

Enter Column Index: 2

Max Value:

[ 41.3]

1) Find Maximum In A Row

2) Find Maximum In A Column

3) Back To Table Operations

1

Enter Row Index: 1

Max Value:

[ 6.00]

1) Find Maximum In A Row

2) Find Maximum In A Column

3) Back To Table Operations

3

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

7

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

1

Enter Row Index: 0

Sum Row 0

James241.32

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

2

Enter Column Index: 0

Sum Column 0

Jamess7

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

3

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

2

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

2

Enter Column Index: 0

Enter Type Then Value (Ex: sJames)

Placing Value At Index 0

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: sJames

Enter Type Then Value (Ex: sJames)

Placing Value At Index 1

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: sJarrod

Enter Type Then Value (Ex: sJames)

Placing Value At Index 2

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: sWynn

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

4

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

1

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

3

jarrod

0 1 2

0 [ James][ 2.00][ 41.3]

1 [ Jarrod][ ][ 6.00]

2 [ Wynn][ 8.00][ 9.00]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

5

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

6

1) Find Maximum In A Row

2) Find Maximum In A Column

3) Back To Table Operations

2

Enter Column Index: 0

Column is NULL!

1) Find Maximum In A Row

2) Find Maximum In A Column

3) Back To Table Operations

3

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

7

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

2

Enter Column Index: 0

Sum Column 0

JamesJarrodWynn

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

3

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

8

Database Menu

[Active Table: jarrod]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

7

1) Write Database To Text File

2) Write Database To Binary File

3) Back to Main Menu

2

Filename (Ex: file.bin): names.bin

Database Written to File

1) Write Database To Text File

2) Write Database To Binary File

3) Back to Main Menu

3

Database Menu

[Active Table: jarrod]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

3

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

3

Filename (Ex: file.bin): names.txt

Invalid File Type

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

3

Filename (Ex: file.bin): names.bin

Table jarrod Added

Table james Added

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

4

Database Menu

[Active Table: jarrod]

[Number of Tables: 4]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

1

jarrod

0 1 2

0 [ James][ 2.00][ 41.3]

1 [ Jarrod][ ][ 6.00]

2 [ Wynn][ 8.00][ 9.00]

james

0

0 [ 2.00]

1 [ 4.00]

jarrod

0 1 2

0 [ James][ 2.00][ 41.3]

1 [ Jarrod][ ][ 6.00]

2 [ Wynn][ 8.00][ 9.00]

james

0

0 [ 2.00]

1 [ 4.00]

Database Menu

[Active Table: jarrod]

[Number of Tables: 4]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

8

Functionality tested:

1. Read tables from a text file and a binary file with serialization
2. Writing (saving) tables to a binary file with serialization
3. Displaying all tables in a database
4. Removing a column from a table
5. Printing a table by name
6. Finding the numerical max of a column and row
7. Finding mixed type summations of a column and row

Test #3

Database Menu

[Active Table: NULL]

[Number of Tables: 0]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

3

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

1

Name of Table: tableX

Number of Rows: 2

Number of Columns: 2

Table tableX Added

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

4

Database Menu

[Active Table: NULL]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

5

Enter Index: 0

Database Menu

[Active Table: tableX]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

6

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

2

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

3

Enter Row Index: 11

Enter Column Index: 1

Enter Type Then Value (Ex: sJames)

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i2

secondassignment.InvalidDimensionsException:

Exception in setCell(Object, int, int): Invalid Row

Handled: Fails Silently...

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

3

Enter Row Index: 1

Enter Column Index: 1

Enter Type Then Value (Ex: sJames)

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i4

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

4

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

1

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

3

tableX

0 1

0 [ ][ ]

1 [ ][ 4.00]

1) Print A Row

2) Print A Column

3) Print Table

4) Print Single Entry

5) Back To Table Operations

5

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

2

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

1

Enter Row Index: 0

Enter Type Then Value (Ex: sJames)

Placing Value At Index 0

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i1

Enter Type Then Value (Ex: sJames)

Placing Value At Index 1

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i2

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

3

Enter Row Index: 1

Enter Column Index: 0

Enter Type Then Value (Ex: sJames)

s - String

i - Integer

c - Character

d - Double

f - Float

n - Null

Value: i3

1) Replace A Row

2) Replace A Column

3) Replace A Single Entry

4) Back To Table Operations

4

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

8

Database Menu

[Active Table: tableX]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

1

tableX

0 1

0 [ 1.00][ 2.00]

