SIT323 Practical Software Development, Trimester 2, 2020

Practical 1 – Week 2

Familiarisation with Assessment Task 1

Introduction

For this practical you will tend to start thinking about many parts of your solution for Assessment Task 1 and probably some of Assessment Task 2. However, this practical focuses on designing your assignment software.

Practical Tasks

1. Your first task for this practical is to understand the requirements for Assessment Task 1. Therefore, obtain the Assessment Task 1 document, read it, comprehend it, ask questions, and wonder about different C# solutions.

   The formats of the taff and cff data files are presented in this Assignment 1 document. These will be helpful for the next practical task.
   - taff represent Task Allocation File Format
   - cff represent Configuration File Format

2. Peruse the data files in the **Programming Task 1** folder in the unit site in CloudDeakin, some of these data files are invalid. These files can be found in the ZIP file called **Assessment Task 1 – Data Files.zip**.

   a) Manually confirm that both files for Test 1 (Test1.taff and Test1.cff) are valid.
   While you're manually validating these files, you should be thinking about how your software will detect any errors too.

   b) Manually confirm that both files for Test 2 are valid.
   While you're manually validating these files, you should be thinking about how your software will detect any errors too.

   c) Manually confirm that both files for Test 3 are valid.
   While you're manually validating these files, you should be thinking about how your software will detect any errors too.

   d) Manually confirm that both files for Test 4 are invalid.
   While you're finding all errors within these invalid files, you should be thinking about how your software will find these errors too.

   e) Several valid allocations are presented in Test1.taff and Test2.taff.
   How long does it take for each of these allocations to run?
   While you're manually determining the runtime of a valid allocation, also think about how your software will determine this runtime.

   f) Which allocations presented in Test3.taff are invalid?
   While you're manually determining these invalid allocations, think about how your software will determine the validity of an allocation.

3. Based on the requirements of Assessment Task 1, design the layout of your GUI. The following functionality will be helpful in considering some GUI elements.

   a) Opening a TAFF file
   b) Validating a TAFF file
   c) Validating a CFF file
   d) Validating allocations
   e) Displaying data
   f) Displaying errors
   g) Exiting the application.

   You may use any GUI mock-up tools. However, you might like to use pen and paper, or create a GUI mock-up by using an IDE as dragging and dropping interface elements is simple and quick in most IDEs.

4. Peruse the configuration and allocation files, and consider how the data in these files can be stored into your C# software.

   a) Which predefined types and classes will you use?
   b) Why would you use the following int, Double, String, List, StreamReader?

5. What are some programmer defined classes that you would consider for your C# software solution?

   Your list might start with:

   - Allocation — this is probably too obvious
   - List<Allocation> — a TAFF file contains 0 or more allocations
   - Configuration — each set of allocations will have a configuration
   - Processor — a processor has attributes such as type, frequency and RAM
   - Task — a task has attributes such as runtime and RAM
   - List<Processor> — an allocation has a number of processors
   - List<Task> — an allocation has a number of tasks
   - ProcessorSpec — a kind of processor has a name and quadratic coefficients