Programming 3(Course code: BMEVIIIAB00)

Academic year: 2021/22/1

Final Project: Morse Converter Java Program

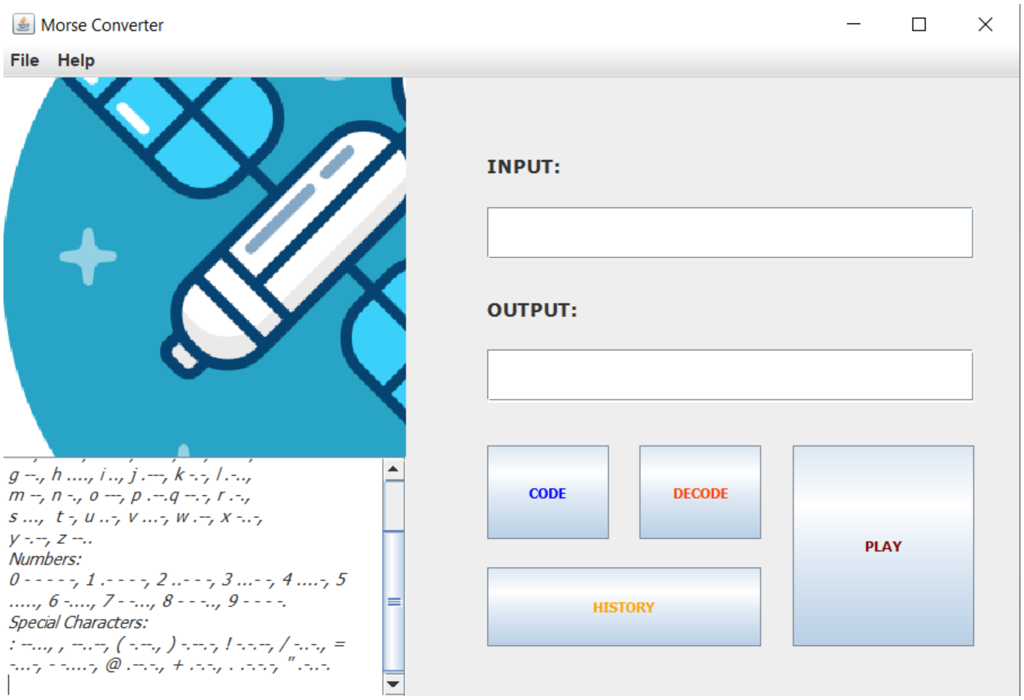
Name: Tushig Bat-Erdene

Neptun ID: QBI3JH

2nd of December 2021

Introduction:

Morse code is mainly used in telecommunication to encode text characters by standardized sequences of different signal durations (as known as dots and dashes). In this paper, I will be explaining the main functionality of this program and describing my solution to understand how exactly it works.



```
g --., h ...., i .., j .---, k -.-, l .-..,
m --, n -., o ---, p .--.q --.-, r .-.,
s ...,  t -, u ..-, v ...-, w .--, x -..-,
y -.--, z --..
Numbers:
0 - - - - -, 1 .- - - -, 2 ..- - -, 3 ...- -, 4 ....-, 5
....., 6 -...., 7 - -..., 8 - - -.., 9 - - - -.
Special Characters:
: ---..., , --..--, ( -.--., ) -.--.-, ! -.-.--, / -..-., =
-...-, - -....-, @ .--.-., + .-.-., . .-.-.-, " .-..-.
```

Figure 1. User Interface of the program

According to the Figure 1, It displays the main user interface of the program. Following operations will be the functionalities of it:

- Input:
  This text field gets the text input from the user. It only read lowercase characters.
- Output:
  This text field displays the output of conversions based on the input from the input field.
- Code:
  This button has the functionality which is making it possible to convert user input and displays the conversion in the output field. If the user input is blank or empty, it throws an invalid message.
- Decode:
  This button has the functionality which is making it possible to convert user input and displays the conversion in the output field. If the user input is blank or empty, it throws an invalid message.
- History:
  This button displays the previous conversion from the “history.txt” file.
- Play:

Play button plays the morse sound from reading dashSound and dotSound files.

- Text area contains the dictionary or translation of characters and Morse code.

