DTU Compute
Department of Applied Mathematics and Computer Science
Advanced Programming (02324), Spring 2025
Carlos E. Budde and Ekkart Kindler

Assignment 2b: Generic Stacks with Exceptions

Assignment 2 is concerned with defining and implementing the stack from Assignment 1 in a more elegant way. Part one of the assignment, Assignment 2a (Friday, February 7) was concerned with making the stack generic. Now, Assignment 2b (Tuesday, February 11) is concerned with properly implementing exceptions and handling. The different implementations need to throw the relevant exceptions, and the client application, needs to catch these applications and handle them appropriately.

For Assignment 2b, you can continue with your project/solution for Assignment 2a, which had been based on **`assignment2.zip`** from DTU Learn. In the following, Assignment 2b is broken down into smaller steps.

1. Before you start with implementing something for Assignment 2b, make sure that your code for Assignment 2a is working properly (and, if you did not do it yet, show it to the TAs or teachers). Then you can continue your work on your solution of Assignment 2a.
2. In the implementations of **`LinkedListStack<E>`** and **`ArrayStack<E>`**, make sure that the code raises (throws) the right exceptions when the respective situations occur.
3. For testing this, run all the tests associated with the project assignment2 and check whether all tests, in particular, the ones related to raising the relevant exceptions run correctly now.
4. When manually testing the stack by running the client application **`IntegerStackApp`**, you will see that a lot of exceptions will be shown in the console of IntelliJ. This means that the client application does not properly handle exceptions yet. Make sure that you add the respective try-catch blocks to this application, so that all raised exceptions are caught and dealt with (no console outputs concerning exceptions anymore); in particular, the ones raised when pressing the buttons.
5. When an exception is thrown during such an action, make sure that it is properly caught and, the message text of that exception is shown in the text area for exceptions in this GUI. In the **`IntegerStackGUI`**, this text area is available as attribute **`textAreaExceptions`**. You can append some text to this area by calling the method **`textAreaExceptions.appendText("some text")`**.
6. When pressing the “clear” button, you should also delete all text in this **`textAreaExceptions`**.
7. The solution should be shown groupwise to the teachers or teaching assistants during the tutorial sessions on Tuesdays or Fridays at latest on Friday, February 14). The code of this task will to some extend be used during the next tutorials, which you later will need to be submit (in week 3) together with the other tasks (details to be announced).

References and further reading

[02324 f25 L03.1] Carlos E. Budde and Ekkart Kindler: Lecture Notes for the course Advanced programming. Spring 2025. (DTU Learn).

[Eck 2022] David J. Eck: Introduction to Programming Using Java. Version 9.0, JavaFX Edition, May, 2022. Section 3.7 and 8.3