PRACTICAL – 2

AIM: Java program that converts a string entered by the user to Morsecode or vice versa. It will require the implementation of data structures, including arrays, loops, and conditional statements.

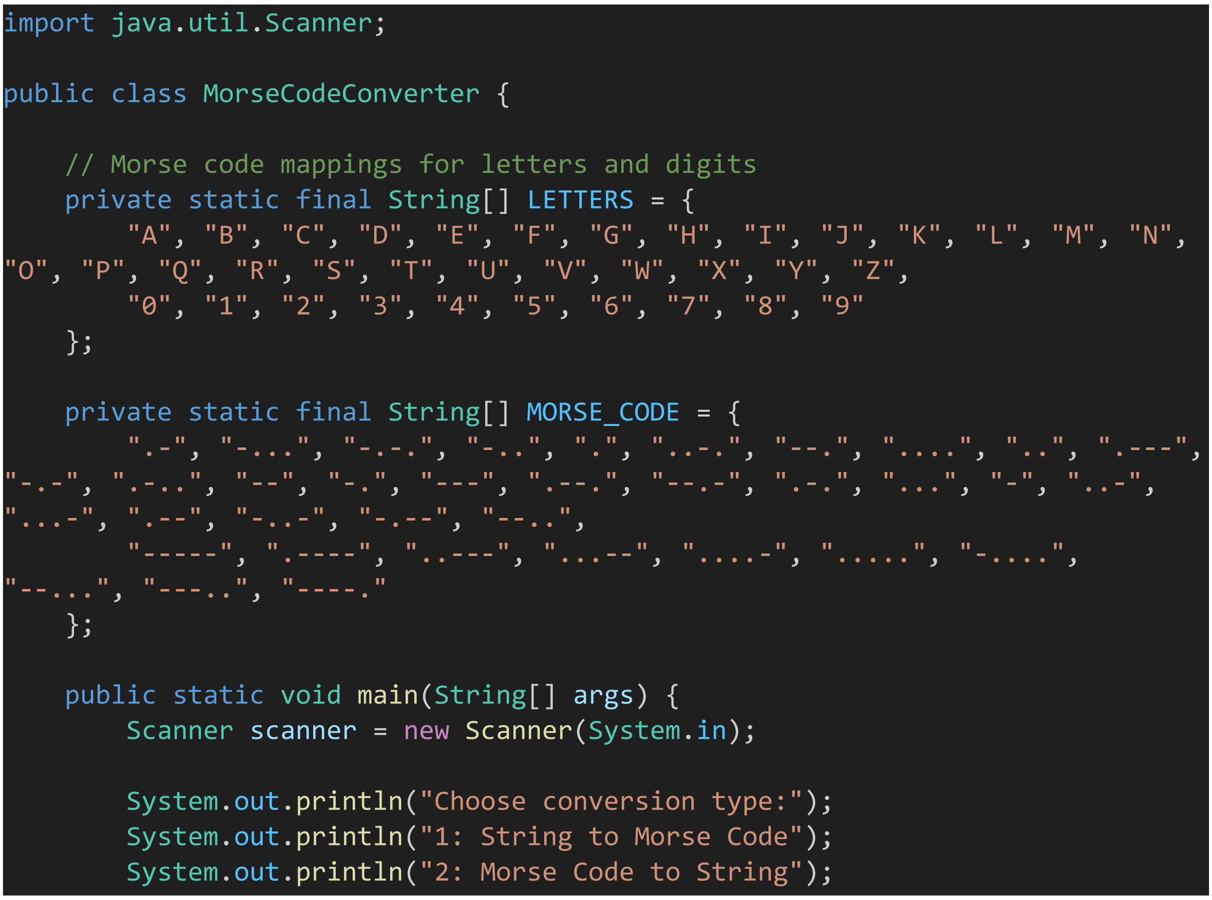
• Create two arrays - one to contain the strings of letters to be converted, and one to contain the Morse codes.

• In the program's main method, prompt the user for input to choose between the string or Morse.