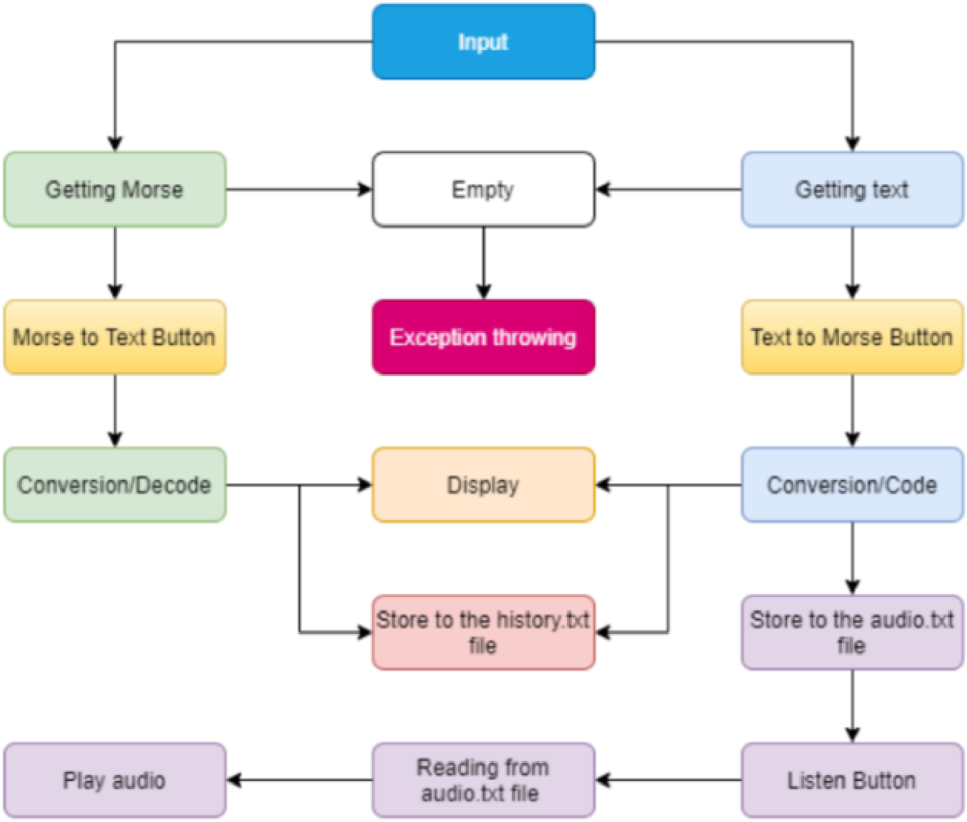


Figure 2. Flow Chart of The Program

Data structure of the program:

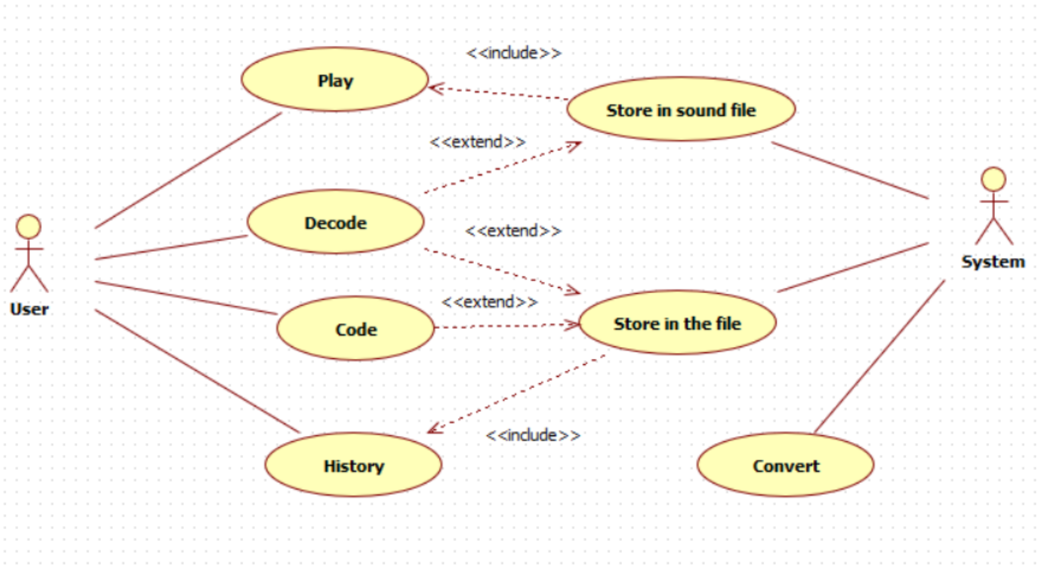
I used linked list and binary tree for my converter because it is the simplest way to represent morse characters by connection of its nodes. It has at most 2 parent nodes "left, right" and it is making it possible to initialize morse library easier. As you can see from the following figure, I made left subtree of a nodes are the morse codes which are starting with dashes and right subtree of a nodes are starting with dots. For instance, finding "g" is Start → left → left → right.

