**Translator**

Date: 2019-03-27

Students: Rokas Jankunas, Emilija Budryte, Calum Logan, Alban Marku

# Introduction

The task that has been set for our group is to create an interactive dictionary using Java. The program will allow the user to input multiple words in English causing the program to translate them into Lithuanian, Swedish or Albanian languages. The program must also allow the user to add and remove words from the dictionary and view it fully. The words will be stored using hashmaps. Hashmaps allow words to be easily added and removed from the list without causing errors with words already in the list. The sources that were used include online spreadsheets for all of the languages.

# Requirements

## Functional Requirements

1. The program shall be a console application.
2. The program shall accept the original text.
3. The program shall allow the user to select from and to languages.
4. The program shall search the dictionary for translation.
5. The program shall display the translated word or phrase.
6. The program shall allow the user to view the entire dictionary.
7. The program shall allow the user to add the word to a dictionary.
8. The program shall allow the user to delete the word from a dictionary.
9. The program shall allow the user to translate the text file.
10. The program shall be able to translate from one chosen language to English and vice versa.
11. The program shall work with dictionaries in separate files.
12. The program shall output the translation time in words per second.
13. The program shall be able to allow the user to enable or disable an option to add a word to a dictionary.
14. The program may have a graphical user interface.
15. The program may support one or more foreign language.
16. The program may use techniques to improve translation quality (i.e. grammar).

## Non-functional Requirements

1. The Java Runtime Environment (JRE) shall be installed on the system.
2. The program shall standalone Java program.
3. The program interface shall be in English.

# Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| Dictionary | | | Alternatives |
|  | User: | Opens the program |  |
|  | System: | Displays user interface |  |
|  | User: | Enables the additional setting for adding a new word | A1 |
|  | User: | Chooses the language from or the language to |  |
|  | User: | Enters the text into the input box | A3, A4, A5 |
|  | User: | Clicks a button for submission |  |
|  | System: | Finds a translation and displays it in the output box | A2 |
|  | User: | Loops 4,5,6,7,8 until the user clicks exit |  |
|  | User: | Clicks exit |  |

|  |  |  |
| --- | --- | --- |
| A1 – Adding a new word to the dictionary is disabled | | |
|  | System: | Loops 4,5,6,7,8 until the user clicks exit |

|  |  |  |
| --- | --- | --- |
| A2 – Translation does not exist | | |
|  | System: | Displays the output with original words for missing translations |
|  | System: | Displays a new window with an input box where the user can enter a word and its translation |
|  | User: | Enters a new word and its translation to the dictionary |
|  | System: | Adds the word to the dictionary |
|  | System: | Saves the dictionary file |
|  | System: | Returns to step 6 in the main flow |

|  |  |  |
| --- | --- | --- |
| A3 – User selects option to remove the word from the dictionary | | |
|  | System: | Displays an input box where the user can enter a word |
|  | User: | Enters the word which should be removed from the dictionary |
|  | System: | Deletes the word from the dictionary |
|  | System: | Saves the dictionary file |
|  | System: | Returns to step 6 in the main flow |

|  |  |  |
| --- | --- | --- |
| A4 – Loading and displaying the dictionary | | |
|  | System: | Displays the whole dictionary to the user |

|  |  |  |
| --- | --- | --- |
| A5 – Loading and translating a text file | | |
|  | System: | Translates the text file |
|  | System: | Returns to step 6 in the main flow |

# Class Design

| **Name of class** | **Fields (with type)** | **Methods** |
| --- | --- | --- |
| Tester | Translator translator; | initialise()  processUserChoices()  displayMenu()  translateInput() |
| Translator | List<HashMap<String, String>> dictionaries;  List<String> exceptions;  boolean fileRead;  boolean isReading;  boolean isWriting;  boolean pendingRead;  boolean pendingWrite;  boolean isAddNewWordsToDictOptionEna  bled;  boolean turboMode;  String[] languageFileNames;  String[] phrasalVerbs; | initialise()  translate()  translateFile()  addToDictionary()  removeFromDictionary()  removeFromDictionaryBy Value()  printDictionary()  getDictionary()  getPhrasalVerbs()  readFile()  writeFile()  flipDictionary()  performPending() |
| MainFrame | JFrame frame;  JTextArea textOriginal;  JTextArea textTranslation;  JButton btnAdd;  JButton btnRemove;  JButton btnPrintDict;  JButton btnTranslateText;  JLabel lblLangFrom;  JLabel lblLangTo;  JMenuBar menuBar;  Translator translator;  int languageIndex;  JScrollPane scrollPane;  JScrollPane scrollPane\_1;  String languageFrom;  String languageTo;  boolean[] englishIsOnLeft;  boolean[] dictionaryLoaded;  boolean isAddNewWordsToDictOptionEnabled;  JCheckBox chckbxNewCheckBox;  JLabel lblFlagIndicator; | main()  MainFrame()  initialize()  translate()  actionPerformed()  changeLabels() |
| OpenFile | - | String pick() |
| FileFrame | JPanel contentPane;  JTextPane fileOutput;  JTextPane fileText;  JButton btnNewButton;  JButton btnTranslate;  String fileName;  Translator translator;  int languageIndex;  JScrollPane scrollPane;  JScrollPane scrollPane\_1; | FileFrame() |
| RemoveFrame | JPanel contentPane;  Translator translator;  int languageIndex;  boolean isLeft;  boolean isEnglishOnLeft;  String leftText;  String rightText;  JLabel lblText; | removeFrame()  actionPerformed() |
| AddFrame | JPanel contentPane;  Translator translator;  int languageIndex;  boolean englishOnLeft; | AddFrame() |
| PrintFrame | JPanel contentPane;  Translator translator;  int languageIndex; | PrintFrame() |