1 [ 3.00][ 4.00]

Database Menu

[Active Table: tableX]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

6

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

7

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

1

Enter Row Index: 0

Sum Row 0

3.00000

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

2

Enter Column Index: 1

Sum Column 1

6.00000

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

3

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

8

Database Menu

[Active Table: tableX]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

7

1) Write Database To Text File

2) Write Database To Binary File

3) Back to Main Menu

1

Filename (Ex: file.txt): tableX.txt

Table tableX Written to File

1) Write Database To Text File

2) Write Database To Binary File

3) Back to Main Menu

3

Database Menu

[Active Table: tableX]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

6

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

4

Table tableX Rotated 90 Degrees

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

8

Database Menu

[Active Table: tableX]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

7

1) Write Database To Text File

2) Write Database To Binary File

3) Back to Main Menu

1

Filename (Ex: file.txt): tableX2.txt

Table tableX Written to File

1) Write Database To Text File

2) Write Database To Binary File

3) Back to Main Menu

3

Database Menu

[Active Table: tableX]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

8

Functionality tested:

1. Write tables to a text file
2. Find the numerical summations of a column and row
3. Invalid dimension exceptions

Test #4

Database Menu

[Active Table: NULL]

[Number of Tables: 0]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

1

secondassignment.NoTableException:

Exception in printAll(): No Table Exists

Handled: Fails Silently...

Database Menu

[Active Table: NULL]

[Number of Tables: 0]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

3

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

2

Filename (Ex: file.txt): tables1.txt

Table jarrod Added

Table james Added

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

4

Database Menu

[Active Table: NULL]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

2

Enter Table Name: jarrod

jarrod

0 1 2

0 [ James][ 2.00][ 41.3]

1 [ s][ ][ 6.00]

2 [ 7.00][ 8.00][ 9.00]

Database Menu

[Active Table: NULL]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

4

Enter Index: 0

Table jarrod Removed

Database Menu

[Active Table: NULL]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

5

Enter Index: 4

java.lang.IndexOutOfBoundsException: Index: 4, Size: 1

Database Menu

[Active Table: NULL]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

5

Enter Index: 0

Database Menu

[Active Table: james]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

6

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

7

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

2

Enter Column Index: 8

secondassignment.InvalidDimensionsException:

Exception in sumColumn(int): Invalid Column

Handled: Fails Silently...

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

2

Enter Column Index: 1

Sum Column 1

6.00000

1) Find Sum Of A Row

2) Find Sum Of A Column

3) Back To Table Operations

3

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

8

Database Menu

[Active Table: james]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

1

james

0 1

0 [ 1.00][ 2.00]

1 [ 3.00][ 4.00]

Database Menu

[Active Table: james]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

6

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

4

Table james Rotated 90 Degrees

1) Display Operations

2) Replace Operations

3) Delete Operations

4) Rotate 90 Degrees

5) Swap Operations

6) Find Maximums

7) Find Summations

8) Back to Main Menu

8

Database Menu

[Active Table: james]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

1

james

0 1

0 [ 2.00][ 4.00]

1 [ 1.00][ 3.00]

Database Menu

[Active Table: james]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

2

Enter Table Name: james

james

0 1

0 [ 2.00][ 4.00]

1 [ 1.00][ 3.00]

Database Menu

[Active Table: james]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

7

1) Write Database To Text File

2) Write Database To Binary File

3) Back to Main Menu

2

Filename (Ex: file.bin): test.bin

Database Written to File

1) Write Database To Text File

2) Write Database To Binary File

3) Back to Main Menu

3

Database Menu

[Active Table: james]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

3

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

3

Filename (Ex: file.bin): test.bin

Table james Added

1) Create from Input

2) Create from File

3) Create from Binary File

4) Back to Main Menu

4

Database Menu

[Active Table: james]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

1

james

0 1

0 [ 2.00][ 4.00]

1 [ 1.00][ 3.00]

james

0 1

0 [ 2.00][ 4.00]

1 [ 1.00][ 3.00]

Database Menu

[Active Table: james]

[Number of Tables: 2]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

4

Enter Index: 1

Table james Removed

Database Menu

[Active Table: james]

[Number of Tables: 1]

1) Display All Tables

2) Display Table By Name

3) Add Tables

4) Remove Table

5) Set Active Table

6) Operate Active Table

7) Write To File (Save)

8) Exit Program

8

Functionality tested:

1. Table Removal
2. Out of Bounds Exception
3. Printing Table by Name
4. Invalid Input Exceptions