My solution includes:

- Swing based GUI
  - I mostly used JFrame on the windows I made.
- Collection Framework
  - I used ArrayList to create Morse library.
- File input, output using serialization
  - It is realized with history class and sound classes.
- Junit Testing
  - Testing for the essential functions.

2 USE CASE DIAGRAM

Use Case Diagram:



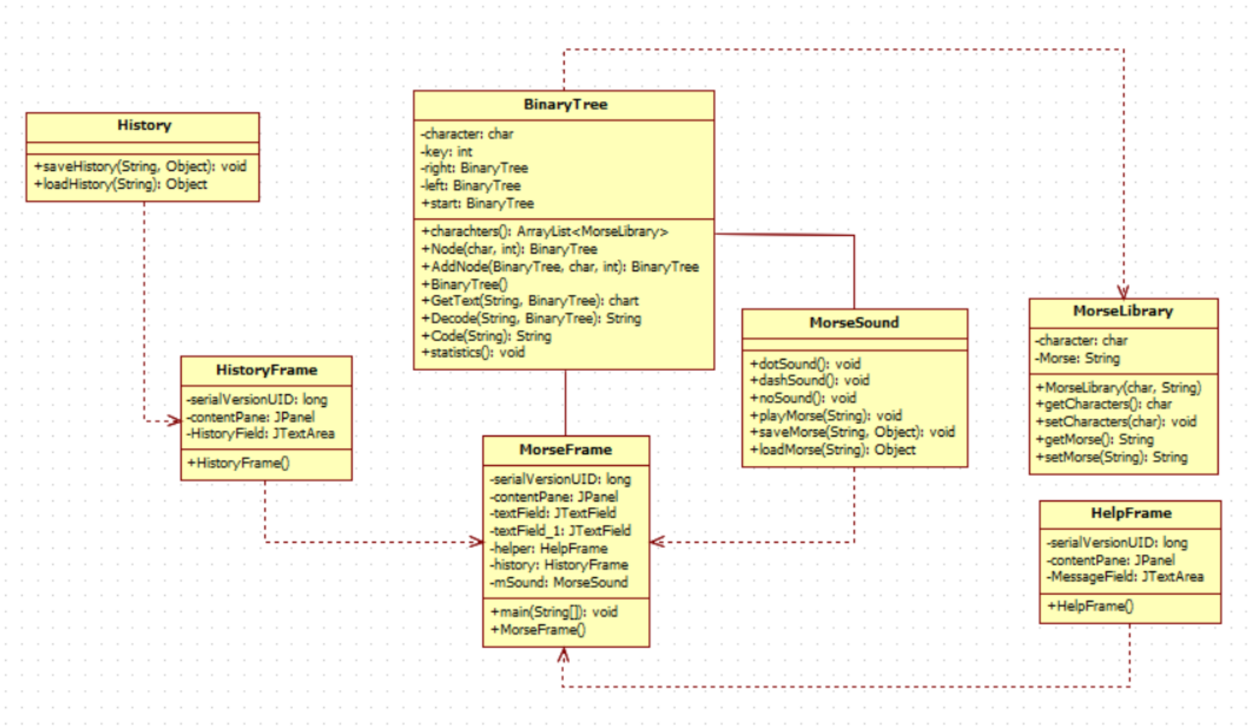
Title	Code
Description	Converts user input to morse code.
Actors	User
Main success scenario	Displays the morse code.
Alternate scenario	Displays invalid if the text field is empty.

Title	Decode
Description	Converts morse code to text.
Actors	User
Main success scenario	Displays text.
Alternate scenario	Displays invalid if the text field is empty.

Title	History
Description	Shows the converted code or text.
Actors	User
Main success scenario	Displays the previous conversion in the new frame.
Alternate scenario	Throws exception.

