Examination Number: Y1466227

All Eclipse files were written using Eclipse Neon.1

Tests

Test Name	Test Goal	Expected Outcome
PlannerLevel1Test. testStudyBlocksAlternate()	Test that the study blocks alternate between topics.	When more than one topic is added, the study plan should alternate between study blocks for each topic.
PlannerLevel1Test. testStudyPlannerExceptionsAreThrown()	Test that StudyPlannerExceptions are thrown.	When the user tries generate a study plan with no topics, or when a user tries to add a topic with the same name as another topic, a StudyPlannerException should be thrown.
PlannerLevel1Test. testBlockLengths()	Test that the block lengths are correct, and automatically reduce when necessary.	If no study block length is set, the default should be 60 minutes. If the block length is changed by the user, the study block lengths should reflect this. When there isn't enough time left to study a full block before the topic duration is met, a reduced duration study block should be added.
PlannerLevel2Test. testDailyStartAndEndTimes()	Test that the study blocks fit within the daily start and end times, and that these times can be altered.	No study blocks should begin before the daily start time, or finish after the daily end time. If there isn't enough time left in the day for a full block, the block length should be reduced, unless there is less than 10 minutes left in the day; in which case the study block should start the following day.
PlannerLevel2Test. testEventHandling()	Test that events can be added, and that the study blocks work around them.	Events added by the user, should be added to the study planner, and should then be retrievable as an event list. When a plan is generated, the study blocks should work around the events, reducing block durations where necessary.

PlannerLevel2Test. testSetTargetEvent()	Test that event type can be specified, and that target events can be set for topics.	When an event is added of type exam or essay, it should be possible to set it as a target for a topic. If it is of type other, a StudyPlannerException should be thrown.
Test 1	Test that topics can be added and displayed in a topics list, and that a study plan can be generated from them.	Should be able to complete these tasks, and dialogue windows should be displayed if the user does something wrong.
Test 2	Tests that events can be added, and that they are integrated into the study plan list in the GUI. It also tests that target events can be set for a topic.	Should be able to complete these tasks, and dialogue windows should be displayed if the user does something wrong.
Test 3	Tests that the data can be saved, remains persistent after the program has been exited, and then reloaded again.	Should be able to complete these tasks, and dialogue windows should be displayed if the user does something wrong.

Test choice justification

These tests were chosen because together they cover much of the crucial functionality, and aim to highlight aspects not immediately obvious when looking through the programme. Examples include subtleties like: moving on to the next day if there is less than 10 minutes left in the study day, and that target events can be set through the GUI by selecting a topic and an event in their respective lists and clicking the set topic target button. Also, these tests aim to incorporate as many of the classes as possible, thus improving the chances of highlighting potential errors.

File format description and justification

The file format used is human readable to make it simpler to understand, and make debugging easier.

The file uses a comma separated variable approach, with array elements being placed on separate lines, and complete items (e.g. arrays) being separated by a line of four #'s. The line separation of array elements allows each array element's data to be read off a single line, minimising the amount of memory used while reading the file. The comma separation allows each line to split into its individual variables, which can then be assigned as required. The use of #'s allows the program to know when it has read to the end of an item, and can therefore move on to the next one. The first section of the file holds the topics, the second section the study plan, the third the events, and the fourth and fifth the daily start and end times. Additional #'s are used at the end of a topic element line if no target event has been set.