

---

# Text2SpeechEditor

## FINAL Report

---

Gang of Three (GoT)

Apostolos Vakalos, 3185

Anastasios Amperiadis, 3178

Vasileios Nasios, 3296

Ioannina

May 2020

---

## VERSIONS HISTORY

---

Date	Version	Description	Author
25/5/2020	1.0	Final report	GoT

## 1 Introduction

---

This document provides information concerning the final release of the project.

### 1.1 Purpose

---

The objective of this project is to develop an application that transforms text to audio. The user can provide text information by creating text files and use the application to hear them. This may help users with speaking problems to communicate.

### 1.2 Document Structure

---

The rest of this document is structured as follows. Section 2 describes out Scrum team and specifies the Project's backlog. Section 3 specifies the main design concepts for this release of the project.

## 2 Scrum team and Project Backlog

---

In this section the project's functional and non-functional details are described.

### 2.1 Scrum team

---

<b>Product Owner</b>	Apostolos Zarras
<b>Scrum Master</b>	GoT
<b>Development Team</b>	Apostolos Vakalos, Anastasios Amperiadis, Vasileios Nasios

## 2.2 Project Backlog

---

### Functional Requirements:

- 1) The user can create a new empty document, by giving its title, author and choosing the location where it will be saved. The application keeps the creation date and time to the document properties.
- 2) The user can edit the contents of the document via the text area of the application.
- 3) The user can save the contents of the document to disk by giving a preferred filename and the application will add to the document's properties the last save date and time.
- 4) The user can open an existing document of the disk for viewing or updating the contents by browsing the file system folders.
- 5) The user can transform the contents of the document to audio.
- 6) The user can transform a selected line of the document to audio.
- 7) The user can play the document in reverse. Specifically, the last word will be played first and so on.
- 8) The user can play a selected line in reverse as described above.
- 9) The user can encode the contents of the document and transform them to speech.
- 10) The user can encode a selected line and transform it to speech.
- 11) The application provides to user the option between two encoding techniques:
  - AtBash
  - Rot-13
- 12) The user can tune the following audio parameters:
  - Volume
  - Speech Rate
  - Pitch
  - Range
- 13) The user can re-execute multiple times the sequence of actions that performed from a specific start until stop point.

The Non-Functional Requirements are described below:

- 1) **Extensibility:** The application is structured with design patterns that allow to change the FreeTTS library with another one or add another encoding strategy with no significant changes.
- 2) **Usability:** The application is designed with simple menus, slide bars for tuning the audio parameters and a help menu that describes the functionality of the application.
- 3) **Performance:** The application transforms text to audio in the minimum of time.

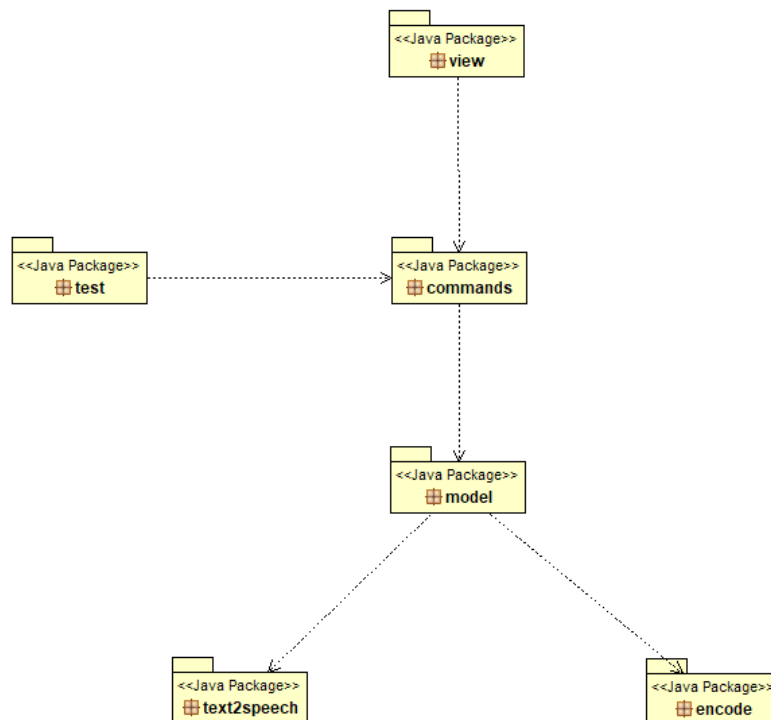
## 3 Design

---

### 3.1 Architecture

---

The package diagram below defines the overall architecture of the application.