Title	Play
Description	It plays the sound of morse code.
Actors	User
Main success scenario	Hearing the sound.
Alternate scenario	Throws exception.

3 CLASS DIAGRAM



3.1.1 BinaryTree

Responsibilities

BinaryTree is the main class to convert text input to Morse code and vise versa.

Attributes

-character: char	Declaring character.
-key: int	Declaring int.
-right: BinaryTree	Declaring BinaryTree left.
-left: BinaryTree	Declaring BinaryTree right.
-start: BinaryTree	Declaring BinaryTree start is null.

Methods

+characters(): ArrayList<MorseLibrary>	Initializing Morse Library. MorseLibrary has all of the characters and its translations in Morse code.
+Node(char, int): BinaryTree	This pointer function is for putting characters in the BinaryTree by creating nodes for each character.
+BinaryTree()	Constructor
+GetText(String, BinaryTree): char	This function has a role to find characters. Function checks whether it is left or right to find dot or dash. If user inserts the morse code which is invalid or do not belong in the library it will say invalid.
+Decode(String, BinaryTree): String	It is the most essential functions for converting morse into the text.
+Code(String): String	It is the most essential functions for converting text into the Morse code.

+statistics(): void	Prints the dictionary or all of the statistics.

3.1.2 MorseLibrary

Responsibilities

MorseLibrary is the class for initializing the library which has characters and Morse code in it.

Attributes

-character: char	Declaring character.
-Morse: String	Declaring Morse code.

Methods

+MorseLibrary(char, String)	Constructor which returns char and String.
+getCharacters(): char	It will get character.
+setCharacters(char): void	It will set character.
+getMorse(): String	It will get Morse string.
+setMorse(String): String	It will set Morse string.

3.1.3 MorseFrame

Responsibilities

The main class with graphical user interface made with Swing.

Attributes

-serialVersionUID: long	Declaring default serial version ID.
-contentPane: JPanel	Initializing JPanel to make left side of the user interface.
-textField: JTextField	Initializing text field for the input.
textField_1: JTextField	Initializing another text field for the output.
-helper: HelpFrame	Declaring helper to call HelpFrame class which is needed to show window when user clicks on help menu.
-history: HistoryFrame	Declaring history to call HistoryFrame class which is needed to show window when user clicks on history button.
-mSound: MorseSound	Declaring mSound to call MorseSound.

Methods

+main(String[]): void	Main method is for setting the window or frame visible.
+MorseFrame()	MorseFrame does every operation in the main window.

3.1.4 MorseSound

Responsibilities

MorseSound class is for storing the converted text or morse code in the sound file and reads from the file to play sound.

Attributes

<attribute>	<description>

Methods

+dotSound(): void	This function is for getting the sound of dot by reading from the file.

+dashSound(): void	This function is for getting the sound of dash by reading from the file.
+noSound(): void	This function is for getting the sound of silence by reading from the file.
+playMorse(String): void	This function finds dot or dash to play the sound.
+saveMorse(String, Object): void	It is for storing morse code into the different file from history.
+loadMorse(String): Object	It is for loading the morse code from saved file.

3.1.5 History

Responsibilities

History class serializes and deserializes the output.

Attributes

<attribute>	<description>

Methods

+saveHistory(String, Object): void	It is for storing conversions into the history.txt file.
+loadHistory(String): Object	It is for loading the conversions from history.txt file.

3.1.6 HistoryFrame

Responsibilities

This class is responsible for showing the serialized conversions by deserializing them from the file and setting it on the history window.

Attributes

-serialVersionUID: long	Declaring default serial version ID.
-contentPane:JPanel	Initializing JPanel to make the user interface.
-HIstoryField:JTextArea	Initializing text area to set deserialized history.txt files contents in it.

Methods

+HistoryFrame()	Creating history frame/window.

3.1.7 HelpFrame

Responsibilities

<responsibilities of the class>

Attributes

-serialVersionUID: long	Declaring default serial version ID.
-contentPane: JPanel	Initializing JPanel to make the user interface.
-MessageField: JTextArea	Initializing text area to show the message.

