Objective

Rewrite the change making application (yet again). This time the program should be flexible enough to handle any kind of currency, and should accept input either from the command line or from user input.

Activity

- Central to this assignment will be five arrays declared as class variables:
  ```java
  public static final String [] aCurrencyName = { "US", "UK" };
  public static final int [] aCurrencyFactor = { 100, 1 };
  public static final int [][] aaDenomination =
  { { 2000, 1000, 500, 100, 25, 10, 5, 1 },
    { 240, 120, 60, 30, 12, 2, 1 } };
  public static final String [][] aaSingularName =
  { { "twenty", "ten", "five", "one", "quarter", "dime", "nickel", "penny" },
    { "pound", "crown", "half crown", "shilling", "sixpence", "tuppence", "penny" } };
  public static final String [][] aaPluralName =
  { { "twenties", "tens", "fives", "ones", "quarters", "dimes", "nickels", "pennies" },
    { "pounds", "crowns", "half crowns", "shillings", "sixpence", "tuppence", "pence" } };
  ```

- You will also need to define at least seven methods in addition to the main method. These methods will
  1. get the currency type
     If there are arguments on the command line, assume the first one is the currency type
     Otherwise get the currency type from the user
  2. get the currency amount
     If there are arguments on the command line, assume the second one is the currency amount
     Otherwise get the currency amount from the user
     Apply the currency factor to get an integer amount
  3. compute the change in all denominations
     For all of the denomination in the chosen currency type
     Make change for the denomination and update the amount
  4. compute the change in a particular denomination
     Compute the integer dividend of the amount and denomination size
     Put in the result array
     Return remaining amount
  5. assemble the output for all of the denominations
     For all of the denominations in the chosen currency type
     Format the value for the denomination
     Add the string to the output string
     Return the assembled output string
6. format a particular denomination
   If the count is 0 for this denomination, output nothing
   If the count is 1, output the 1 followed by the singular denomination name
   If the count is > 1, output the count followed by the plural denomination name
7. output the value appropriately
   If there are arguments on the command line, output the results to System.out
   Otherwise, open in dialog box and display the result.

- Once you have this program working (check it against your previous versions), add one more currency. If your program is correctly constructed, this will only require that you change the contents of the arrays, no other program modifications should be necessary.
  “Silly” currency (basic unit = the blip)
  - 1 portzebie = 180 blips
  - 1 halfzebie = 90 blips
  - 1 foo = 60 blips
  - 1 hafoo = 30 blips
  - 1 bar = 10 blips
  - 1 baz = 3 blips

Submission

Students should

Turn in at the start of class a hardcopy of the code of your class file 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 #4. This folder is accessible as a shared volume on lab and classroom machines at \Doctor\Assignments\Burke\ISDS 372\HW4\. 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_HW4. If you make a mistake and have to submit the folder again, add a letter to the end. The system will not permit you to overwrite your first submission. I will grade the most recent folder (submitted up to class time).

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.

Hints and Notes

1. You can use the length object variable to find out the length of an array. For example, to find out how many different currencies are in the system, you can get the value of aCurrency.length.
2. You will need to have a local array variable to hold the result of breaking the amount into denominations. It might be declared as follows:
   ```java
   int [] acDenominations = new int [aaDenomination[currencyType].length];
   ```
3. You can use the getValue code from the previous assignment in this one, if you wish.
4. Remember that + can be used to concatenate String values together. The special character “\n” can be used to insert a newline character in a String.