# Class Diagram

# D:\Desktop\diagram.png

# Pseudocodes for main methods

**Translate method**

takes the word or a phrase as an input

takes language index as an input

**if** the word is not in the exceptions list

gets the value from the hashmap for this key

**if** that value is null

**if** language is Lithuanian

splits sentence into words

creates a new string builder

**for** each word in a sentence

gets character between previous and this word

**If** the word is not in the exception list

gets the value from the hashmap for this key

**if** the value is null

adds characters and original words into the string builder

**else**

adds character and translation into the string builder

**endif**

**end** loop

**return** string builder toString

**else**

**return** original word

endif

**else**

**returns** translation

**endif**

**endif**

**return** empty string

**Read file**

set isReading to true

set fileRead to false

creates a new thread

creates file reader

creates buffered reader

**while** the line is not null

read the line

splits the line into 2 tokens separated by the tab character

associates the first value with the second value as a key to the language specific array value dictionary hashmap

**closes** buffer reader

set fileRead to true

set isReading to false

**prints** dictionary size

**Write file**

**if** isReading is false

creates a new thread

creates file output stream

creates print writer

**for** each entry value in the dictionary hashmap

writes key, tab, value and new line symbol

closes print writer

performs pending tasks

**else**

pending write equals true

**Adding word to dictionary**

receives English and other language words as input

**if** this such word exists

gets this value

puts a new foreign word as a value to the key which is an English word to the dictionary

**if** old value is not null

prints what value was overridden

**else**

prints translation added message

**Removing word from dictionary**

receives English word as input

**if** this such word exists

gets this value

removes translation from dictionary hashmap

**if** old value is not null

prints translation removed message

**else**

prints translation was not in the dictionary message

**Flip dictionary**

Creates a new thread

Creates a new hashmap

**For** each entry in the language specific array value dictionary hashmap

Adds entry’s key as a value and entry’s value as a key to the new hashmap

Replaces language specific array value dictionary hashmap with the new hashmap

Sets language specific boolean flipped array value to the opposite value

Runs new **thread**

**Perform pending**

Takes language index for writing and language index for reading

**If** pendingWrite is true and language index for writing is not -1

Calls method to write file with language index for writing

**If** pendingRead is true and language index for reading is not -1

Calls method to read file with language index for reading

# Achieved tasks

Rokas – methods for translating, adding and removing words, quality assurance, handling errors and grammar issues, report, preparation for the demonstration.

Emilija – secretary, working with text files and dictionaries, handling grammar issues, created test plan and worked with testing, report, preparation for the demonstration.

Calum – made GUI, report, preparation for the demonstration.

Alban – partially contributed to writing report and making GUI.

# Test plans & Results

| **Test Description** | **Test Data** | **Expected result** | **Worked?** |
| --- | --- | --- | --- |
| Simple one word translation  (English to Lithuanian) | meal | Valgis | Y |
| Simple one word translation  (Lithuanian to English) | keleivis | Passenger | Y |
| Simple one word translation when the word is not in the dictionary  (English to Lithuanian) | drove | Drove  if an option to add a new words is enabled, the new window pops up | Y |
| Phrasal verb translation  (English to Lithuanian) | Break up | Išsiskirti | Y |
| Phrasal verb translation  when the phrase is not in the dictionary  (English to Lithuanian) | Aim at | Tikslas  (literal translation) | Y |
| Number translation | We read 2 books | Mes skaityti 2 knygos | Y |
| Lithuanian exceptions | The king, a tree | Karalius, medis  (‘the’ and ‘a’ should be omitted) | Y |
| Advanced translation  (English to Lithuanian) | My name is Bob and I live in Dundee. | Mano vardas yra Bob ir aš gyventi Dundee. | Y |
| Advanced translation  (Lithuanian to English) | Kiekvienas vasara man ir mano šeima eiti į jūros | Every summer me and my family go to sea | Y |
| Adding a word to the dictionary when it is in the dictionary | Screen, ekranas | Outputs the word pair was overridden | Y |
| Adding a word to the dictionary when it is not in the dictionary | Snowboard, snieglentė | Successfully adds a word to the dictionary and shows a pop up message | Y |
| Removing an English word from the dictionary when it is in the it | About | Successfully removes a word from the dictionary  and shows a pop up message | Y |
| Removing a Lithuanian word from the dictionary when it is in the it | Gyventojas | Successfully removes a word from the dictionary  and shows a pop up message | Y |
| Removing an English word from the dictionary when the word is not in it | flat | Outputs ‘word cannot be deleted because it is not in the dictionary’ pop up message | Y |
| Removing a Lithuanian word from the dictionary when the word is not in it | butas | Outputs ‘word cannot be deleted because it is not in the dictionary’ pop up message | Y |
| Printing the whole dictionary | lithuanian.txt | Successfully prints out the whole dictionary to the console | Y |
| Translating a big file (English to Lithuanian) | inputEnglish.txt | Outputs translation in output the text box and pops up a message with translation speed | Y |
| Handling punctuation and uppercase letter after relevant symbols (.?!) | My location – Dundee. Am I in the Scotland, UK? Yes, I am! | Mano vieta – Dundee. Esu aš Scotland, UK? Taip, aš esu. | Y |