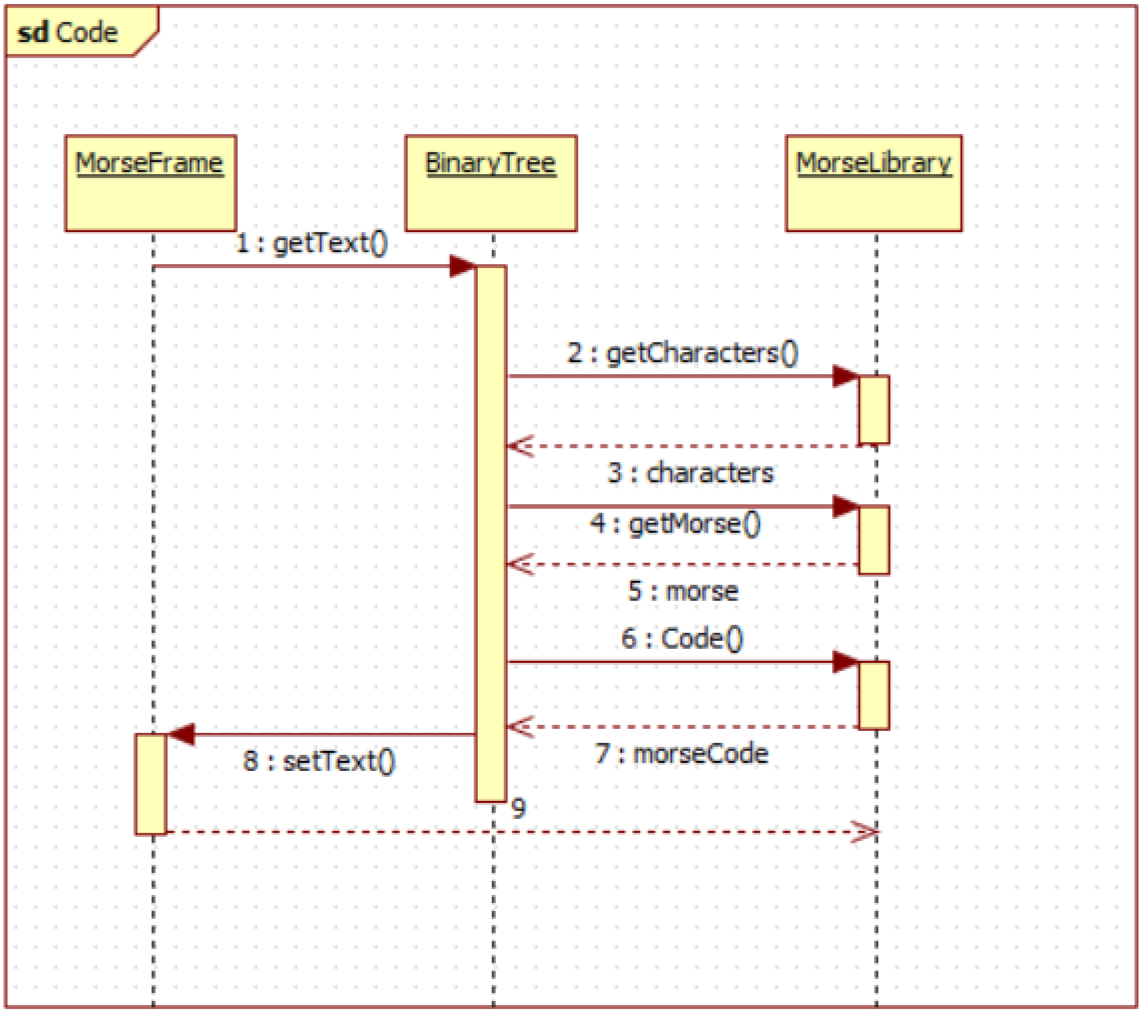
Methods

+HelpFrame()	Creating the complete window for helper.

