Alan Haverty DT211/3
Network Programming – CA1
TxtClock Documentation

Name: Alan Haverty

Course/Year: DT211/3

Student ID: C12410858

DIT Email: alan.haverty@student.dit.ie

Module: Network Programming – CA1

Document: TxtClock Documentation

Github Repo: https://github.com/ahaverty/NP-CA-1/

Table of Contents

TxtClock Program Overview

The TxtClock program is a java program that tells the time using words rather than digits. The program outputs a time as an English phrase e.g. “It is a quarter past two o’clock” for the 24hr time “14:15”.

TxtClock has three methods of getting a time to translate

1. A Remote NTP Server
2. The Local System Time
3. User Input through command-line argument

Requirements

The TxtClock requires:

1. The project source code from Github https://github.com/ahaverty/NP-CA-1
2. The project lib folder with the commons-net-3.3-sources.jar included
3. Java version 1.7
4. (Optional) An internet connection to retrieve the time from a remote NTP server

To note: The commons-net-3.3-sources.jar is already contained in the project’s lib folder on Github, but can also be found at here if needed:

http://commons.apache.org/proper/commons-net/download_net.cgi

Installation

1. Clone the entire ‘NP-CA-1’ project https://github.com/ahaverty/NP-CA-1 to a folder, preferably named NP-CA-1 on your computer.
2. Open a console and change your directory to the root folder of the project e.g. (C:\NP-CA-1)
3. Make a directory called ‘bin’ in this folder:

Dos & Unix: **mkdir bin**

4. Compile the java source code along with the apache commons net jar file to the newly created bin folder using the following command:

Dos: javac -d bin -cp lib\commons-net-3.3-sources.jar src\txtclock*.java
Unix: javac -d bin -cp lib/commons-net-3.3-sources.jar src/txtclock/*.java

5. Copy the reference text file ‘eng_words.txt’ from the src folder to the bin folder using the command:

Dos: **copy src\txtclock\words_eng.txt bin\ie\dit\student\haverty\alan\txtclock**
Unix: cp src/txtclock/words_eng.txt bin/ie/dit/student/haverty/alan/txtclock/

6. Run the program with no commands to avail of the NTP time or System Time

Dos & Unix: **java -cp bin ie.dit.student.haverty.alan.txtclock.TxtClock**

OR

Use a 24hr time as an argument to translate into English (hh:mm or hh:mm:ss)

Dos & Unix: **java -cp bin ie.dit.student.haverty.alan.txtclock.TxtClock 10:15**

Testing

The TxtClock program has been tested in the following ways:

White Box Testing

Tested internal functions including the validation function and the time to sentence conversion function.

Note: The main test case can be run using the following command without arguments:

java -ea -cp bin ie.dit.student.haverty.alan.txtclock.TestTxtClock && echo success

Note: The –ea parameter enables the assert's, within the test class, to fail a test.

TimeController.validateTime()

This function handled the validation of a user's input by matching it to the regex string: ([01]?[0-9]|2[0-3]):([0-5][0-9])(:([0-5][0-9]))?

The function was tested using asserts on the returned Boolean by providing it with known valid 24hr strings and known invalid 24hr strings. The results can be viewed by running the TestTxtClock program with the command provided above.

```
C:\Users\Alan\workspace\NP-CA-1>java -ea -cp bin ie.dit.student.haverty.alan.txt
clock.TestTxtClock && echo success
Input '13:20' passed validation as expected.
Input '14:15' passed validation as expected.
Input '15:30' passed validation as expected.
Input '15:35' passed validation as expected.
Input '16:45' passed validation as expected.
Input '17:00' passed validation as expected.
Input '23:59' passed validation as expected.
Input '00:40' passed validation as expected.
Input '22:19:00' passed validation as expected.
Input '04:08:15' passed validation as expected.
Input '22:19:30' passed validation as expected.
Input '12:00' passed validation as expected.
Input '00:00' passed validation as expected.
Input '24:00' failed validation as expected.
Input '25:00' failed validation as expected.
Input '-1:20' failed validation as expected.
Input 'one:05' failed validation as expected.
Input '13:20:' failed validation as expected.
Input '22:19:60' failed validation as expected.
Input '21:01:5' failed validation as expected.
Input ':::' failed validation as expected.
Input 'nan' failed validation as expected.
Input 'null' failed validation as expected.
Input '' failed validation as expected.
Input '0:0' failed validation as expected.
Input '9398487.324095403976387345655' failed validation as expected.
```

Input such as regular 24hr time format was used to test that the requirements were met. Edge tests, out of bound time strings and irregular input and characters were tested to ensure the validation regular expression only allowed good input.