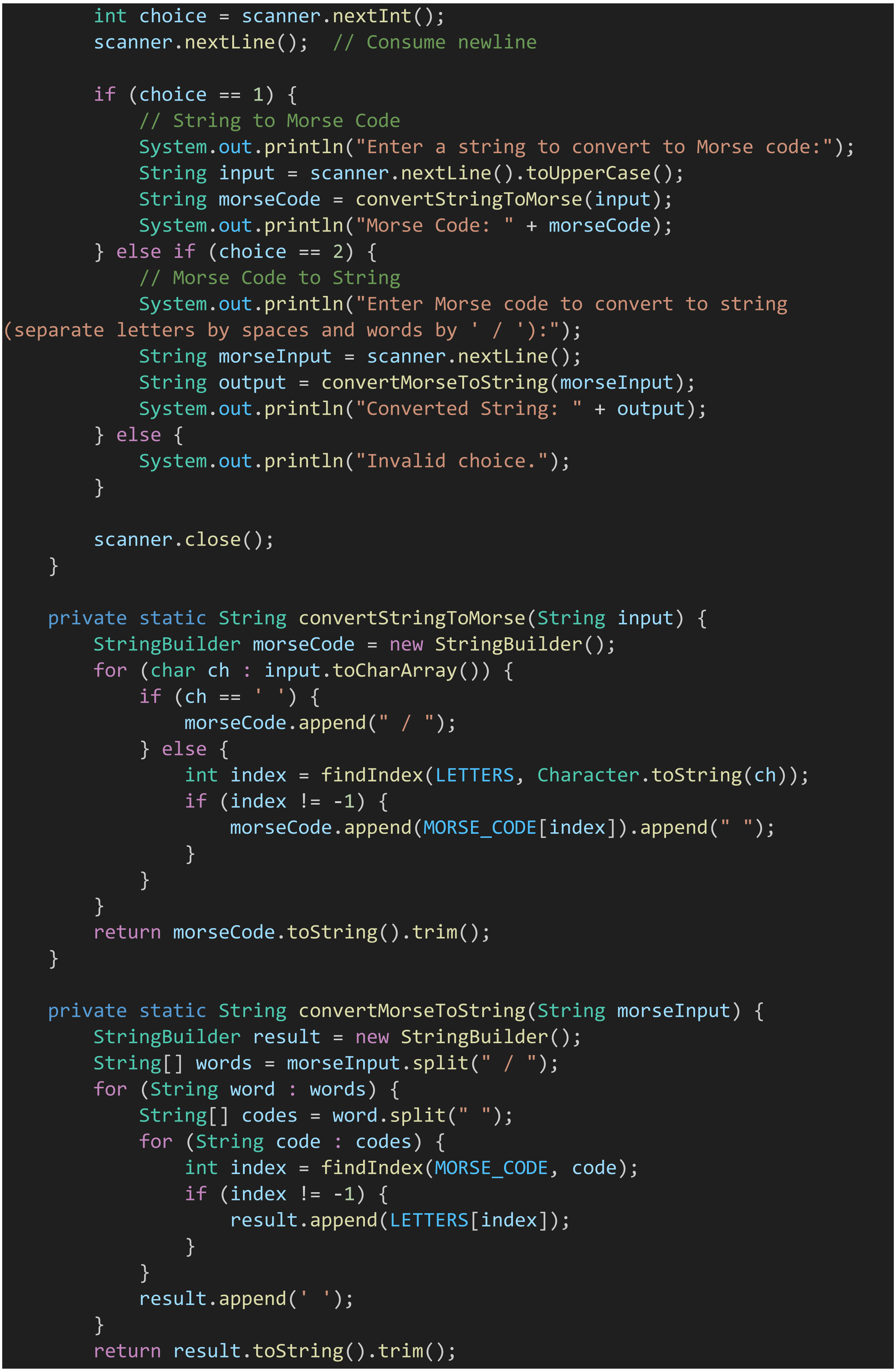
• For Morse code conversion, read in a string from the user; use conditional statements, looping, and array methods to convert the string to Morse-code.

• For string conversion, read in a Morse-coded string from the user; use arrays, conditional statements, and looping to convert Morse code to a string

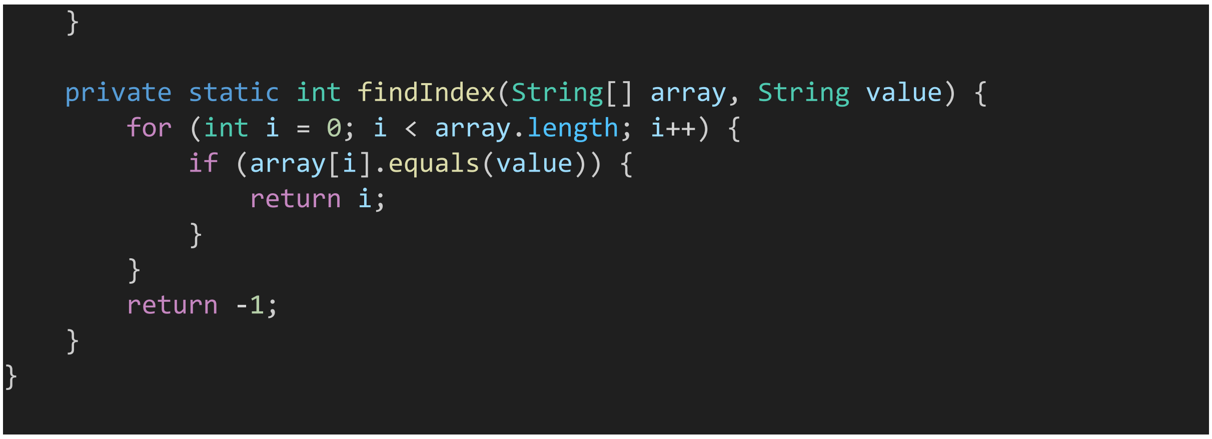
CODE:



```java
import java.util.Scanner;

public class MorseCodeConverter {

    // Morse code mappings for letters and digits
    private static final String[] LETTERS = {
        "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N",
"O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z",
        "0", "1", "2", "3", "4", "5", "6", "7", "8", "9"
    };

    private static final String[] MORSE_CODE = {
        ".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..", ".---",
"-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-", "..-",
"...-", ".--", "-..-", "-.--", "--..",
        "-----", ".----", "..---", "...--", "....-", ".....", "-....",
"--...", "---..", "----."
    };

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Choose conversion type:");
        System.out.println("1: String to Morse Code");
        System.out.println("2: Morse Code to String");
```