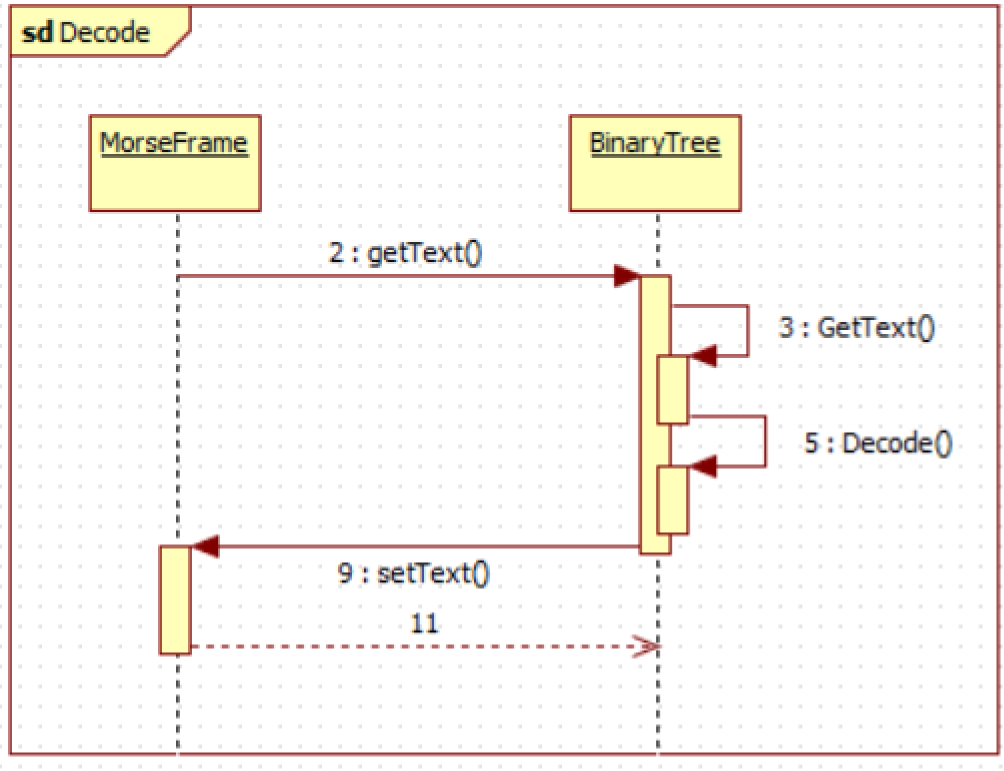
4.1 SEQUENCE DIAGRAMS

I made the essential sequence diagrams of the most important operations such as coding text to morse, decoding morse code to text, storing text conversion in the history, storing morse conversion in the history, and playing the morse code.