LanguageController.timeInEnglish()

The time to language function was also tested using known good 24hr input and their expected English sentence output. The output from the function was matched with the expected string and asserted; proving the time in English function works as expected for valid 24hr time.

Here is the output from TestTxtClock:

```
13:20 : It is exactly twenty minutes past one o'clock
13:20 : Translated to english sentence as expected
14:15 : It is exactly a quarter past two o'clock
14:15 : Translated to english sentence as expected
15:30 : It is exactly half past three o'clock
15:30 : Translated to english sentence as expected
15:35 : It is exactly twenty-five minutes to four o'clock
15:35 : Translated to english sentence as expected
16:45 : It is exactly a quarter to five o'clock
16:45 : Translated to english sentence as expected
17:00 : It is exactly five o'clock
17:00 : Translated to english sentence as expected
23:59 : It is exactly one minute to twelve o'clock
23:59 : Translated to english sentence as expected
00:40 : It is exactly twenty minutes to one o'clock
00:40 : Translated to english sentence as expected
22:19:00 : It is exactly nineteen minutes past ten o'clock
22:19:00 : Translated to english sentence as expected
04:08:15 : It is eight minutes past four o'clock and fifteen seconds
04:08:15 : Translated to english sentence as expected
22:19:30 : It is nineteen minutes past ten o'clock and thirty seconds
22:19:30 : Translated to english sentence as expected
12:00 : It is exactly twelve o'clock
12:00 : Translated to english sentence as expected
00:00 : It is exactly twelve o'clock
00:00 : Translated to english sentence as expected
success
```

Functional Testing

The TxtClock was mainly tested internally by ensuring the validation and translation functions worked as expected through the TestTxtClock program.

However, manual testing was also performed to ensure the TxtClock's main method performed the correct procedures depending on the arguments provided.

The program was provided with correct and incorrect arguments to ensure the expected success and error messages were output:

```
C:\Users\Alan\workspace\NP-CA-1>java -ea -cp bin ie.dit.student.haverty.alan.txt
clock.TxtClock
Time taken from: NTP Server
20:37:41

It is twenty-three minutes to nine o'clock and forty-one seconds

C:\Users\Alan\workspace\NP-CA-1>java -ea -cp bin ie.dit.student.haverty.alan.txt
clock.TxtClock 14:15
Time taken from: User Input
14:15:00

It is exactly a quarter past two o'clock

C:\Users\Alan\workspace\NP-CA-1>java -ea -cp bin ie.dit.student.haverty.alan.txt
clock.TxtClock Incorrect:Time
TxtClock has detected incorrect input.
Options:
1: Provide one argument in the 24hr time format "HH:MM:SS" or "HH:MM".
2: Do not provide any argument to use a remote NTP server to get the time.
```

Installation Testing:

The TxtClock program was also tested for issues with the installation, in particular when the words_eng.txt file was missing. Removing the text file and running the program resulted in a graceful program exit and information message.

```
C:\Users\Alan\workspace\NP-CA-1>java -ea -cp bin ie.dit.student.haverty.alan.txt
clock.TxtClock 14:15 && echo success
Time taken from: User Input
14:15:00

Error: Unable to find file: 'words_eng.txt'. Please check that it exists.
```

Internet Connection Tested:

The TxtClock was tested for connection issues when a user provided no arguments. Usually the program will query a pool of NTP servers using 'pool.ntp.org' for a NTP packet in the case of no user input, but when the NTP request times out, the local system time is used instead. The TxtClock program outputs the method used on every run.

The possibilities are:

- Time taken from: User Input
- Time taken from: NTP Server
- Time taken from: Local System Time

Error Handling

Error Codes

0	The program completed successfully
-1	When the user provided an argument but the String failed the regular expression validation
-2	When the user enters more than 1 argument
-3	When an incorrect file was passed to the importer function
-4	When the words text file could not be found
-5	When the file parser encountered invalid data within the file