Development of a Software Application for a Community Library to Manage Its Movie DVDs

Phase 2

Due Date: 18 May 2022

Weighting: 35%

Group or Individual: Individual

1. Introduction

In this project you develop a software application for a community library to manage its movie DVDs. In the development of this software application, you need to use some data structures and algorithms that are covered in this unit to store, manage, and manipulate the data in the software application. In this software application development, you also need to design algorithms to solve some computational problems in the software application and to analyse the time efficiency of your algorithms.

In Phase 1, you have implemented two ADTs, *Member* and *MemberCollection*. When implementing the ADTs, you used your knowledge and skills in linear data structure and array-based algorithms. In this phase (Phase 2), you need to implement two more ADTs, *Movie* and *MovieCollection*. When implementing these ADTs, you need to use your knowledge and skills in non-linear data structures and algorithms.

You are provided the specifications of the two ADTs in C# interfaces, *IMovie.cs* and *IMovieCollection.cs*. You are also provided a skeleton of the implementation of the two ADTs (*Movie.cs* and *MovieCollection.cs*). These C# interfaces and incomplete C# classes can be downloaded from *Assessment Task 2* under *Assessment Tasks* in the CAB01 Blackboard.

Movie is an ADT that models a movie. In addition to the title, genre, classification, duration (in minutes) of a movie, it also contains other information, such as the total number of DVDs of the movie in the library, the number of available DVDs of the movie currently in the library, and a list of members who are currently borrowing/holding a DVD of the movie. In the specification and the skeleton of the implementation of *Movie*, we use an object of the *MemberCollection* class, which you have implemented in Phase 1, to store those members. In this software application, we do not distinguish the DVDs of the same movie. In other words, all the DVDs of the same movie are identical.

MoiveCollection is an ADT that is used to store a collection of movies. The underlying data structure that is used to store the movies in its implementation is a Binary Search Tree (BST). A node of the BST has three fields, left child reference, right child reference as well as a Movie object. Each node in the BST is an instance of *BTreeNode* class, which is defined in *MovieCollection.cs*.

It is assumed in this software development that the movie titles are unique, and the full names of the members are also unique. Your task in this phase is to complete the two new ADTs' implementation. It is important to point out that in this phase, you need to use *Member* and *Membercollection* ADTs that you have implemented in Phase 1.

2. Detailed Tasks

You are provided a skeleton of the *Movie* and *MovieCollection* ADT implementations. Your jobs are to complete the following tasks:

- Complete the implementation of the following methods in *Movie.cs*:
  - *AddBorrower*
  - *RemoveBorrower*
  - *CompareTo*
  - *ToString*
- Complete the implementation of the following methods in *MovieCollection.cs:*
  - *IsEmpty*
  - *Insert*
  - *Delete*
  - *Search by movie object*
  - *Search by movie title*
  - *ToArray*
  - *Clear*
- Use the most efficient algorithm to implement the methods, where applicable
- Comprehensively test all your method implementations using a variety of test data, including normal test values and boundary values, to make sure they meet the functional requirements, to check if the pre-condition(s) and post-condition(s) are satisfied. The pre-condition(s) and post-condition(s) of the methods can be found in the C# interfaces.

3. Assignment Requirements

- The programming language used in this assignment must be *C#*
- You are not allowed to make any change to *IMovie.cs* or *IMovieCollection.cs*
- You are allowed to add private methods to *Movie.cs* or *MovieCollection.cs*
- You are not allowed to add any namespace to *Movie.cs* or *MovieCollection.cs*

4. Submissions

- Your submission should be a single zip file named by *your-student-number*.zip, which compresses *Mpvie.cs* and *MovieCollection.cs*
- Do not include any other file into the zip file except for a copy of the assessment approval if you have an approved assessment extension

- Your submission must be submitted via the Blackboard. Email submissions are not accepted as any email containing a C# program, including attachment, may be blocked by the QUT email server, and then this happens neither the sender nor the receiver will be sent any notice
- You may resubmit your assignment as many times as you wish before the deadline. If you submit your assignment multiple times, we will only mark the last submission before the deadline.