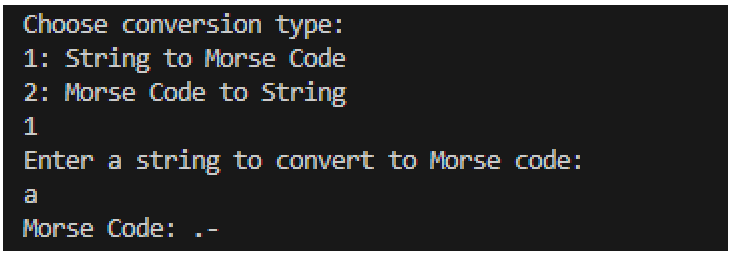


```java
        int choice = scanner.nextInt();
        scanner.nextLine();  // Consume newline

        if (choice == 1) {
            // String to Morse Code
            System.out.println("Enter a string to convert to Morse code:");
            String input = scanner.nextLine().toUpperCase();
            String morseCode = convertStringToMorse(input);
            System.out.println("Morse Code: " + morseCode);
        } else if (choice == 2) {
            // Morse Code to String
            System.out.println("Enter Morse code to convert to string
(separate letters by spaces and words by ' / '):");
            String morseInput = scanner.nextLine();
            String output = convertMorseToString(morseInput);
            System.out.println("Converted String: " + output);
        } else {
            System.out.println("Invalid choice.");
        }

        scanner.close();
    }

    private static String convertStringToMorse(String input) {
        StringBuilder morseCode = new StringBuilder();
        for (char ch : input.toCharArray()) {
            if (ch == ' ') {
                morseCode.append(" / ");
            } else {
                int index = findIndex(LETTERS, Character.toString(ch));
                if (index != -1) {
                    morseCode.append(MORSE_CODE[index]).append(" ");
                }
            }
        }
        return morseCode.toString().trim();
    }

    private static String convertMorseToString(String morseInput) {
        StringBuilder result = new StringBuilder();
        String[] words = morseInput.split(" / ");
        for (String word : words) {
            String[] codes = word.split(" ");
            for (String code : codes) {
                int index = findIndex(MORSE_CODE, code);
                if (index != -1) {
                    result.append(LETTERS[index]);
                }
            }
            result.append(' ');
        }
        return result.toString().trim();
```

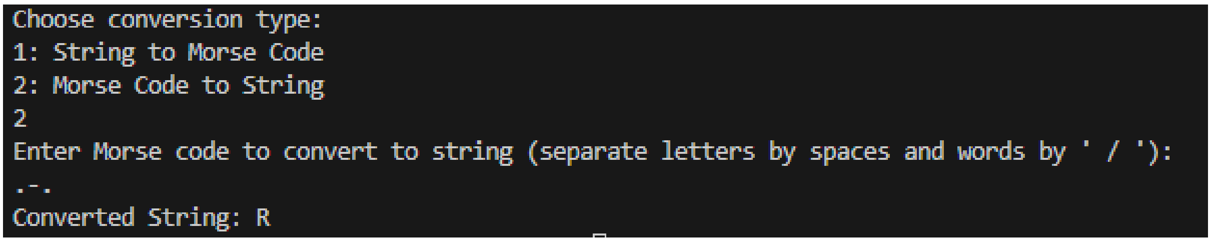


```java
    }

    private static int findIndex(String[] array, String value) {
        for (int i = 0; i < array.length; i++) {
            if (array[i].equals(value)) {
                return i;
            }
        }
        return -1;
    }
}
```

OUTPUT:



```
Choose conversion type:
1: String to Morse Code
2: Morse Code to String
1
Enter a string to convert to Morse code:
a
Morse Code: .-
```



```
Choose conversion type:
1: String to Morse Code
2: Morse Code to String
2
Enter Morse code to convert to string (separate letters by spaces and words by ' / '):
.-.
Converted String: R
```

CONCLUSION:

The `MorseCodeConverter` program showcases basic Java data structures and control flow for string manipulations. It uses arrays to map characters to Morse code and vice versa, employs loops and conditionals for processing, and handles user input for conversion tasks. This robust implementation demonstrates key Java concepts effectively and could be enhanced with features like punctuation handling or error checking.