Yesreshe bade

Main.java

```java
import java.util.*;
public class Main {

    public static void MorseCodeConverter(String[] code,
                                           String englishLang,
                                           char[] letter)
    {
        System.out.print("");
        for (int i = 0; i < englishLang.length(); i++) {
            for (int j = 0; j < letter.length; j++) {
                if (englishLang.charAt(i) == letter[j]) {
                    System.out.print(code[j] + " ");
                    break;
                }
            }
        }
    }

    public static void main(String[] args)
    {
  Scanner s=new Scanner(System.in);
        // store the all the alphabet in an array
        char[] letter = { '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', ' ' };
        // Morse code by indexing
        String[] code
            = { ".-",   "-...", "-.-.", "-..",  ".",
                "..-.", "--.",  "....", "..",   ".---",
                "-.-",  ".-..", "--",   "-.",   "---",
                ".--.", "--.-", ".-.",  "...",  "-",
                "..-",  "...-", ".--",  "-..-", "-.--",
                "--..", "/" };

        // Given Strings

        String englishLang;
        System.out.println("enter the String");
        englishLang=s.nextLine();
        System.out.println();
        // English to morse code
        MorseCodeConverter(code, englishLang, letter);
    }
}
```

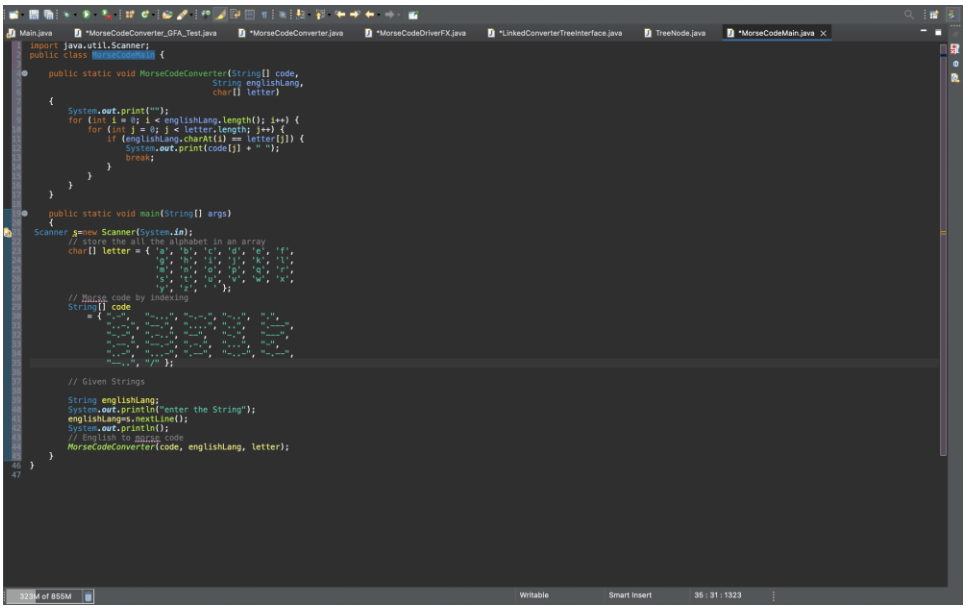
```java
import java.io.File;
import java.io.FileNotFoundException;

import javax.swing.JFileChooser;
import javax.swing.JOptionPane;


public class MorseCodeConverter_GFA_Test {


    public void setUp() throws Exception {
    }


    public void tearDown() throws Exception {
    }


    public void testConvertToEnglishString() {
        String converter1 = MorseCodeConverter.convertToEnglish(".... . .-.. .-.. --- / .-- --- .-. .-.. -.. ");
        assertEquals("hello world",converter1);
    }


    private void assertEquals(String string, String converter1) {
        // TODO Auto-generated method stub

    }

}
```

```java
public class MorseCodeConverter {

    public static String convertToEnglish(String string) {
        // TODO Auto-generated method stub
        return null;
    }

}
```

```java
import javax.swing.JFrame;

public class MorseCodeDriverFX extends Application{


    public static void main(String[] args){
        launch(args);
    }

    public void start(Stage stage)
    {

        //call the main scene which is a BorderPane
        MorseCodeMain mainPane = new MorseCodeMain();
        //PasswordMain root = mainPane.getTopContainer();
        Scene scene = new Scene(mainPane, 550, 350);
        stage.setScene(scene);
        stage.setTitle("Morse Code Converter Driver");
        stage.show();
    }

}
```

```java
import javax.swing.JFrame;

public class MorseCodeDriverFX extends Application{


    public static void main(String[] args){
        launch(args);
    }

    public void start(Stage stage)
    {

        //call the main scene which is a BorderPane
        MorseCodeMain mainPane = new MorseCodeMain();
        //PasswordMain root = mainPane.getTopContainer();
        Scene scene = new Scene(mainPane, 550, 350);
        stage.setScene(scene);
        stage.setTitle("Morse Code Converter Driver");
        stage.show();
    }

}
```

```java
import java.util.ArrayList;
public interface LinkedConverterTreeInterface<T> {

    public TreeNode<T> getRoot();

    public void setRoot(TreeNode<T> newNode);


    public LinkedConverterTreeInterface<T> insert(T code, T result);


    public void addNode(TreeNode<T> root, T code, T letter);


    public T fetch(String code);


    public T fetchNode(TreeNode<T> root, T code);


    public LinkedConverterTreeInterface<T> delete(T data) throws UnsupportedOperationException;


    public LinkedConverterTreeInterface<T> update() throws UnsupportedOperationException;


    public void buildTree();


    public ArrayList<T> toArrayList();


    public void LNRoutputTraversal(TreeNode<T> root, ArrayList<T> list);


}
```



Lesson learned:

These codes are composed of just dots and dashes.