A brief description of each package:

**View:** Includes the graphical user interface of the application and all the classes for viewing information to user. (See class analysis for each package below)

**Commands:** Includes the commands for each user's action implementing the main functionality of the application.

**Model:** Includes the two main entities of the application. A Document and a Line class.

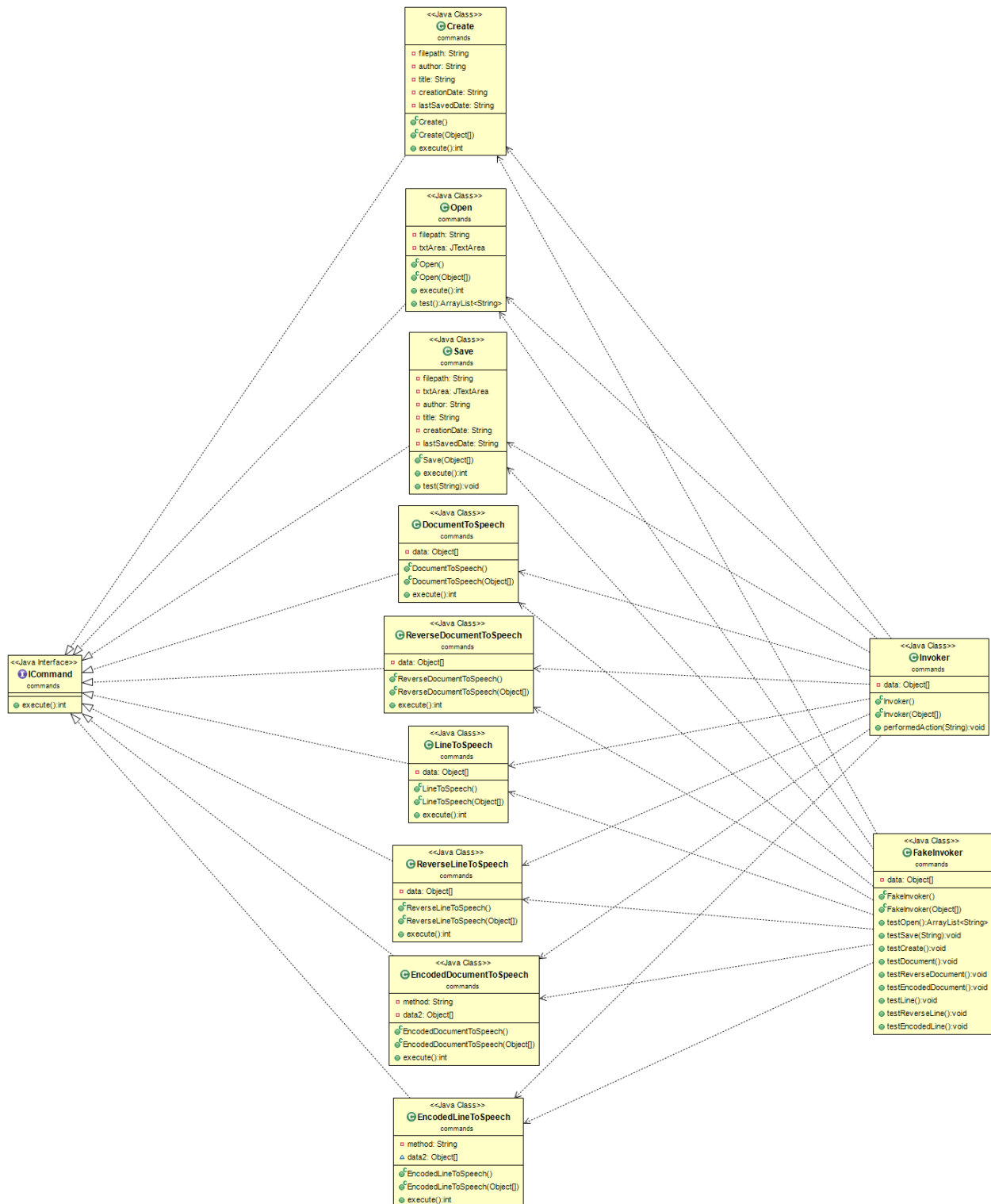
**Encode:** Includes the implementation of all encoding strategies.

