# IS-211 Mandatory Assignment 1

Even Åby Larsen

## 1 Introduction

This document contains the instructions for the first mandatory. Your task is to write a simple text editor. You should implement the following functionality:

**Insert char** When the user presses the key for a printable character, the character should be inserted at the location of the cursor, and the cursor moved one position forward. Printable characters are a-zæøå, A-ZÆØÅ, 0-9, and separators (,.;:!) item[Delete char] Pressing backspace (the key labeled with an x inside a box) should delete the character before the cursor and the Delete/Del button should delete the character under the cursor

**Navigation** The cursor can be moved by pressing one of the arrow keys. When moving up or down the cursor should stay, if possible, in the same horizontal position.

If you want a challenge, try to implement automatic linebreaking; When the cursor moves to far to the right, it is moved to the next line, along with any text to the right of the cursor.

Another thing you can try is to save/open files.

## 2 How the application works

Whenever the user presses a key on the keyboard, an event object is created andd sent to the handler for that type of event. An example of a handler is editor.action.InsertAction, which handles insertion of characters. The tables used to find event handlers are in editor.Editor.

The handler calls the appropriate method in the Document. You will have to write those methods. They must update the data structure, and the Display.

# 3 Your task

1. Decide which data structure to use internally in the Document class, and present the reasons for you choice in a separate document, or in the javadoc comment for Document.

2. Write the bodies of the the actionPerformed() method in the Action classes. In general they just call corresponding service method in Document. The following action classes are provided. They should work as is, unless you choose a data strucure that is radically different form my choice.

   - EditorAction superclass for the "real" action classes.
   - InsertAction - inserts one character
   - InsertLineAction - insert a new line
   - deleteNextAction - delete the caracter after the cursor
   - deleteNextAction - delete the character after the cursor
   - deltePrevAction - delete the character before the cursor
   - LineBreakAction - automatically split an overlong line
   - ArrowKeyAction - move the cursor around

## 3.1 Hints

Describe a text document to you partner. Explain its structure without thinking about how to code it. Make some object diagrams to remind you later.

Now you're ready to select (a) data structure(s).

# 4 Classes used

## 4.1 Application specific

### 4.1.1 editor.Document

This is perhaps the most important class in the application. It contains the document data, and has methods for the functions listed above.

#### 4.1.2 editor.Editor

This is the main class of the application. It sets up the Graphical user interface and the key bindings. The user interface is in editor.display.CharacterDisplay. You do not have to make changes to any of these classes, unless you add functionality beyond what is required.

#### 4.1.3 editor.action.EditorAction

This is the superclass for all actions. Actions represent something the user can do in the user interface.

#### 4.1.4 editor.action.InsertAction

This is an example of a subclass of EditorAction. The actions represent user actions, usually a keyboard button press. The actionPeformed() method should call the appropriate method in Document to perform the action. (You will most likely have to write the method in Document).

### 4.2 Library classes

#### 4.2.1 java.awt.ActionEvent

Used to transfer info from the GUI to the document

#### 4.2.2 java.awt.BorderLayout and java.awt.Container

Manages positioning of the parts of the user interface

#### 4.2.3 Other java.awt classes

Used in event handling

#### 4.2.4 javax.swing.*

GUI components