# Self-evaluation

For implementing our dictionary, we have decided to choose HashMaps as our main data structure since it is very effective. HashMaps usually take O(1) time to access and add elements. We have managed to fulfil all mandatory requirements as well as optional ones. In order to keep 3 different languages (Lithuanian, Albanian and Swedish) we have used an ArrayList of HashMaps.

At first, we were thinking of using stacks to add translations and separating characters so that we could easily remove the most recent one for multiple word phrases. However, that turned out to be inefficient in both memory allocation and speed.

In addition, we have used a simple array list for Lithuanian exceptions. Exceptions and phrasal verbs is an addition for improving grammar for translations between English and Lithuanian and vice versa. Besides, for splitting sentences into separate words we have settled on using regular expressions (regex). This takes care of the characters between the words and returns them into corresponding positions in the final translation.

We have also used multithreading for most methods, typically the ones that might take longer to complete, for example those that involve traversing through the dictionary. Because functions like writing and reading a dictionary file might occur at the same time (which might break a file), we have come up with our own solution of pending tasks – when reading or writing a file is completed, it checks if the other task needs to be done and then calls appropriate functions (and does not open the file when the other task is being performed).

When making the GUI, we were initially going to write the whole thing writing code with Swing, but we found the WindowBuilder plugin, which uses Swing with a visual editor to create a GUI much faster. There were not many problems with creating the GUI – the main issue was ensuring that each component was fit for purpose to make sure all of the code worked as planned. Using TextAreas inside ScrollPanes to make sure scrolling worked for large amounts of text was important and took a while to figure out, but other than that the GUI was fine, since WindowBuilder makes it easier to build a basic Swing GUI.

# User Manual

Our Java translator uses dictionary files to translate words from English into 3 languages – Lithuanian, Swedish, and Albanian. The user can use a drop-down menu in the top left corner to change their desired language, which will affect which language the program uses for each function.

With the selected language, the program can allow the user to: translate a word from English into their chosen language (or vice versa); add or remove a new word to the selected language’s dictionary; print the entire selected dictionary; and translate an entire text file on the user’s system.

**Main Menu**

On the main screen of the application, you can select the language from the menu bar (default is Lithuanian), and then type your input into the left-hand-side text input box. Then, click “translate” and the translated word in the language selected (or in English if you’re translating from a non-English language) will appear on the right hand side. If it does not exist in the selected dictionary, the word will instead display as typed on the left hand side.

You can also check an option which will let you add a word to the dictionary if the program attempts to translate a word and cannot find the translation. This works in the same way as the Remove word screen as below, but will automatically fill the left-hand-side text field with the word you tried to translate. In both fields, if you’re trying to translate a large amount of words, there’s a scroll bar you can use to view more of the text.

From the main screen you can also pick 4 other options – Add a word, Remove a word, Print dictionary, and Translate a file.

**Add a word**

Selecting this option will allow you to add a new word to the dictionary for the language you’ve selected. A new window will pop up, and you must type the word in both languages in the two given text fields. Simply click “Add” and the new word will be in the dictionary!

**Removing a word**

To remove a word is similar to adding one, except this time all you have to do is type the English word in the text field. After clicking “Remove,” the word you typed and its translation will be deleted from the dictionary.

**Print Dictionary**

Choosing this option will simply open up a new window with a text field with a scroll bar containing each word and its translation in the dictionary for the language you selected from the main menu.

**Translate a Text File**

Finally, this option will open a new window with two text fields and two buttons – “choose” and “translate.” To translate a file, you must first click “choose” which will open up a file explorer, from which you can find a .txt file to load into the program. Once this has been selected, select “translate” and the original .txt file contents will be displayed on the left, followed by its translation (in the currently selected language) on the right.

**Running automated tests:**

Entering 1 into console will run automated tests (from the test sheet).