**Text2speech:** Includes the interconnection between application and FreeTTS library.

**Test:** Includes all tests for the application.

## 3.2 Design

### Package Commands:



Class Name: Invoker	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Create the Command objects for each action of the user</li> <li>▪ Provides all the constructor arguments for each Command object from GUI</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ With all Command classes due to object creation</li> </ul>

Class Name: Create	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Create a file to disk</li> <li>▪ Write the properties at the beginning of the file</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Implements the ICommand interface method execute()</li> </ul>

Class Name: Open	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Reads the contexts of a specified file</li> <li>▪ Writes them to the text area of GUI</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Implements the ICommand interface method execute()</li> </ul>

Class Name: Save	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Reads the contexts of the document from the text area of GUI</li> <li>▪ Updates the properties with last save date and time</li> <li>▪ Updates the file on disk</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Implements the ICommand interface method execute()</li> </ul>



Class Name: DocumentToSpeech	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Create a new Document object to be transformed to speech</li> <li>▪ Pass the audio parameters from GUI to Document constructor</li> <li>▪ Play the transformed document</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Implements the ICommand interface method execute()</li> <li>▪ With Document class from package Model to use the playContexts() method</li> </ul>

Class Name: ReverseDocumentToSpeech	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Create a new Document object to be transformed to speech</li> <li>▪ Pass the audio parameters from GUI to Document constructor</li> <li>▪ Play the transformed document in reverse</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Implements the ICommand interface method execute()</li> <li>▪ With Document class from package Model to use the playReverse() method</li> </ul>

Class Name: LineToSpeech	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Create a new Line object to be transformed to speech</li> <li>▪ Pass the audio parameters from GUI to Line constructor</li> <li>▪ Play the transformed line</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Implements the ICommand interface method execute()</li> <li>▪ With Line class from package Model to use the playLine() method</li> </ul>

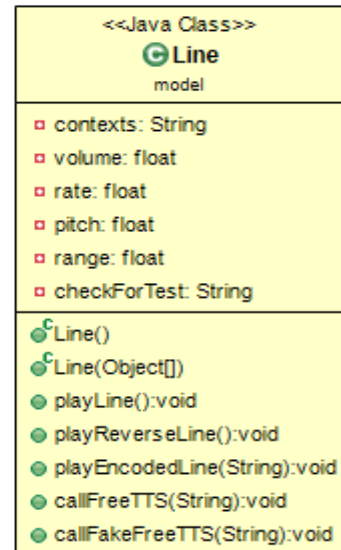
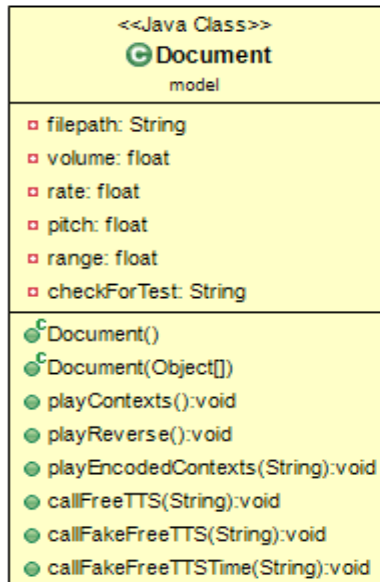
Class Name: ReverseLineToSpeech	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Create a new Line object to be transformed to speech</li> <li>▪ Pass the audio parameters from GUI to Line constructor</li> <li>▪ Play the transformed line in reverse</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Implements the ICommand interface method execute()</li> <li>▪ With Line class from package Model to use the playReverseLine() method</li> </ul>

Class Name: EncodedDocumentToSpeech	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Create a new Document object to be transformed to speech</li> <li>▪ Pass the audio parameters from GUI to Document constructor</li> <li>▪ Pass the selected encoding strategy from GUI to Document constructor</li> <li>▪ Play the encoded document</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Implements the ICommand interface method execute()</li> <li>▪ With Document class from package Model to use the playEncodedContexts() method</li> </ul>

Class Name: EncodedLineToSpeech	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Create a new Line object to be transformed to speech</li> <li>▪ Pass the audio parameters from GUI to Line constructor</li> <li>▪ Pass the selected encoding strategy from GUI to Line constructor</li> <li>▪ Play the encoded line</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ Implements the ICommand interface method execute()</li> <li>▪ With line class from package Model to use the playEncodedLine() method</li> </ul>

Class Name: FakeInvoker	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Used only for testing purposes</li> <li>▪ Simulates the execution of the real application by calling test methods of corresponding classes of real Invoker</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ With all command classes due to object creation</li> </ul>

