CS209 Lab11
fa2014
dutta/streller

PROBLEM STATEMENT:

The use of binary files; basic file input/output programming.

Notes:
In this lab exercise you will run and test an instructor-provided program.
Then you will extend your program from Lab 8 to include the saving
and retrieving of Loan data to/from file.

CODE :

Obtain the zip file. The zip file contains LoanMultiTypeSeqFileExe.java: main class/file and driver for the demo program. LoanMultiTypeOpenListener.java: contains listener class and event handler method for opening and reading a sequential file comprised of multiple data types.
LoanMultiTypeSaveListener.java: contains listener class and event handler method for writing a sequential file comprised of multiple data types.
Loan.java: file to be used only to run instructor-provided program; contains only a rudimentary toString() method.

Create a project in NetBeans for this assignment.
Add the 4 instructor-provided classes (files) to your project.
Run and test the program as indicated below.
Note: This program only writes and reads part of the Loan data to/from file.
As explained below, your job is to extend your Lab Exercise 8 program to include capabilities for writing the list of loans (including all their attributes) to file and reading a list of Loans (including all their attributes) from a file.

Run the program comprised of the instructor-provided files mentioned above.
To run the program, run the file LoanMultiTypeSeqFileEx.java.

Test the functionality of the instructor-provided program.

If you enter an integer in the first field and hit the Enter key, the data on the loan in that position in the list (array) is displayed in the JTextFields. If no loan has been inserted into the selected list position, a message to that effect is displayed.

If you enter data in all the fields and hit the Enter key in any field other than the first Loan Position field, then:

if the array cell at the indicated position is empty, the loan is added to the array;

otherwise if there is an object in the array cell at the indicated position, then the loan object is replaced with a loan created using the new updated data.
.

After having entered several loan objects into the array, click on the Save File button. This will write the array of loan objects to a file, whose name and location is specified by the user.

Important note: The saved file is not a text file, and therefore cannot be read or viewed in a tool such as Microsoft Word, NotePad, or WordPad. The data items are saved in binary form.

After having saved the list of loans to file by clicking the Save File button, exit the program. To do this, click the “X” button in the upper-right corner of the window.

Re-start the program. Do not enter data on any loans. Instead, click the Open File button, and specify the name and location of the file to which you saved the list of taxpayers in the previous step. The program should read the file and display the list of loans in the JTextArea.

You should then be able to continue to add and change loans in the array list. You should be able to save the Loan list to file whenever desired, or open (read) an loan list from file into the program, as desired.

Instructions for file writing and reading.

In this program and in the program that you write as part of this exercise, file writing and reading is performed using the methods of the DataInputStream and DataOutputStream classes. These classes are being used because they provide methods to read/write the different Java data types.

The class MultiTypeSaveListener contains code that writes a list of loans to a file. This class illustrates how to write each item to file in its binary form using the methods of the DataOutputStream class.

As part of this exercise and as illustrated in the class MultiTypeSaveListener, your code must write each data item to file in its binary form:

Text string data items are written to the file as Strings of variable length. Since the strings are of variable length, a semi-colon is used as the special character to mark the end of a string in the file. Use the method writeChars from the DataOutputStream class.

Loan member variables of type String: name, city, loanType.

Integers are written as binary ints. Therefore, each has the same fixed length, namely 32 bits (4 bytes). Since all ints have the same fixed length, no special character is needed to mark the end of an int. Use method writeInt from the DataOutputStream class. Do not insert a special character to mark the end of an int.

Loan member variables of type int: idNumber,numberOfYears.

Decimal numbers are written as binary doubles. Therefore, each has the same fixed length, namely 64 bits (8 bytes). Since all doubles have the same fixed length, no special character is needed to mark the end of a double. Use method writeDouble from the DataOutputStream class. Do not insert a special character to mark the end of a double.

Loan member variables of type double: propertyCost, downPayment, interestRate.

The class MultiTypeOpenListener contains code that reads a list of taxpayers from a file that has been written by the program. This class illustrates how to read each item from a file containing the data in binary form, using the methods of the DataInputStream class.

As part of this exercise and as illustrated in the class MultiTypeOpenListener, your code must read each data item from a file containing the data in binary form.

In the next steps, you will modify and extend your Lab 8 program so that all the data on all the “loaners”is written to a file and read from a file.

Add panel of buttons for file operations to your Lab 8 program.

> Copy the code from the file LoanMultiTypeSeqFileExe.java that creates and adds the button panel for file operations to the GUI (the panel containing the buttons Open File and Save File). The primary code consists of lines 105 - 122. Paste this code into your Lab 8 program.
>
> Copy and paste any other necessary related code also (e.g., declaration of the String variable named docPath).
>
> Make any other necessary modifications to the code. This would include making it consistent with your Lab 8 program.

In class LoanMultiTypeSaveListener, add code to write all the rest of the data on every loan to file.

