Department of Electrical and Computer Engineering
ECSE 202 – Introduction to Software Development
Assignment 1
Introduction to Java Programming

Problem Description

The objective of this assignment is to learn how to write a simple Java program that uses a simple text-based interface to communicate with a user. Specifically this program, baseConversion, will read in a text string representing a number in a base from 2 to16, convert it to a Base-10 number, and then convert it to a number in Base-N, where N is specified by the user. Some examples are shown below.

```
Base conversion program
Enter a number (Base 2-16): 1010011110011101010
Enter the corresponding base as an integer: 2
The decimal equivalent of your input is 343274
Enter the target base for conversion (2-16): 16
1010011110011101010 Base-2 ---> 53cea Base-16

Enter a number (Base 2-16): 177371
Enter the corresponding base as an integer: 8
The decimal equivalent of your input is 65273
Enter the target base for conversion (2-16): 16
177371 Base-8 ---> fef9 Base-16

Enter a number (Base 2-16): ff00
Enter the corresponding base as an integer: 16
The decimal equivalent of your input is 65280
Enter the target base for conversion (2-16): 10
ff00 Base-16 ---> 65280 Base-10

Enter a number (Base 2-16): 2000
Enter the corresponding base as an integer: 3
```

```
The decimal equivalent of your input is 54
Enter the target base for conversion (2-16): 2
2000 Base-3 ---> 110110 Base-2

Enter a number (Base 2-16): 65535
Enter the corresponding base as an integer: 10
The decimal equivalent of your input is 65535
Enter the target base for conversion (2-16): 16
65535 Base-10 ---> ffff Base-16

Enter a number (Base 2-16):
Program terminated
```

Where To Begin?

You can do much of this assignment with what is provided in the class notes, but there are some items that we will not get to for a while. These will be covered in the tutorials during the weeks preceding the due date. If you have written code before, then it is likely that you know how to proceed. For those of you who have not, consider this an introductory tutorial. Here we will provide a skeleton for the program to get you started (which you may use provided that you so indicate in your documentation).

Step 1 – the main class

```
/**
 * A Java program must consist of at least one class containing one method.
 * In the examples done in the lectures, our main class was an extension of
 * one of the acm classes (ConsoleProgram, DialogProgram, GraphicsProgram).
 * For this assignment we will use ConsoleProgram.
 */

public class baseConv extends ConsoleProgram {

/**
 * The run method is called when the program is run; consequently it is
 * the starting point for our program.  Our strategy will be to keep this
 * method simple – communicate with the user and call other methods to
 * perform the requisite functions.
 */

  public void run() {
            println(“Base conversion program”);   // Identify program
/**
 * This program runs in a loop until the user enters a blank line.  It starts off
 * by prompting the user to enter a number to be converted which is read in as a
 * character sring.  The program then prompts for an integer corresponding to the
 * input base.
 */

    while(true) {
      String input = readLine(“Enter a number (Base 2-16): “);                // Read input as string
      if (input.equals(“”)) break;                                            // Exit if blank
      int inBase = readInt(“Enter the corresponding base as an integer: “);   // Read input base

/**
 * Convert the input string to an integer using the String2Int method (which you
 * will have to write).  Echo the result back to the user.
 */

      int numBase10 = String2Int(input,inBase);
      println("The decimal equivalent of your input is "+numBase10);

/**
 * Finally, prompt the user for the base in which to render the final result.
 * Call the Dec2Base method (again, which you need to write) to perform the
 * conversion, and print the result.
 */

      int targetBase = readInt("Enter the target base for conversion (2-16): ");    // Read target base
      String output = Dec2Base(numBase10,targetBase);                                   // Int Base10 to target
      println(input+" Base-"+inBase+" ---> "+output+" Base-"+targetBase);           // Print result
      println();                                                                    // Skip a line
    }

    println(“Program terminated”);
  }
```

Step 2 – String to Integer

The method that converts a string representation to an integer has the following form (signature):

```
private int String2Int(String input, int inBase) {

}
```

The formula for the Base-10 value for a number in an arbitrary base is given by the following formula, $Int10 = \sum_{i=0}^{i=N-1} d_i b^i$, where d_i corresponds to a character representing a digit, and b corresponds to the input base (inBase). Note that the i=0 corresponds to the least significant digit and i=N-1, the most significant digit.