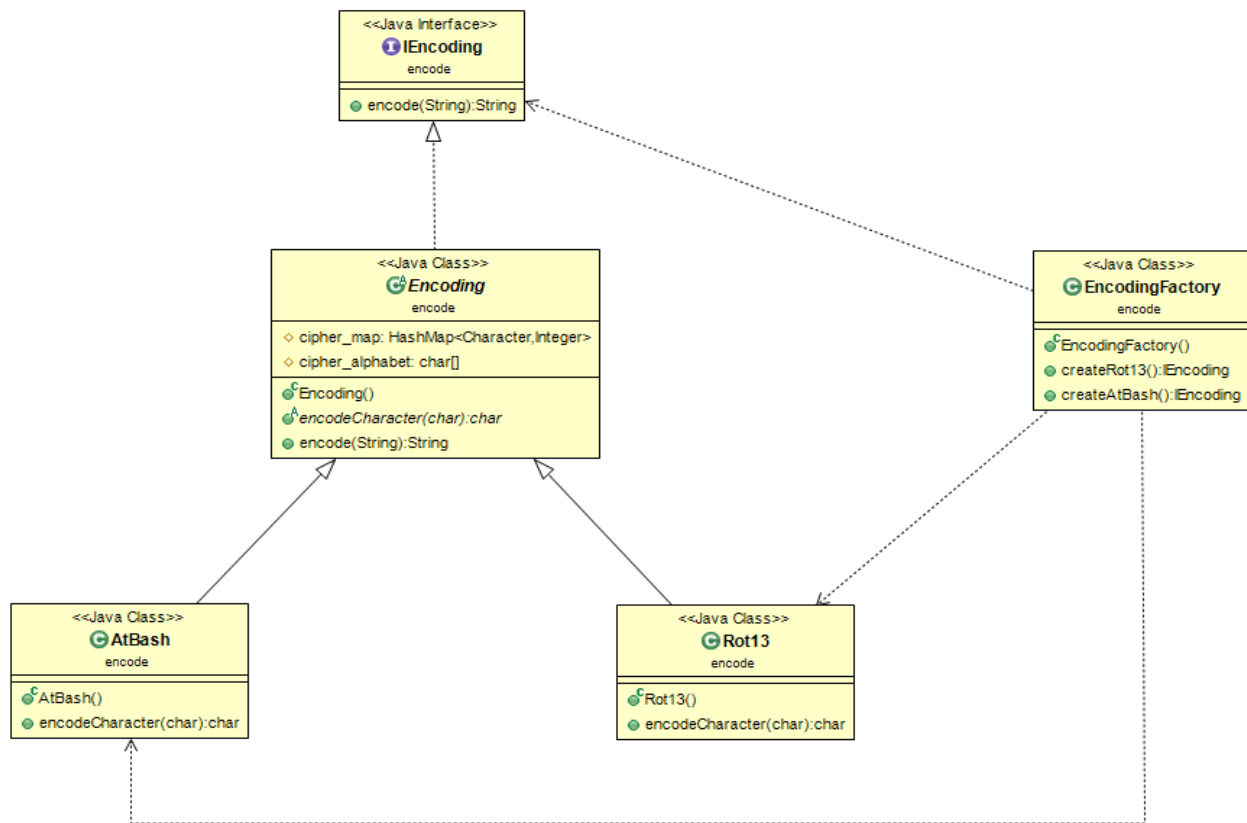
## Package Model:



Class Name: Document	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Implement all the actions for a document that transform it to audio (play the document, play it reverse, play it encoded...)</li> <li>Initialize a FreeTTS object with the audio parameters passed from Command objects</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>With IEncoding interface and EncodingFactory class to play an encoded document</li> <li>With FreeTTS class from package text2speech</li> </ul>

Class Name: Line	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Implement all the actions for a line that transform it to audio (play the line, play it reverse, play it encoded...)</li> <li>Initialize a FreeTTS object with the audio parameters passed from Command objects...</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>With IEncoding interface and EncodingFactory class to play an encoded document</li> <li>With FreeTTS class from package text2speech</li> </ul>

## Package Encode:



### Class Name: Encoding

#### Responsibilities:

- Provides the abstract method `encodeCharacter()` to subclasses of each different encoding strategy
- Implement the encoding of a document by using the `encodeCharacter()` method

