Test cases

>java ChangeUKConsole 351
Your change is
1 pounds
1 crowns
1 half-crowns
1 shillings
1 sixpence
1 tuppence
1 pennies

>java ChangeUKConsole 473
Your change is
1 pounds
3 crowns
1 half-crowns
1 shillings
1 sixpence
2 tuppence
1 pennies
System.out.println(cTen + " tens");
System.out.println(cFive + " fives");
System.out.println(cOne + " ones");
System.out.println(cQuart + " quarters");
System.out.println(cDime + " dimes");
System.out.println(cNickel + " nickels");
System.out.println(cPenny + " pennies");
}

ChangeUSUI

import javax.swing.JOptionPane;

public class ChangeUSUI {
    public static void main(String[] args) {
        double amt = Double.parseDouble(
            JOptionPane.showInputDialog("Enter amount:")
        );
        int intAmt = (int) (amt * 100);

        int cTwenty = intAmt / 2000;
        intAmt = intAmt % 2000;

        int cTen = intAmt / 1000;
        intAmt = intAmt % 1000;

        int cFive = intAmt / 500;
        intAmt = intAmt % 500;

        int cOne = intAmt / 100;
        intAmt = intAmt % 100;

        int cQuart = intAmt / 25;
        intAmt = intAmt % 25;

        int cDime = intAmt / 10;
        intAmt = intAmt % 10;

        int cNickel = intAmt / 5;
        intAmt = intAmt % 5;

        int cPenny = intAmt;

        String result = "Your change is\n" + cTwenty + " twenties\n" +
            cTen + " tens\n" + cFive + " fives\n" + cOne + " ones\n" +
            cQuart + " quarters\n" + cDime + " dimes\n" +
            cNickel + " nickels\n" + cPenny + " pennies";

        JOptionPane.showMessageDialog(null, result, "Result",
            JOptionPane.PLAIN_MESSAGE);

        System.exit(0);
    }
}
Hints and Notes

1. The source files for ChangeUSConsole.java and ChangeUSUI.java are available in the course drop box area G:\Assignments\Burke\ISDS 372\ as well as from the course web site: http://ecommerce.cbe.fullerton.edu/~rburke/courses/f01/isds372/hw/

2. The line “int cTwenty = intAmt / 2000;” computes how many twenty dollar bills should be given in change. How does it do this? Note that the variable cTwenty is declared as an integer, so any fractional amount after the division on the right-hand side is discarded. intAmt is the monetary quantity in cents, so the result of division by 2000 (= $20 in cents) and rounding is the maximum number of whole twenties that can be gotten from the input.

3. The line “intAmt = intAmt % 2000;” reduces intAmt by a quantity corresponding to the quantity of cents in the twenty dollar bills calculated in the line above. The % operator is the modulus operator, meaning “remainder after integer division”. The same effect could be achieved with the following line “intAmt = intAmt – (cTwenty * 2000);”

4. The “c” prefix is a stylistic convention meaning “count”. cTwenty can be read as the “count of twenties,” precisely the meaning that is desired. We will discuss naming conventions and other stylistic issues later in the class.

5. When running the console application from JBuilder, the command line arguments must be input on the “Project Preferences” dialog, “Run” tab, “Program arguments” text box. This is the same part of the Preferences dialog that you use to set your “main” class to run your program.

Java code

ChangeUSConsole

```java
public class ChangeUSConsole {
    public static void main(String[] args) {
        double amt = Double.parseDouble ( args[0]);
        int intAmt = (int) (amt * 100);

        int cTwenty = intAmt / 2000;
        int intAmt = intAmt % 2000;

        int cTen = intAmt / 1000;
        int intAmt = intAmt % 1000;

        int cFive = intAmt / 500;
        int intAmt = intAmt % 500;

        int cOne = intAmt / 100;
        int intAmt = intAmt % 100;

        int cQuart = intAmt / 25;
        int intAmt = intAmt % 25;

        int cDime = intAmt / 10;
        int intAmt = intAmt % 10;

        int cNickel = intAmt / 5;
        int intAmt = intAmt % 5;

        int cPenny = intAmt;

        System.out.println("Your change is");
        System.out.println(cTwenty + " twenties");
    }
}
```
Homework #1
Make Change (Console and Swing versions)
ISDS 372, Prof. Robin Burke
Assigned: 8/23/01
Due: 8/30/01

Objective

Adapt a pair of Java classes to make change in (old-time) English currency rather than US.

Activity

Students will adapt the Java classes included in this assignment to make change in a different currency system. There are two Java classes: ChangeUSConsole and ChangeUSUI. They both take in a dollar amount in decimal format and print out correct change. ChangeUSConsole is a console application that takes its input on the command line and writes its output to the standard output (console). ChangeUSUI is a UI application that takes its input from a dialog box and writes its output in an output window.

To adapt the classes, create a new JBuilder project called Change. Copy the supplied class files to its “src” directory and add these files to the project. Then, make a copy of each file renamed as ChangeUKConsole and ChangeUKUI, respectively. Edit the class names in the file to be ChangeUKConsole and ChangeUKUI, respectively.

Then change the numeric calculation in each class to reflect the old English currency system:

- Penny
- Tuppence = 2 pence
- Sixpence = 6 pence
- Shilling = 12 pence
- Half-crown = 2 shillings + sixpence = 30 pence
- Crown = 5 shillings = 60 pence
- Pound = 20 shillings = 240 pence

Input should be in terms of pence, since decimal notation makes no sense in this system. Choose which application to run by picking the Project Properties “Main class” setting appropriately.

Submission

Students should

Turn in at the start of class a hardcopy of the code of your two class files with a cover page clearly indicating the number and name of the assignment and the student’s name and ID #.

Before class time, submit a folder containing the complete JBuilder project for the Java classes to the on-line course Drop Box for homework #1. This folder is accessible as a shared volume on lab and classroom machines at G:\Assignments\Burke\ISDS 372\HW1. You must copy the entire folder at once: files and folders placed on the server cannot be modified. Your folder should be named with your last name, the last four digits of your student id and the assignment number. For example: Burke1234_HW1.

Assessment

This assignment will be assessed on the completeness of the solution to the problem. Partial solutions will be given partial credit but only for those features of the application that operate. No credit will be given for non-functional code.