

# CSC 212 Project

## Developing an Application for Manipulating Subtitles

College of Computer and Information Sciences  
King Saud University

Fall 2017

### 1 Introduction

Subtitles are captions displayed at the bottom of a cinema or television screen that translate or transcribe the dialogue or events taking place in the video (see Figure 1 for an example). Subtitles can be hard-coded into the video stream or come as a separate file. The goal of this project is to develop an application for manipulating subtitle files.



Figure 1: Example subtitle.

### 2 Data Format

There are many file formats for representing subtitles, but arguably the simplest of these formats is the *SubRip* format, also known as *SRT* format because of its file extension. An SRT file is a text file consisting of a sequence of subtitles, each having the following structure:

1. A sequential number which starts at 1.
2. The start and end times when the subtitle should appear on screen. The time is specified in the format: *HH:MM:SS,ms* (the last field is an integer value in milliseconds). The display intervals of subtitles must not overlap.
3. The text of the subtitle. Line breaks are allowed.

4. An empty line indicating the start of a new subtitle (the last subtitle is not followed by a an empty line).

The following shows sample subtitles from the movie *Winnie the Pooh* (2011):

```
140
00:08:40,103 --> 00:08:41,813
Just as I suspected.

141
00:08:41,897 --> 00:08:45,108
Owl, we need honey.
```

### 3 Requirements

In this phase, you are required to implement the following classes and interfaces (**This specification must under no circumstances be modified**):

```
// Interface representing time
public interface Time {
    int getHH();

    int getMM();

    int getSS();

    int getMS();

    void setHH(int hh);

    void setMM(int mm);

    void setSS(int ss);

    void setMS(int ms);
}

// This interface represents a single subtitle.
public interface Subtitle {
    // Return the start time of the Subtitle.
    Time getStartTime();

    // Return the end time of the Subtitle.
    Time getEndTime();

    // Return the subtitle text.
    String getText();

    // Set the start time of the Subtitle.
    void setStartTime(Time startTime);

    // Set the end time of the Subtitle.
    void setEndTime(Time endTime);
}
```

```

    // Set the subtitle text.
    void setText(String text);
}

// This interface represents a subtitle sequence.
public interface SubtitleSeq {

    // Add a subtitle.
    void addSubtitle(Subtitle st);

    // Return all subtitles in their chronological order.
    List<Subtitle> getSubtitles();

    // Return the subtitle displayed at the specified time, null if no
// subtitle is displayed.
    Subtitle getSubtitle(Time time);

    // Return, in chronological order, all subtitles displayed between the
// specified start and end times. The first element of this list is the
// subtitle of which the display interval contains or otherwise comes
// immediately after startTime. The last element of this list is the
// subtitle of which the display interval contains or otherwise comes
// immediately before endTime.
    List<Subtitle> getSubtitles(Time startTime, Time endTime);

    // Return, in chronological order, all subtitles containing str as a
// sub-string in their text.
    List<Subtitle> getSubtitles(String str);

    // Remove all subtitles containing str as a sub-string in their text.
    void remove(String str);

    // Replace str1 with str2 in all subtitles.
    void replace(String str1, String str2);

    // Shift the subtitles by offsetting their start/end times with the specified
// offset (in milliseconds). The value offset can be positive or negative.
// Negative time is not allowed and must be replaced with 0. If the end time
// becomes 0, the subtitle must be removed.
    void shift(int offset);

    // Cut all subtitles between the specified start and end times. The first
// subtitle to be removed is the one for which the display interval contains
// or otherwise comes immediately after startTime. The last subtitle to be
// removed is the one for which the display interval contains or otherwise
// comes immediately before endTime. The start and end times of all
// subtitles must be adjusted to reflect the new time.
    void cut(Time startTime, Time endTime);
}

public class SubtitleSeqFactory {

    // Return an empty subtitles sequence
    public static SubtitleSeq getSubtitleSeq() {
    }
}

```

```

    // Load a subtitle sequence from an SRT file. If the file does not exist or
    // is corrupted (incorrect format), null is returned.
    public static SubtitleSeq loadSubtitleSeq(String fileName) {
    }
}

```

## 4 Deliverable and rules

You must deliver:

1. A report written using the provided template.
2. Source code submission to Web-CAT.

The submission **deadline** is: **16/11/2017**.

You have to read and follow the following rules:

1. The specification given in the assignment (**class and interface names, and method signatures**) must not be modified. Any change to the specification results in compilation errors and consequently the mark zero.
2. All data structures used in this project **must be implemented** by the students. The use of Java collections or any other library is strictly forbidden.
3. This project is to be conducted by groups of **four** students. Groups of more than four students are not accepted. Groups of less than four students are strongly discouraged and can only be accepted with a special permission from the course instructor.
4. All the members of a group must have the **same course instructor**.
5. All students must **submit** the list of their **group members** within one week of the announcement of this project. Once the groups are chosen, no student can change the group (even if some group members have dropped the course).
6. **Every member** of the group must participate in **all parts of the project**: designing the software, programming and writing the report. Members of the same group may receive different marks according to their participation in the project.
7. The submitted software will be evaluated automatically (using Web-Cat) and in a demonstration (after phase 3) to which all the group members must attend.
8. Any member of the group who fails to **attend the demonstration** without a proper excuse (consult the university and college regulations) shall receive the **mark 0** in the project.
9. In accordance with the university regulation, **cheating** in the project will be sanctioned by the **grade F**.