DIT636 / DAT560 - Structural Testing Exercise

Last week, you wrote unit tests for a meeting planner system. Those were based on your ideas regarding proper functionality. Now, we want to finish the job with test cases based on the code structure. The goal for your unit tests is to achieve 100% statement coverage (also known as line coverage) of the code for all classes ***except*** **PlanningInterface or the exception.**

1. Measure coverage of your current test cases (from last week, if you have them) for the meeting planner classes with a tool, such as Emma, Cobertura, or JaCoCo.
    a. If you are using Eclipse, we recommend EclEmma: https://www.eclemma.org/
    b. If you are using IntelliJ IDEA, you can use the bundled IntelliJ IDEA code coverage runner: https://www.jetbrains.com/help/idea/code-coverage.html. You may use any of the three coverage runners.
2. If portions of code are not already covered, add additional unit tests targeting that code.
    a. You may target any of the coverage criteria discussed (Statement, Branch, Basic Condition, All DU Pairs, etc.). We recommend targeting Branch Coverage.
3. If you find portions of the code that cannot be covered by a test case, explain why those elements are not coverable. Alternatively, if you feel some elements should not be covered, justify why that is the case[1].

[1] For example, single-line methods that only call methods from other classes.