CS-162

Lab #7: Exception Handling

General

At the top of the every document that you create (word processing files or source files) include:

```
/**
 * Description of the class or document.
 *
 * @author YOUR NAME
 * @version CS162 Lab #, mm/dd/yyyy (replace with the last edit date)
*/
```

Submit in this lab via Moodle using the "Lab #" link. Your assignment must be uploaded by the assigned due date and time. Moodle will automatically close the link at that time. It is strongly highly recommend you do not wait until the very last minute to submit your work.

Concepts

This lab will introduce you to Java's syntax for handling abnormal execution situations known as exception handling. You will learn how to test for these situations and how to use Java syntax to handling these occurrences that interrupts the normal flow of execution.

Background

Read chapter #12 and review the online tutorials and self study quizzes on exception handling. Review your pre-lab assignment before completing this lab.

Assignment

Description: For this lab, you will be creating a new project and several new classes. You do not start with any existing code. You will be creating 4 classes, CustomerClient, AccountServer, AccountManager, and InvalidWithdrawalException.

This lab is open ended for you to design within the required features in the lab requirements list below. You are free to enhance or explore anything you wish beyond these lab requirements.

Lab Requirements:

1. Create a new project called "Exceptions"
2. Create 4 classes, "CustomerClient", "AccountServer", "AccountManager" and "InvalidWithdrawalException"
3. Create an Exception class called InvalidWithdrawalException that inherits from Exception. This will be a checked exception.
4. The AccountServer must throw "InvalidWithdrawalException" (a custom exception class the you write) when a withdrawal is attempted that exceeds the balance value of the account
5. The AccountServer must throw "IllegalArgumentException" (an unchecked exception) when a negative amount is attempted for either withdrawals or deposits
6. Both exceptions being thrown must add detailed description information including the invalid number value that created the error conditions.
7. The "CustomerClient must have at least a constructor and two method called "customerWithdrawal" and "customerDeposit. These methods must catch and handle (not throw) any CHECKED exceptions, but it should not catch any UNCHECKED exceptions.

8. The AccountManager must have a "main" method that creates the instances of a customer and account, and passes the account object handle to the customer so that the customer can do their bank activity. It should do a series of deposits and withdrawals, including some where the withdrawal attempted is more than the bank balance. A negative deposit and withdrawal should also be tested

Additional lab requirements:

1. Add JavaDoc comments for ALL public METHODS and CLASSES that you write
2. Add a JUnit test class for ONE of your classes, and create at least 2 JUnit test methods that test some aspect of your class.

Submission:

Submit your entire BlueJ project (zip only, no tar, 7zip, or other file compression types) folder for this lab into the CS162, Lab2 upload link (just as was done in CS161).