#### Collaborations:

- Implements the `encode()` method of **IEncoding** interface

### Class Name: AtBash

#### Responsibilities:

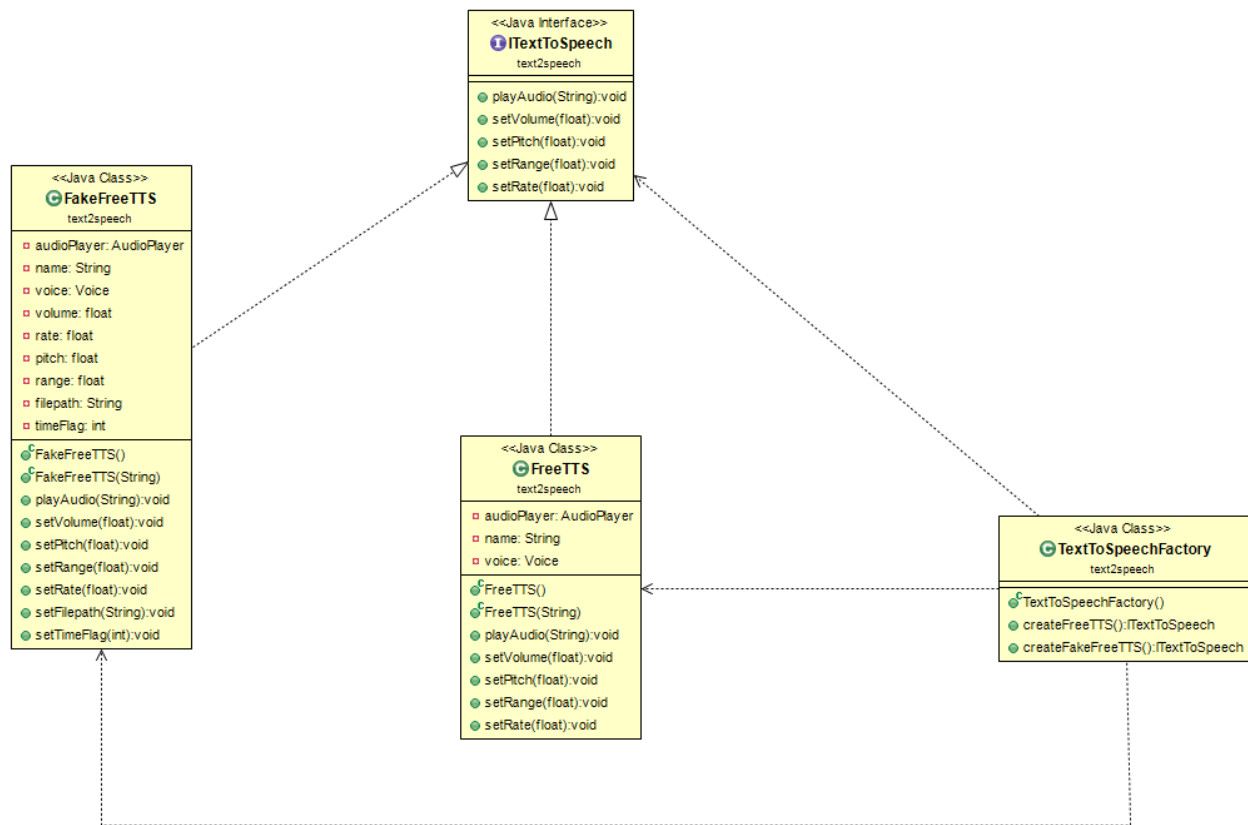
- Implements the **AtBash** encoding strategy method for one character

#### Collaborations:

- Extends the abstract class **Encoding**

<b>Class Name: Rot13</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Implements the Rot-13 encoding strategy method for one character</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>Extends the abstract class Encoding</li> </ul>

