Name of the Course : Complete Java SE8 Developer Bootcamp

Level : Medium

Tool Stack: Java 8

Problem Statement : Provide a code solution to convert numbers to words up to four digits.

Provide code to achieve the desired output.

Sample Input : 111

Sample Output : one hundred and eleven

Description : Create class **MainClass** with following methods .

1.Public static void main(String arg[])for accepting user input and invokes methods convertToWords(int inumber)

2. public static void convertToWords(int inumber) .It accepts an integer and prints the corresponding value in words.

Code:

import java.util.Scanner;

public class ConvertToWords {

public static void main(String[]args) {

Scanner s = new Scanner(System.in);

int ss = Integer.parseInt(s.nextLine());

System.out.println(convertToWords(ss));

}

// A function that prints

// given number in words

public static String convertToWords(int inumber)

{

char[] num =(new Integer(inumber)).toString().toCharArray();

StringBuffer sb =new StringBuffer();

// Get number of digits

// in given number

int len = num.length;

// Base cases

if (len == 0)

{

// System.out.println("empty string");

sb.append( "empty string");

}

if (len > 4)

{

// System.out.println("Length more than 4 is not supported");

sb.append( "Length more than 4 is not supported");

}

/\* The first string is not used, it is to make

array indexing simple \*/

String[] single\_digits = new String[]{ "zero", "one",

"two", "three", "four",

"five", "six", "seven",

"eight", "nine"};

/\* The first string is not used, it is to make

array indexing simple \*/

String[] two\_digits = new String[]{"", "ten", "eleven", "twelve",

"thirteen", "fourteen",

"fifteen", "sixteen", "seventeen",

"eighteen", "nineteen"};

/\* The first two string are not used, they are to make array indexing simple\*/

String[] tens\_multiple = new String[]{"", "", "twenty", "thirty", "forty",

"fifty","sixty", "seventy",

"eighty", "ninety"};

String[] tens\_power = new String[] {"hundred", "thousand"};

/\* Used for debugging purpose only \*/

// System.out.print(String.valueOf(num)+": ");

/\* For single digit number \*/

if (len == 1)

{

// System.out.println(single\_digits[num[0] - '0']);

sb.append( single\_digits[num[0] - '0'] );

return sb.toString();

}

/\* Iterate while num

is not '\0' \*/

int x = 0;

while (x < num.length)

{

/\* Code path for first 2 digits \*/

if (len >= 3)

{

if (num[x]-'0' != 0)

{

// System.out.print(single\_digits[num[x] - '0']+" ");

// System.out.print(tens\_power[len - 3]+" ");

// here len can be 3 or 4

sb.append( single\_digits[num[x] - '0']+" "+tens\_power[len - 3]+" ");

}

--len;

}

/\* Code path for last 2 digits \*/

else

{

/\* Need to explicitly handle

10-19. Sum of the two digits

is used as index of "two\_digits"

array of strings \*/

if (num[x] - '0' == 1)

{

int sum = num[x] - '0' +

num[x+1] - '0';

// System.out.println(two\_digits[sum]);

sb.append( two\_digits[sum]);

return sb.toString();

}

/\* Need to explicitly handle 20 \*/

else if (num[x] - '0' == 2 &&

num[x + 1] - '0' == 0)

{

// System.out.println("twenty");

sb.append( "twenty");

return sb.toString();

}

/\* Rest of the two digit

numbers i.e., 21 to 99 \*/

else

{

int i = (num[x] - '0');

if(i > 0) {

// System.out.print(tens\_multiple[i]+" ");

sb.append( tens\_multiple[i]+" ");

}

else {

// System.out.print("");

sb.append("");

}

++x;

if (num[x] - '0' != 0)

// System.out.println(single\_digits[num[x] - '0']);

sb.append(single\_digits[num[x] - '0']);

}

}

++x;

}

return sb.toString();

}

}

Junit Testing

import static org.junit.Assert.assertEquals;

import org.junit.Test;

import handson.ConvertToWords;

import handson.DuplicateWord;

public class TestConvertToWords {

@Test

public void testConvertToWords() {

assertEquals("nine thousand nine hundred ninety nine" ,ConvertToWords.convertToWords(9999));

assertEquals("fifty six" ,ConvertToWords.convertToWords(56));

assertEquals("five hundred fifty nine" ,ConvertToWords.convertToWords(559));

assertEquals("seven hundred seventy eight" ,ConvertToWords.convertToWords(778));

assertEquals("three hundred thirty three" ,ConvertToWords.convertToWords(333));

assertEquals("one hundred seventeen" ,ConvertToWords.convertToWords(117));

}

}

Test Data1

Sample input:

111

sample output:

one hundred eleven

Test Data2

sample input :

2546

sample output :

two thousand five hundred forty six

Learning outcome: Participant could able to learn how to use String , StringBuffer ,Arrays.