University of Maryland University College

CMIS 242 6382 Intermediate Programming (2165)

Submit Files - Project 2

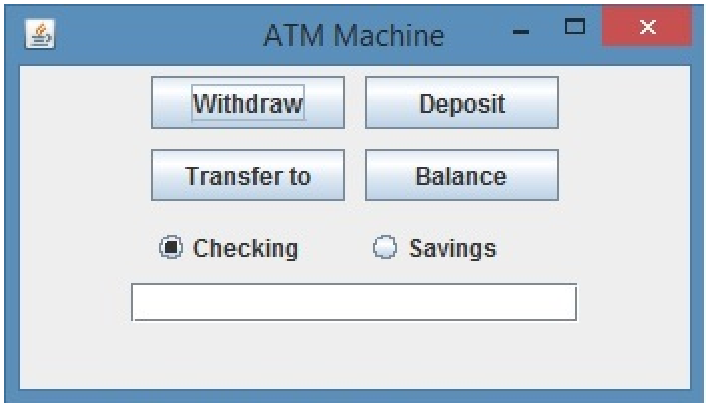
CMIS 242 6382 Intermed... Justin Casteel

Submission Folder

Project 2

Instructions

The second project involves writing a program that implements an ATM machine. The interface to the program should be a GUI that looks similar to the following:



The program should consist of three classes. The first class should define the GUI and should be hand-coded and not generated by a GUI generator. In addition to the main method and a constructor to build the GUI, event handlers will be needed to handle each of the four buttons shown above. When the *Withdraw* button is clicked, several checks must be made. The first check is to ensure the value in the text field is numeric. Next a check must be made to ensure the amount is in increments of $20. At that point an attempt to withdraw the funds is made from the account selected by the radio buttons. The attempt might result in an exception being thrown for insufficient funds, If any of those three errors occur a `JOptionPane` window should be displayed explaining the error. Otherwise a window should be displayed confirming that the withdrawal has succeeded. When the *Deposit* button is clicked the only necessary check is to ensure that the amount input in the textfield is numeric. Clicking the *Transfer* button signifies transferring funds to the selected account from the other account. The checks needed are to confirm that the amount supplied is numeric and that there are sufficient funds in the account from which the funds are being transferred. Clicking the *Balance* button will cause a `JOptionPane` window to be displayed showing the current balance in the selected account. The main class must contain two `Account` objects, one for the checking account and another for the savings account.

The second class is `Account`.It must have a constructor plus a method that corresponds to each of the four buttons in the GUI. It must also incorporate logic to deduct a service charge of $1.50 when more than four total withdrawals are made from either account. Note that this means, for example, if two withdrawals are made from the checking and two from the savings, any withdrawal from either account thereafter incurs the service charge. The method that performs the withdrawals must throw an `InsufficientFunds` exception whenever an attempt is made to withdraw more funds than are available in the account. Note that when service charges apply, there must also be sufficient funds to pay for that charge.

The third class is `InsufficientFunds`, which is a user defined checked exception.

Be sure to follow good programming style, which means making all instance and class variables `private`, naming all constants and avoiding the duplication of code. Furthermore you must select enough scenarios to completely test the program.

Due Date

Jun 12, 2016 11:59 PM

Hide Rubrics

Submit Cancel

CMIS 242 6382 Intermed... Justin Casteel

Design	20 points (18-20 points) Employs Modularity (including proper use of parameters, use of local variables etc.) most of the time Employs correct & appropriate use of programming structures (loops, conditionals, classes etc.) most of the time Efficient algorithms used most of the time Excellent use of object-oriented design	17 points (15-17 points) Employs Modularity (including proper use of parameters, use of local variables etc.) some of the time Employs correct & appropriate use of programming structures (loops, conditionals, classes etc.) some of the time Efficient algorithms used some of the time Good use of object-oriented design	14 points (0-14 points) Rarely employs Modularity (including proper use of parameters, use of local variables etc.) Rarely employs correct & appropriate use of programming structures (loops, conditionals, classes etc.) Poorly structured and inefficient algorithms Rarely uses good object-oriented design
Functionality	40 points (36-40 points) Program fulfills all functionality All requirements were fulfilled Extra effort was apparent	35 points (29-35 points) Program fulfills most functionality Most requirements were fulfilled	28 points (0-28 points) Program does not fulfill functionality Few requirements were fulfilled
Test	20 points (18-20 points) Comprehensive test plan	17 points (15-17 points) Good test plan included	14 points (0-14 points) No test plan included
Documentation	20 points	17 points	14 points

	(18-20 points) Excellent comments Comprehensive lessons learned Excellent possible improvements included	(15-17 points) Good comments Some lessons learned Some possible improvements included Some approach	(0-14 points) No comments No lessons learned No possible improvements No approach discussion

Submit Cancel

CMIS 242 6382 Intermed... Justin Casteel

Overall Score	**90 or more**	**70 or more**	**0 or more**

Submit Files

Files to submit *

(0) file(s) to submit

After uploading, you must click Submit to complete the submission.

Add a File Record Audio

Comments

Paragrap ▼