Example:

Suppose the input string was 7a3 with base=11 (Base-11). The Base-10 equivalent would be $7\times 11^2 + 10\times 11^1 + 3\times 11^0 = 960$. Note that digits are encoded such that a=10, b=11, c=12, d=13, e=14 and f=15. This means that your algorithm has to perform the latter conversion, which is best handled by a separate method. With what we know so far, we can write a psuedo-code description of the String2Int algorithm.

input: a string s, corresponding to the number to be converted
input: an integer b, corresponding to the base of the above number

```
int power = 1
int sum = 0
int d

for i=length(s) to 0 by -1
        d = Char2Int(s[i])
        sum = sum + d * power
        power = power * b
end for
```

Coding this algorithm as a method in Java is straightforward. Recall that the charAt(i) method allows you to extract the character at position i in a string.

The String2Int method requires the Char2Int method to convert a character to its corresponding integer value. If the character is in the range [‘0’,’9’], the method returns the corresponding values [0,9]; if the character is in the range [‘a’,‘f’], the method returns the corresponding values [10,15], and if the character is in the range ['A','F'] the method returns the corresponding values in the range [10,15]. This can easily be implemented using if statements and && operators.

Once this routine is written, you can test it in the main program to make sure that the integer value printed is correct.

Step 3 – Integer in Base-10 to Integer in Target Base

This method generates a string corresponding to an integer value in a specified base and has the following signature:

```
          private String Dec2Base(int numBase10, int targetBase) {

          }
```

It uses the following algorithm (described in psuedo-code):

```
     input:  an integer n, number to be converted
     input:  an integer b, the target base for conversion
     output: a list of digits in the target base formatted as a string

     empty string
     while n > 0
          current bit = n % b
          n = n / b
          convert current bit in range [0,15] to character in range ['0','9'], ['a','f'].
          place character at front of string (concatenation operator)
     end while
```

To concert an integer [0,15] to its corresponding character code requires a method, Int2Char with the following signature:

```
     char Int2Char(int digit) {

     }
```

Here is the corresponding psuedo-code:

input: an integer d corresponding to the single integer to be converted in the range [0,15]
output: a single character corresponding to the input digit.

```
if  0 <= d <= 9
        return '0'+ d
else if 10 <= d <= 15
        return 'a'+ d - 10
```

where '0' and 'a' are the integer values of the character codes for 0 and a respectively.

Step 4 – Testing

A good strategy is to build your program incrementally, making sure each component works before attempting to put everything together.

Instructions

1. Write a Java program consisting of a class, baseConv, that implements an interactive program along the lines described earlier in this document. If this is one of the first programs you have written, you can elect to use the example in this document as a starting point (*make sure to indicate this in your comments). More experienced Java programmers may to do this without using acm – the choice is up to you.
2. Run the examples shown earlier and make sure that your results are correct. Save these results in a screen capture and save the results as a pdf file called baseConv.pdf.
3. Make sure that your Java source file is properly documented and that it contains your name and student ID in the comments.
4. Upload baseConv.java and baseConv.pdf on myCourses as indicated.

* Plagiarism also applies to software. If you incorporate code from an external source (i.e. code that you did not write), then you must cite the source of this code in your comments.

About Coding Assignments

We encourage students to work together and exchange ideas. However, when it comes to finally sitting down to write your code, this must be done *independently*. Detecting software plagiarism is pretty much automated these days with systems such as MOSS.

https://www.quora.com/How-does-MOSS-Measure-Of-Software-Similarity-Stanford-detect-plagiarism

Please make sure your work is your own. If you are having trouble, the Faculty provides a free tutoring service to help you along. You can also contact the course instructor or the tutor during office hours. There are also numerous online resources – Google is your friend. The point isn't simply to get the assignment out of the way, but to actually learn something in doing.

fpf/Jan 17, 2020