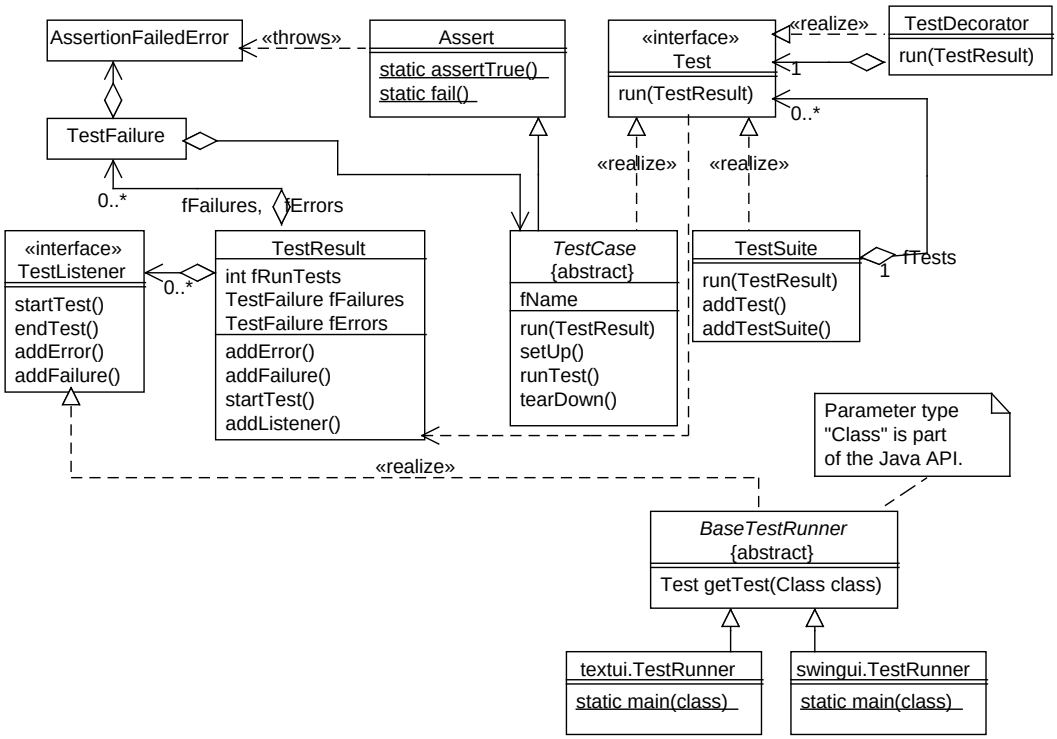
HÁSKÓLI ÍSLANDS
Iðnaðarverkfræði-, vélaverkfræði- og tölvunarfræðideild

HBV205M: Prófun hugbúnaðar / Software Testing · Spring 2021
Dr. Helmut Neukirchen

Assignment 2 · Due 1.2.2021, 10:00

Objectives: Reason about which integration test strategy requires least efforts which is mainly due to creation of test drivers (=easy to create)/stubs (=hard to create); Read the dependency tree/graph from a UML class diagram (including finding leaves/root of dependencies and cyclic dependencies making all members of the cycle belong to the same level).
For planning integration testing, consider the following UML class diagram[1] of the architecture of a software under test[2]:



1. Which integration test strategy (top-down, bottom-up, big-bang[3]) would you select for the above system? Justify your answer!
2. Describe the resulting order of integration[4]. (Just describe the order – you do not need to write down any test cases.)
3. Where do you need test drivers and/or test stubs?

If you have a Google account, work online on a copy of the following template:
`https://docs.google.com/spreadsheets/d/17NbxZZLjm0pbCjlh-2w1jX725s_oIlp3tKsMHxRggYM/copy`
If you have no Google account, you can view and download a copy of the template from here:
`https://docs.google.com/spreadsheets/d/17NbxZZLjm0pbCjlh-2w1jX725s_oIlp3tKsMHxRggYM/edit?usp=sharing`

[1] In case you need to refresh your UML knowledge, e.g. `https://holub.com/uml` provides an overview (scroll down to "Static-Model (Class) Diagram" for a coverage of class diagrams).

[2] Assume that method BaseTestrunner.getTest(...) returns type TestSuite as realisation of interface Test.

[3] As no completion date is given for the individual components, ad-hoc integration is not considered here.

[4] Take care to treat interfaces and abstract classes accordingly: interfaces are not executable and best tested by testing a class that realizes that interface. Abstract classes are not executable, but need a subclass that inherits and provides missing methods. The required subclasses need either to be provided by dummy classes or typically an appropriate subclass can be found within the set of classes that are subject of integration. Finally, note that class *BaseTestRunner* has a not explicitly depicted dependency to interface *Test* due to the return parameter of method *getTest()*.

Formalities

- Encouraged to submit as pair of 2. Submission of solution as PDF via Gradescope (accessible via Canvas)
- To show that you are active right from the start, you need to have submitted 2 assignments after the first 4 assignments.
- The best 8 (out of approx. 13) assignments contribute 20% to the final overall grade. For further formalities, have a look at Chapter 0.
- You need to have read Chapter 2 for solving this assignment. Direct any questions to Piazza: `https://piazza.com/hi.is/spring2021/hbv205m/home`
- Should you have joined the course late, you might not have been added in Piazza nor Gradescope: contact `helmut@hi.is` to get added there.

To keep in mind during flipped classroom sessions

- Have audio working (use smartphone with Zoom app in case of problems, also non-audio problems. In case of crashes, download/install latest Zoom version).
- Switch on video to create a more personal atmosphere.
- First: Solution of last week's assignment presented.
- Next: Questions concerning slides/videos discussed.
- Then: New assignment introduced.
- Finally: Work on assignment in breakout rooms (random allocation of 2 students each).
  - Breakout rooms do not get recorded.
  - First thing to do:
    1. Remember breakout room number: in case you need to reconnect, Helmut can then add you again to your old breakout room – if Helmut remembers to enable, you can also join breakout rooms yourself / if you join late, you can add yourself to an empty breakout room or one that has only one student.
    2. Exchange contact details (email/phone), e.g. via chat in each breakout room, so that you can continue in case of technical problems/after class finishes.
  - Create some shared document to work together on the solution. You can also share your screen inside your breakout room.
  - Use "Ask for help" button (question mark icon) to make Helmut join (but may take time if he is busy in another breakout room). In case of being idle, Helmut will come along each breakout room.
  - Submit one PDF per team to Gradescope (unless you disagree on solution): use Gradescope's group submission feature. Gradescope is reachable via Canvas. You have one week for (re-)submission.
  - In case of some announcement for all: Helmut sends a chat message to all or ends all breakout room sessions for a video session with all. (If Helmut does not mess it up, should be possible to continue in the old breakout room.)
- There is no wrap-up at end of class, i.e. if you are finished with your assignment, there is no need to wait – you are welcome to leave the class.
- Typically, there are not Wednesday classes: use the time to read new slide/watch videos as preperation for next Monday.