Future students | International students | Current students | Research | About Victoria | Log In

XMUT102 home

Course Outline

Additional Information

Schedule & Notes

Assignments

Your Marks

Weekly timetable

People

Java Resources

FAQ

Java documentation

Test/Exam Archive

School of Engineering and Co… Courses/XMUT102_2018T2 Assignments

Assignment7

Assignment 7

Introduction to Computer Program Design: Assignment 7

- Due 10 Dec 2018 7 pm

Goals

By the end of the assignment, you will have gained experience in writing programs using ArrayLists and will also have had more practice at designing programs using loops.

Resources and links

- Download zip file and unzip it.
- Submit your answers.
- Marks and Feedback

Summary

Programs with ArrayLists

- WaveformAnalyser:
  ➡ reads a waveform from a file, displays it, analyses it, and edits it.
- Reflection.
  ➡ Write up your reflections on this assignment.

Overview

The assignment focuses on using ArrayLists to store collections of values. Mostly, the programs read data from a file and construct an ArrayList of the data. They then do things to each element of the list, or manipulate the list in more complicated ways. The assignment is a bit shorter than usual because of the terms test. `WaveformAnalyser` is like `TemperatureAnalyser` from assignment 3, but reads numbers from a file into an ArrayList and analyses the signal. It also displays the waveform, displays the amplitude, highlights the peaks, draws the upper envelope (the lines linking the peaks), and lets the user edit out parts of the waveform.

To Submit:

Submit your best version of `WaveformAnalyser.java` and your `Reflection.txt` files by the due date. Remember to click the button for the final step of the submission process after you have uploaded the files (which does some automated checking for you).

You may work in pairs for the Core and Completion parts, but if you do, you *must* include a comment at the top of your program saying who you worked with. You must do the Challenge parts on your own.

Preparation

Download the zip file and extract it to your home folder. It should contain template for the Java program you are to complete. Read through the whole assignment to see what you need to do.

Look again at your answer and/or the model answer to the TemperatureAnalyser program from assignment 3 and go over the code examples in slides from the lectures on ArrayLists.

Waveform Analyser

Many music players will allow you to display the sound being played. One of the displays is typically the waveform. For this program you will write a very simple waveform display.

There are several things about a waveform that one may be interested in:

- The highest and lowest (most negative) points on the waveform are significant.
- The distortion level: if the signal is too large (positive or negative), the sound might get distorted. Reporting on the points gone over the distortion level becomes then useful.
- The peaks of the waveform which are all the points that are greater than both their neighbouring points.
- The upper envelope which is the graph that links all the peaks of the signal.
- Editing the waveform, eg by scaling it down so it is all below the distortion level, or removing some segments of the waveform.

`WaveformAnalyser` is an extension of the `TemperatureAnalyser` program from assignment 3 which displays a waveform and performs some analysis and editing. However, instead of reading data directly from the user, `WaveformAnalyser` reads the data from the file (eg `waveform1.txt` and `waveform2.txt`) into an ArrayList stored in the `waveform` field.

The code that sets up and responds to the buttons is provided for you - the parts of the program you have to complete are the ones involving the ArrayList.

The constructor sets up an interface with nine buttons, and responds to the mouse:

- **Read Data**: [core] reads data from a waveform file into an ArrayList field.
- **Display Waveform**: [core] displays the waveform as a line graph.
- **Spread**: [core] displays the maximum and minimum values with two horizontal lines (on top of the waveform).

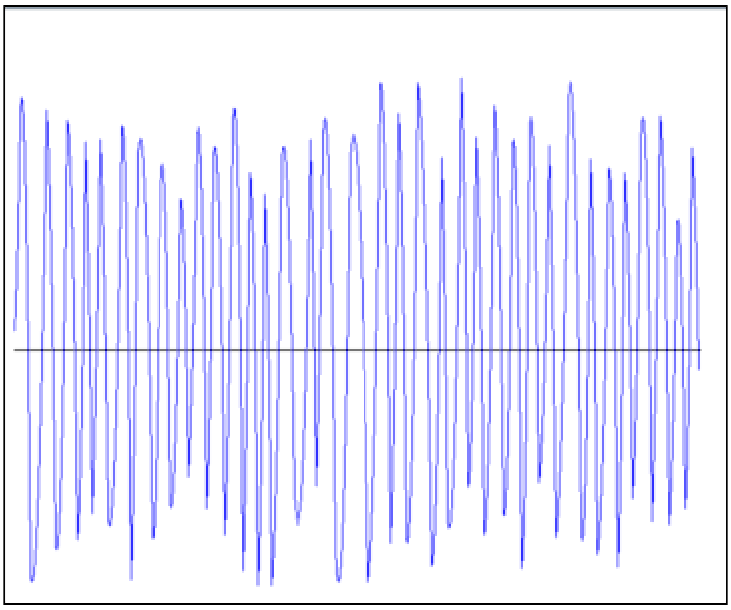
- **Display Distortion**: [completion] displays a waveform that highlights in red the distorted values.
- **Peaks**: [completion] plots the peaks with small green circles.
- **Normalise**: [completion] normalises all the values by scaling them down so there is no distortion and redisplays the waveform.

- **Envelope**: [challenge] displays the upper "envelope" of the displayed waveform by connecting all the peaks.
- **Save**: [challenge] saves the current waveform values into a file.
- The mouse will allow the user to select a region of the waveform to delete it.