> In the actionPerformed method, modify and extend the code so that it writes the rest of the data on each taxpayer to file. The only taxpayer attributes that are currently written to file are the name and idNumber.
>
> You must add code to write the following member variables or attributes to the file in addition to the name and idNumber:
>
> city as a String.
> loanType as a String.
> numberOfYears as an integer.
> propertyCost as a double.
> downPayment as a double
> interestRate as a double
>
> occupation as a String.
> taxpayerType as a String.
> grossPay as a double.

In class MultiTypeOpenListener, add code to read all the rest of the data on every loan from file.

> In the actionPerformed method, modify and extend the code so that it reads the rest of the data on each taxpayer from file. The only taxpayer attributes that are currently read from file are the name and idNumber.
>
> You must add code to read the following member variables or attributes from the file in addition to the name and idNumber:
>
> city as a String.
> loanType as a String.
> numberOfYears as an integer.
> propertyCost as a double.
> downPayment as a double.
> interestRate as a double

Make any other necessary additions and changes to your program.

Run, test, and debug your program.

> Make sure that all the member variables are written to file by the Save File handler.

Make sure that all the member variables are read from file by the Open File handler.

Perform the following test when running your program: Create a “loaner” list, write the list to a file, and exit the program. Then re-start the program and read the taxpayer list from the saved file. Make sure that the taxpayer list displayed in the program's JTextArea is the same as it was just before the list was written to file.

After getting the modified program to work correctly, make window captures for submission for credit.

Create batch file.

Create ReadMe file.

Zip the project folder and all its contents.

Change the name the zip file so that its name consists of your name along with the assignment number, as follows: "JohnJones-12.zip".

Deliverables:

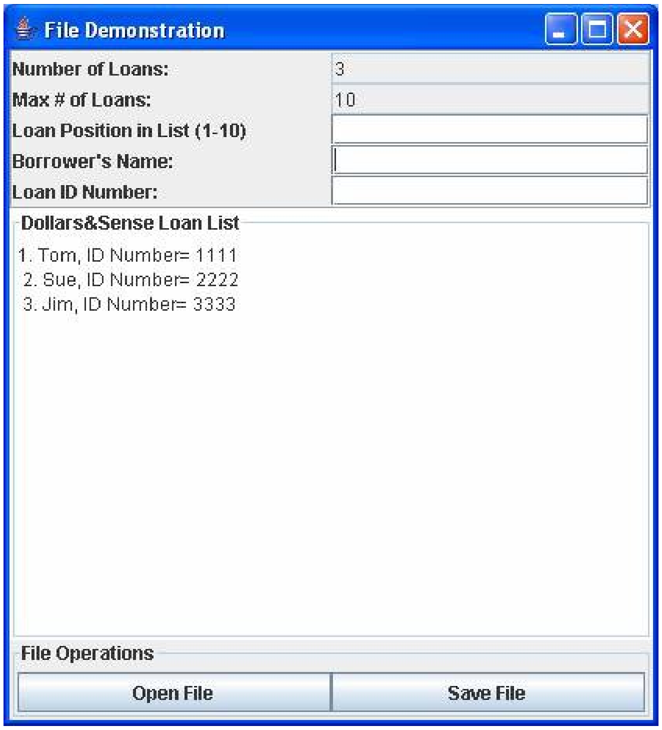
Send to streller@ecc.edu an email this the exact subject

cs209_ Lab_12

In this email attached the above named zip file

LastName_ Lab_12_cs209.zip

Due Date : 5:00pm 11 December 2014



Number of Loans: 3

Max # of Loans: 10

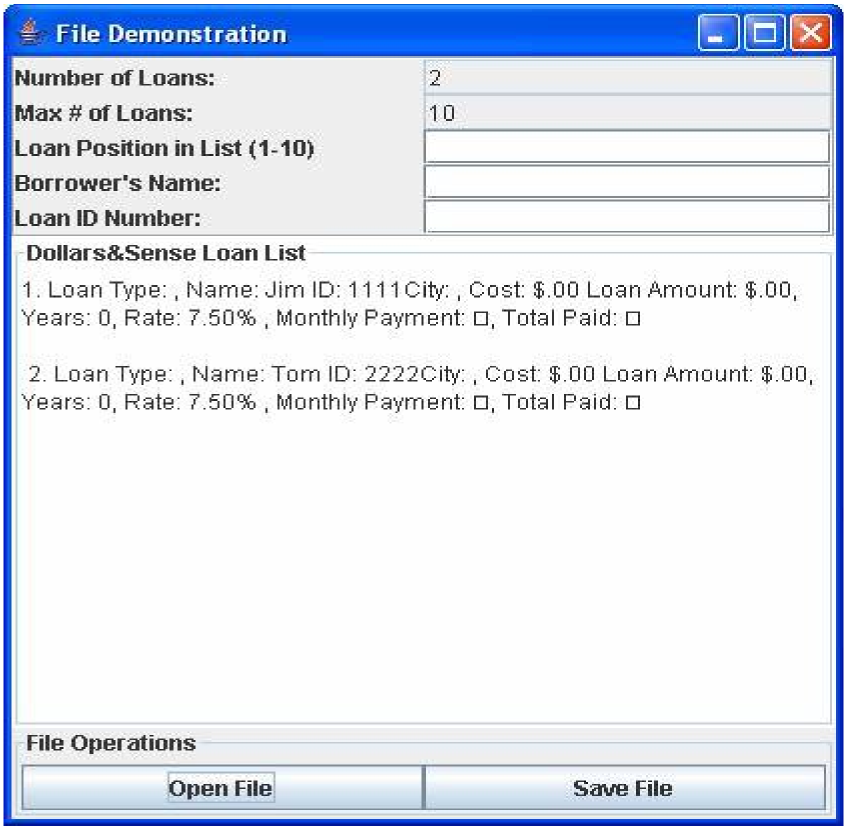
Loan Position in List (1-10)

