Software Development, 2018-2019

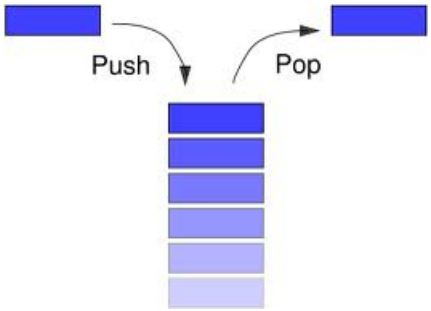
Patricia Navarro Martín, Karsten Gielis, Jonas Geuens, Tim Stas, Jeroen Wauters, Kymeng Tang, Koen Pelsmaekers.

Lab session 3: Implement an interface

Goal

The goal of this lab session is to write an implementation of a given Stack interface and use this implementation to solve the bracket checker exercise. Unit tests are provided.

Exercise 1: Stack



The file StackIntf.java (package be.kuleuven.groept.util) defines a Stack interface with a push, pop, peek and isEmpty method:

- push(): puts a new element on the StackIntf.java
- pop(): takes the top element away from the Stack and returns it; returns null when the Stack is empty
- peek(): returns a reference to the top element of the Stack; returns null when the Stack is empty
- isEmpty(): returns true when the Stack is empty
- clear(): clears the Stack

Write a class (MyStack) that implements this interface.

Use the Unit tests (StackTest.java) to check if your implementation matches the description of the methods of the interface; you can write your own tests too.

Exercise 2: Bracket checker

Use your Stack implementation to implement a “bracket” tester program that uses a GUI with a text box and a “Check” button to check if the brackets in an expression are “valid” (= neatly nested open and close brackets)

f.i. `({a + b}(c + d)[[({{}})]])`: correct ✓
`(a { ) }`: wrong ✗

You can use the View implementations (in Swing and Java FX) or the Unit tests in brackets.jar on Toledo to test your implementation.

Exercise 3: Exception handling

Provide an implementation for the StackIntfWithException interface. Adapt the Unit test and your Brackettester implementation.