

Run.java

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package morsecode;
```

```
import java.util.Scanner;
```

```
/**  
 *  
 * @author Slunger  
 */  
public class Run {
```

```
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        Scanner i = new Scanner(System.in);  
        System.out.println("Enter the string:");  
        String[] str = MorseCode.getCode(i.nextLine());  
        for (String s: str )  
            System.out.println(s);  
    }  
}
```

MorseCode.java

```
/*
```

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

package morsecode;

/**

*

* @author Slunger

*/

public class MorseCode {

private static final String[] LETTERS = new String[]

{".-", "-...", "-.-.", "-..", ".", "-.-", "-.-.",
"....", "..", ".---", "-.-", ".-..", "--", "-.",
"---", "-.-.", "-.-.", "-.", "...", "-", "-.-",
"....", "-.-", "-.-.", "-.-.", "-.-."};

private static final String[] DIGIT = new String[]

{"-----", ".----", "..---", "...--", "....-", ".....",
"-....", "-...-", "---..", "----."};

public static String[] getCode(String string){

string = string.trim();

int n = 1, lastIndex;

int index = string.indexOf(" ");

while (index > 0){

if (string.charAt(index+1) != ' ') n++;

index = string.indexOf(" ", index+1);

}

String[] result = new String[n];

int i = 0;

```

while (i < n){
    lastIndex = string.indexOf(" ", 1);
    if (lastIndex == -1){
        result[i++] = string.substring(0, string.length());
    } else {
        result[i++] = string.substring(0, lastIndex);
        string = string.substring(lastIndex, string.length());
        string = string.trim();
    }
}
for (i = 0; i < n; i++)
    result[i] = getCodeString(result[i]);
return result;
}

```

```

private static String getCodeString(String string){
    StringBuilder line = new StringBuilder();
    for (int i = 0; i < string.length(); i++){
        line.append(charToCode(string.substring(i,i+1).toUpperCase().charAt(0)));
    }
    return line.toString();
}

```

```

private static String charToCode(char c){
    int i = -1, j = -1;
    switch (c) {
        case 'A':
            i = 0;
            break;
        case 'B':
            i = 1;

```

```
        break;
    case 'C':
        i = 2;
        break;
    case 'D':
        i = 3;
        break;
    case 'E':
        i = 4;
        break;
    case 'F':
        i = 5;
        break;
    case 'G':
        i = 6;
        break;
    case 'H':
        i = 7;
        break;
    case 'I':
        i = 8;
        break;
    case 'J':
        i = 9;
        break;
    case 'K':
        i = 10;
        break;
    case 'L':
        i = 11;
        break;
```

case 'M':

 i = 12;

 break;

case 'N':

 i = 13;

 break;

case 'O':

 i = 14;

 break;

case 'P':

 i = 15;

 break;

case 'Q':

 i = 16;

 break;

case 'R':

 i = 17;

 break;

case 'S':

 i = 18;

 break;

case 'T':

 i = 19;

 break;

case 'U':

 i = 20;

 break;

case 'V':

 i = 21;

 break;

case 'W':

```
    i = 22;

    break;
case 'X':

    i = 23;

    break;
case 'Y':

    i = 24;

    break;
case 'Z':

    i = 25;

    break;
case '0':

    j = 0;

    break;
case '1':

    j = 1;

    break;
case '2':

    j = 2;

    break;
case '3':

    j = 3;

    break;
case '4':

    j = 4;

    break;
case '5':

    j = 5;

    break;
case '6':

    j = 6;
```

```
        break;
    case '7':
        j = 7;
        break;
    case '8':
        j = 8;
        break;
    case '9':
        j = 9;
        break;
    default:
        i = -1;
        j = -1;
        break;
}
if (i > -1)
    return LETTERS[i];
else if (j > -1)
    return DIGIT[j];
else
    return "";
}
}
```