Lab 5

1 Program to convert a given number to words

Write code to convert a given number into vietnamese words. For example, if “1234” is given as input, output should be “mot nghin hai tram ba muoi tu”.

Example

```java
package edu.hus.oop.lab5;

import java.util.HashMap;
import java.util.Map;

public class Integers {

  // Integer to English
  private static final Map<Integer, String> singleDigits = new HashMap<
      ↪ Integer, String>();

  static {
    singleDigits.put(0, "zero");
    singleDigits.put(1, "one");
    singleDigits.put(2, "two");
    singleDigits.put(3, "three");
    singleDigits.put(4, "four");
    singleDigits.put(5, "five");
    singleDigits.put(6, "six");
    singleDigits.put(7, "seven");
    singleDigits.put(8, "eight");
    singleDigits.put(9, "nine");
    singleDigits.put(10, "ten");
    singleDigits.put(11, "eleven");
    singleDigits.put(12, "twelve");
    singleDigits.put(13, "thirteen");
    singleDigits.put(14, "fourteen");
    singleDigits.put(15, "fifteen");
    singleDigits.put(16, "sixteen");
```

```
29        singleDigits.put(17, "seventeen");
          singleDigits.put(18, "eighteen");
31        singleDigits.put(19, "nineteen");
      }
33
      private static final Map<Integer, String> multiDigits = new HashMap<
          ↪ Integer, String>();
35
      static {
37        multiDigits.put(10, "ten");
          multiDigits.put(20, "twenty");
39        multiDigits.put(30, "thirty");
          multiDigits.put(40, "forty");
41        multiDigits.put(50, "fifty");
          multiDigits.put(60, "sixty");
43        multiDigits.put(70, "seventy");
          multiDigits.put(80, "eighty");
45        multiDigits.put(90, "ninety");
      }
47
      private static final int BILLION = 1000000000;
49    private static final int MILLION = 1000000;
      private static final int THOUSAND = 1000;
51    private static final int HUNDRED = 100;
      private static final int TEN = 10;
53
      private static final String handleUnderOneThousand(int number) {
55        StringBuilder builder = new StringBuilder();
          int x = number;
57        int m = x / HUNDRED;
          int r = x % HUNDRED;
59        if (m > 0) {
            builder.append(singleDigits.get(m)).append("-hundred");
61          x = x % HUNDRED;
          }
63        if (r > 0) {
            if (m > 0) {
65            builder.append(" ");
            }
67          if (x <= 19) {
              builder.append(singleDigits.get(x));
69          } else {
              m = x / TEN;
```

```
71            r = x % TEN;
              if (r == 0) {
73              builder.append(multiDigits.get(x));
              } else {
75              x = x - r;
                builder.append(multiDigits.get(x)).append("-");
77              builder.append(singleDigits.get(r));
              }
79          }
        }
81      return builder.toString();
    }
83
    public static final String toEnglish(int number) {
85      int x = number;
        if (x > Integer.MAX_VALUE || x <= Integer.MIN_VALUE) {
87        throw new IllegalArgumentException("Number has to be <= Integer.
              ↪ MAX_VALUE and > Integer.MIN_VALUE. number=" + x);
        }
89      StringBuilder builder = new StringBuilder();
        if (x == 0) {
91        //Zero is a special case
          builder.append(singleDigits.get(x));
93        return builder.toString();
        }
95      boolean billion = false;
        boolean million = false;
97      boolean thousand = false;
        if (x < 0) {
99        builder.append("negative ");
          // Make the number positive
101       x = x * -1;
        }
103     int m = x / BILLION;
        if (m > 0) {
105       billion = true;
          builder.append(handleUnderOneThousand(m)).append("-billion");
107       x = x % BILLION;
        }
109     m = x / MILLION;
        if (m > 0) {
111       if (billion) {
            builder.append(" ");
```

```
113           }
              million = true;
115           builder.append(handleUnderOneThousand(m)).append("-million");
              x = x % MILLION;
117         }
            m = x / THOUSAND;
119         if (m > 0) {
              if (billion || million) {
121             builder.append(" ");
              }
123           thousand = true;
              builder.append(handleUnderOneThousand(m)).append("-thousand");
125           x = x % THOUSAND;
            }
127         if (billion || million || thousand && x != 0) {
              builder.append(" ");
129         }
            builder.append(handleUnderOneThousand(x));
131         return builder.toString();
        }
133   }
```