Borrower's Name:

Loan ID Number:

Dollars&Sense Loan List

```
1. Tom, ID Number= 1111
 2. Sue, ID Number= 2222
 3. Jim, ID Number= 3333
```



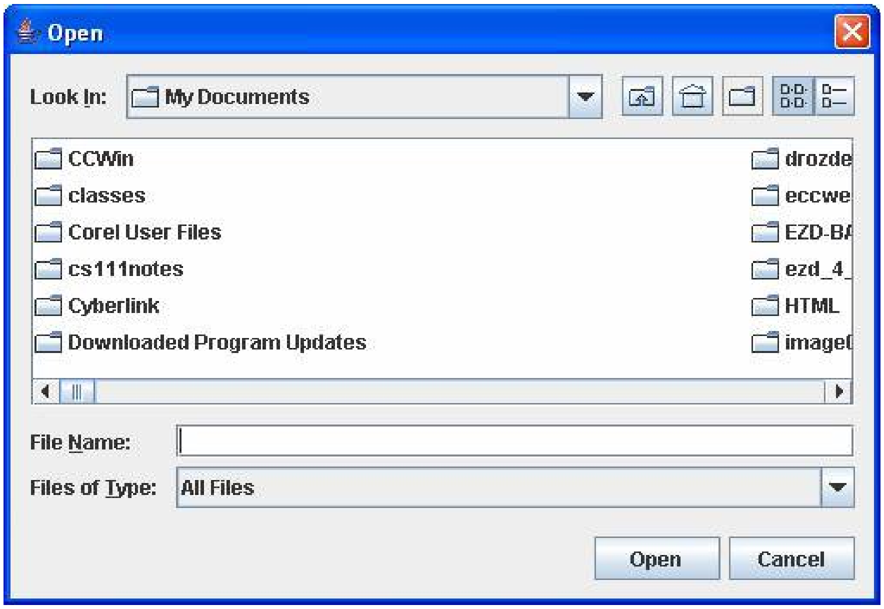
Number of Loans:	2
Max # of Loans:	10
Loan Position in List (1-10)	
Borrower's Name:	
Loan ID Number:	

Dollars&Sense Loan List

```
1. Loan Type: , Name: Jim ID: 1111City: , Cost: $.00 Loan Amount: $.00,
Years: 0, Rate: 7.50% , Monthly Payment: □, Total Paid: □

 2. Loan Type: , Name: Tom ID: 2222City: , Cost: $.00 Loan Amount: $.00,
Years: 0, Rate: 7.50% , Monthly Payment: □, Total Paid: □
```

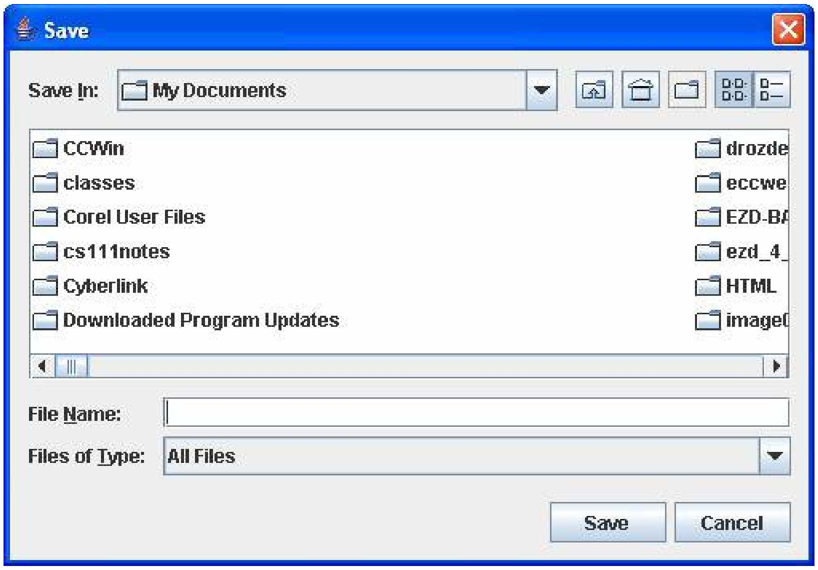
File Operations



Look In: My Documents

File Name:

Files of Type: All Files



Save In: My Documents

File Name:

Files of Type: All Files