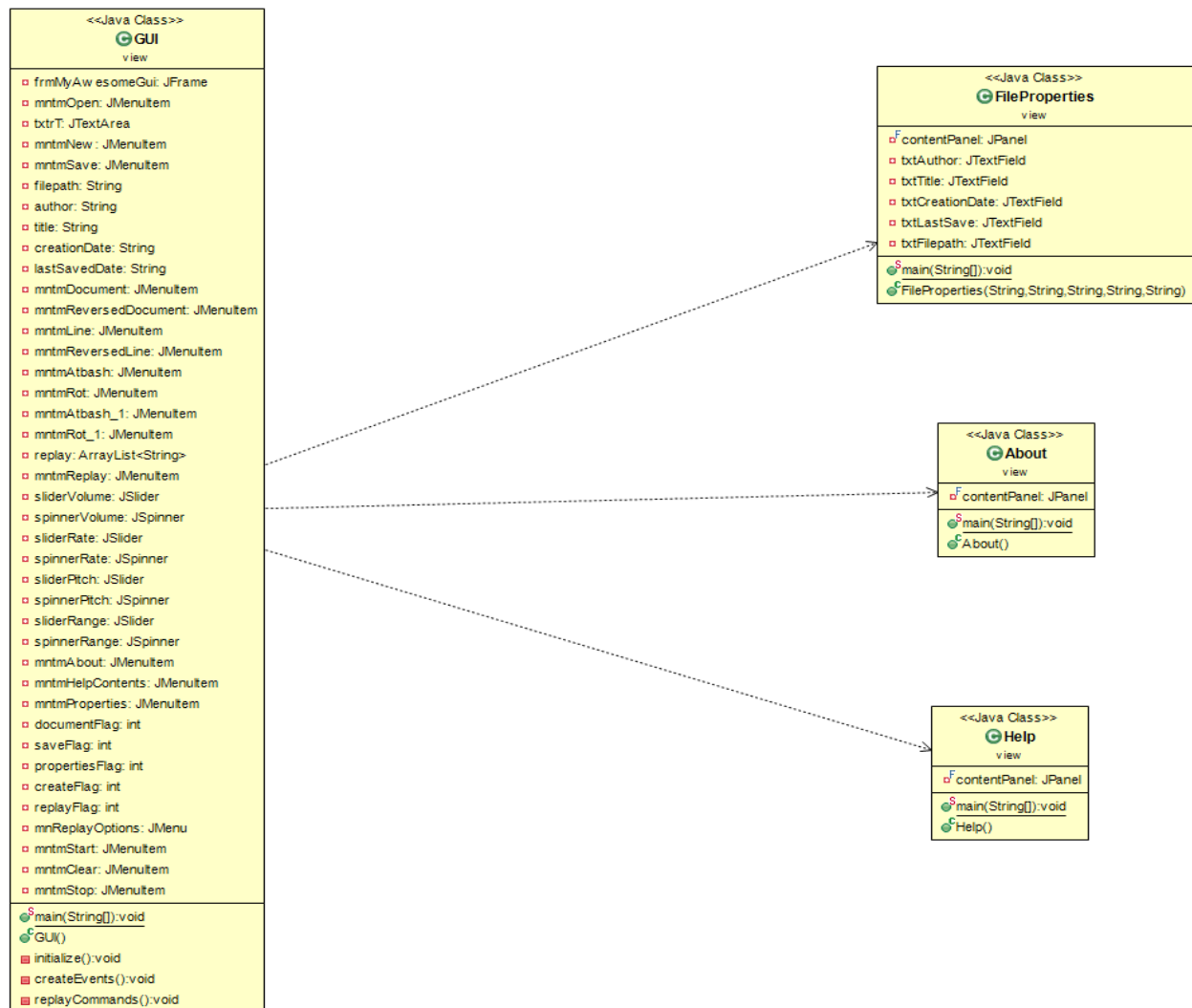
## Package Text2speech:



<b>Class Name: FreeTTS</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Initialize the object of FreeTTS library</li> <li>Calls the methods of external FreeTTS library to transform the text to audio (play audio, set parameters of audio)</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>Implements the play and set audio parameter methods of interface ITextToSpeech</li> </ul>

Class Name: FakeFreeTTS	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>Used only for testing purposes</li> <li>Simulates the calls of methods of real FreeTTS library to transform the text to audio (play audio, set parameters of audio) and writes this information to test files</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>Implements the test of play and set audio parameter methods of interface ITextToSpeech</li> </ul>

## Package View:



Class Name: GUI	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Implements the graphical user interface</li> <li>▪ Implements the actions of the buttons</li> <li>▪ Implements the replay functionality for the actions of user</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ With class About</li> <li>▪ With class Help</li> <li>▪ With class FileProperties</li> </ul>

Class Name: About	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Display information about the application (Authors, Release Date,Version)</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ -</li> </ul>

Class Name: Help	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Display help information to the user about the application</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ -</li> </ul>

Class Name: FileProperties	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>▪ Display information about the current file properties (Author, Title, Creation/Save date, path)</li> </ul>	<b>Collaborations:</b> <ul style="list-style-type: none"> <li>▪ -</li> </ul>