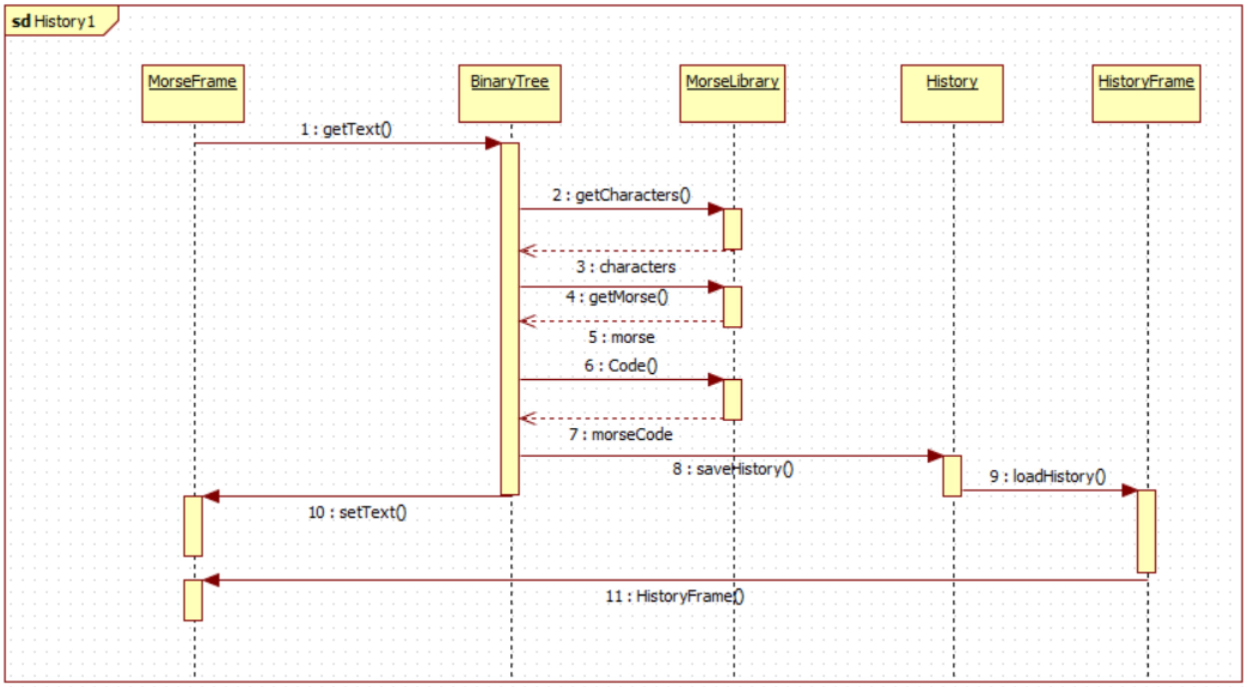
4.1.1 Code



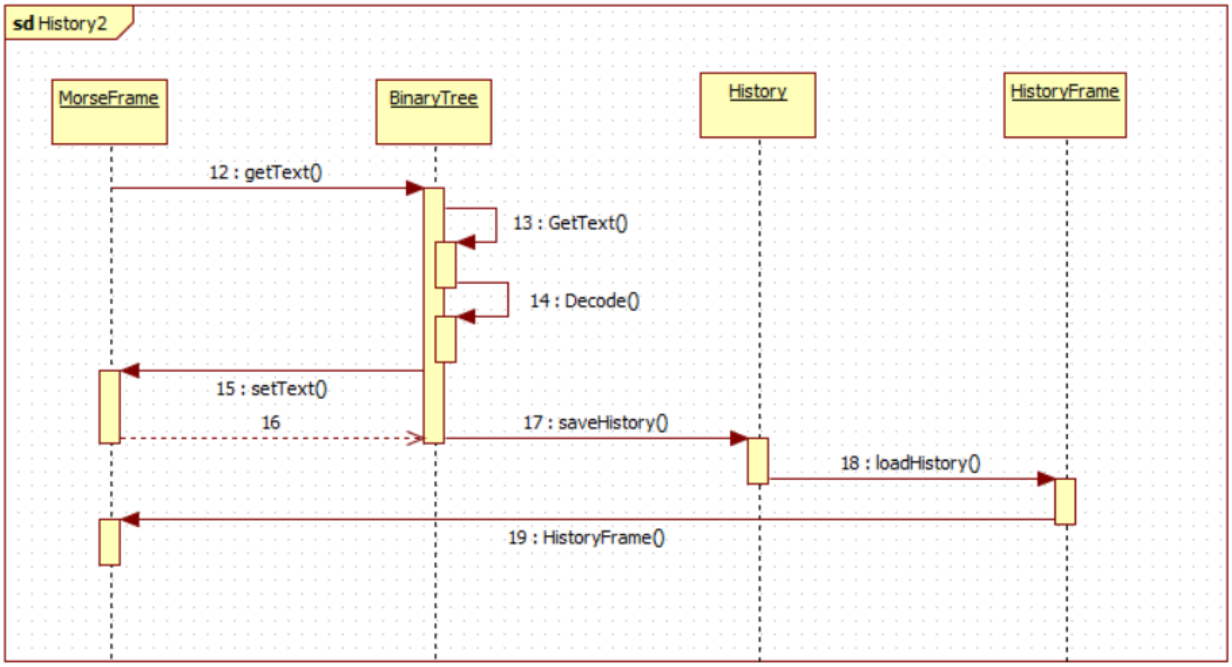
4.1.2 Decode



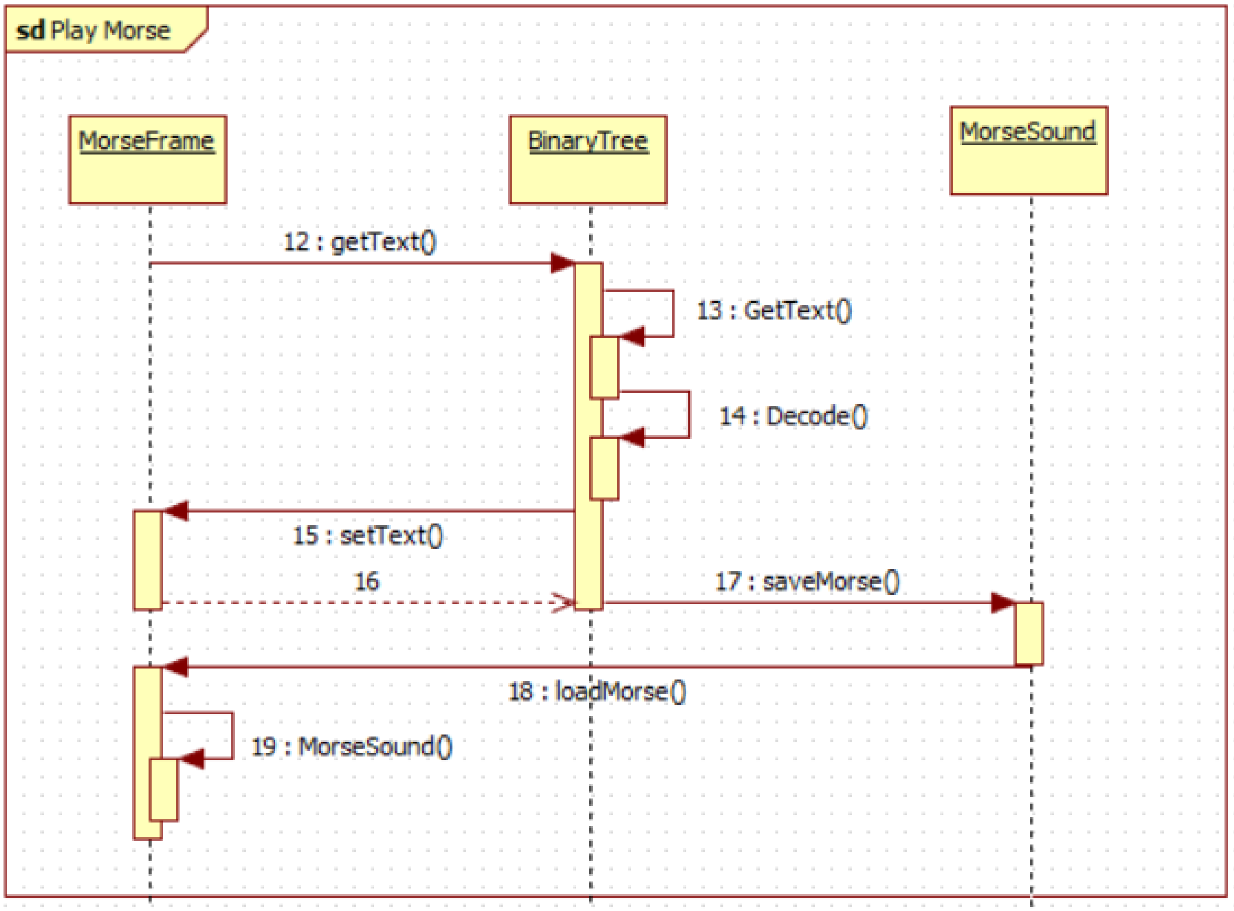
4.1.3 History 1



4.1.4 History 2



4.1.5 Playing Morse code



5 JUNIT TEST DESCRIPTION

5.1 TEST OF BINARY TREE CLASS

@Test `public void testCode()`	
Description	Tests the method to make sure it converts the text string to morse code.
Main success scenario	Test passes if the convertion function matches the expected result or string and it means the Code function is converting the strings.
Alternate scenario	Test fails if the converted value doesn't match the expected result. This could happen if: - BinaryTree was wrong. - getCharacter() from MorseLibrary was wrong. - getMorse() from MorseLibrary was wrong.

@Test `public void testDecode()`	
Description	Tests the method to make sure it converts the morse code to the text string.
Main success scenario	Test passes if the convertion function matches the expected result or string and it means the Code function is converting the strings.
Alternate scenario	Test fails if the converted value doesn't match the expected result. This could happen if: - BinaryTree was wrong. - getText() was wrong. - Start root is empty.

5.2 TEST OF MORSE SOUND CLASS

@Test `public void testSound()`	
Description	Tests the method to make sure playMorse finds dot, dash, or empty.
Main success scenario	Test passes if the function finds the expected result it should play the sound of dot or dash.
Alternate scenario	Test fails if the converted value doesn't match the expected result. This could happen if: - playMorse couldn't find dot or dash. - Clip is not processing.

References:

[1] – https://javatpoint.com

[2] – https://www.iit.bme.hu/targyak/BMEVIIIAB00

[3] – https://morsecode.world/international/translator.html

[4] – https://en.wikipedia.org/wiki/Morse_code

[5] – https://www.codeproject.com/Questions/1210248/Play-wav-file-in-java

[6] – https://docs.oracle.com/javase/tutorial/