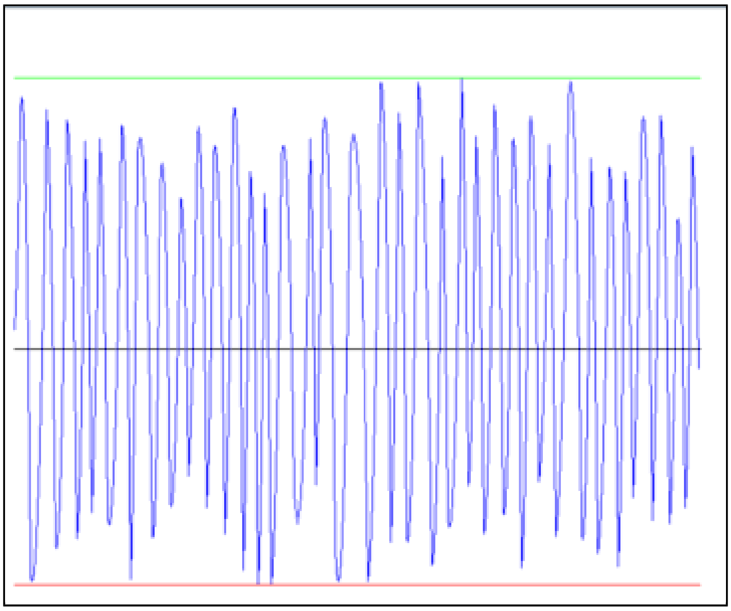
Core

For the core, you should complete the following methods:

- `read()`: creates an ArrayList stored in a field, asks user for a waveform file and reads data from the file into the ArrayList.
- `display()`: Displays the waveform as a line graph. The horizontal line representing the value zero should also been drawn. If the lines go off the window, that is OK.



- `showSpread()`: Displays two horizontal lines on the waveform, one green line for the maximum value and one red line for the minimum (most negative) value.



Completion

For the completion, you should complete the following methods:

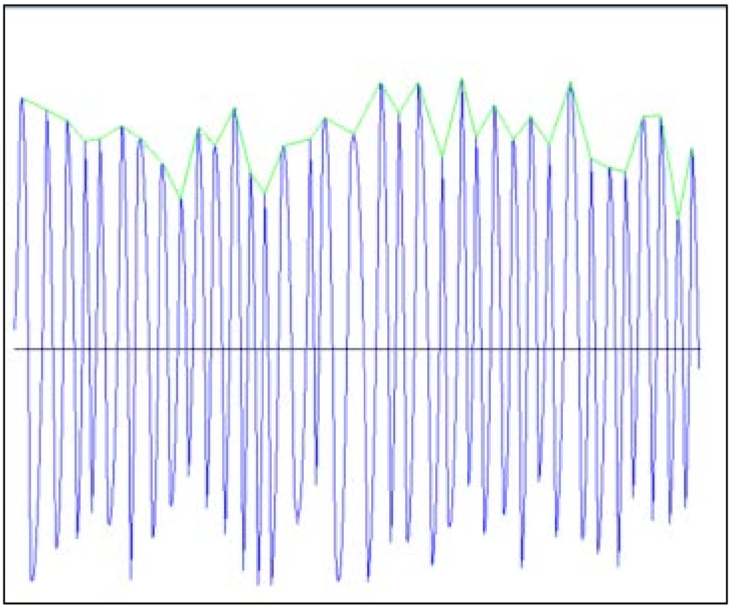
- `displayDistortion()`: Shows in red the distorted part of the signal. A distorted value is defined as one that is either greater than the positive value of the threshold or less than the negative value of the threshold. This method draws in red every line that has either end point beyond the distortion threshold. Note that the threshold value is stored in the constant field `THRESHOLD`.

- `highlightPeaks()`: plots the waveform and the places small green circles on all the peaks. A peak is defined as a point that is greater or equal to **both** neighbouring points. Note the size of the circle is stored in the constant field `SIZE_CIRCLE`
- `normalise()`: Finds the largest value (positive or negative) in the waveform, and scale all the values down so that the largest value is now equal to the distortion threshold. Then redraws the waveform.
  Hint: multiply each value by *threshold/max value*

Challenge:

For the challenge, you should complete the following methods:

- `doEnvelope()`: displays the upper envelope with green lines connecting all the upper peaks.



- `save()`: asks the user for a filename and saves the current waveform to that file.
- `doMouse()`: Lets user select a region of the waveform with the mouse, deletes that section of the waveform, and redisplays the waveform.

Reflection

Answer the following questions in the `Reflection.txt` file, and make sure you submit it.

0. List which methods of the WaveformAnalyser you completed.

1. Explain why ArrayLists are better than using lots of variables when your program needs to store lots of values.

2. ArrayLists allow you to write programs with collections of data, but so do files. What are the similarities and differences between files and ArrayLists. What can you do with one that is hard or impossible with the other?

Print version | Backlinks |

Useful links

Undergraduate study Postgraduate study

Research groups Staff

Wiki

Useful contacts More contacts □

□ +64 4 463 5341

□ office@ecs.vuw.ac.nz

Victoria University Faculties Contacts and directories Campuses Study Students Staff

Site info Site map Feedback Glossary