Prof. Dr. Christoph Bockisch
MSc Steffen Dick
Fachbereich Mathematik und Informatik
AG Programmiersprachen und -werkzeuge

Philipps Universität Marburg

Companion exercises
Objektorientierte Programmierung: Wintersemester 2021/2022

No. 9, due until 24.01.2022

Task 9.1: PRNDL — 3 Points

Implement a method `public static boolean isReverse(String a, String b)` that returns `true` if `a` is the reverse of `b`. You are not allowed to use `.equals` or other methods from objects within your implementation. Your allowed to only use methods from the class `String` and `==`.

Write enough useful JUnit-Tests to test your implementation.

Task 9.2: CNTRL+F, CNTRL+V — 3 Points

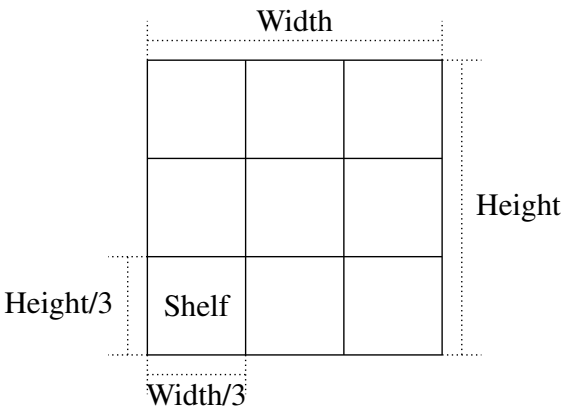
Implement a method `String replace(String source, String search, String replace)` that replaces every (exact) occurences of `search` within `source` with `replace`. Only use `indexOf(String)`, `String substring(int, int)`, `int length()` and `int indexOf(String str, int pos)` from the class `String`. No other methods are allowed to be used.

Test your implementation with enough useful JUnit-Tests.

Task 9.3: Segmentation

6 Points

An entrepreneur wants to automate their storage of goods. In the one they own, there exists different shelf-systems that are divided into segments. Each segment has a height, width, depth and a maximum payload. Furthermore, every segment has 9 shelves. Each shelf can contain one crate.



Different goods are packed into one crate. Implement the class `GoodsCrate` that has the following components:

a) Four `public double`-fields for height, width and depth of a crate, as well as its weight. Furthermore, a constructor that can initialize these four variables with arguments. 1

Implement the class `GoodsSegment` that has the following components:

b) Four `public double`-fields for its height, width, depth and the maximum payload. Furthermore, a one dimensional public array that represents the 9 shelves of the segment in each of which only one crate can be deposited. A constructor that initializes the four fields is also required. It should also initialize every other field that you may need. 1

c) Implement a method `void addCrate(GoodsCrate crate)` that tries to shelf a crate into a free shelf within a segment. A crate should only be shelved if its dimensions are equal to or lower than the dimensions of the shelf and if the maximum payload is not exceeded. You should be aware that crates can be turned around to fit inside the shelf. If a crate is not shelved, a message should be printed to the console. 2

To use the space of every shelf efficiently, you are supposed to implement a class `GoodsShelfSystem` that contains an array of `GoodsSegment`. Initialize that Array with 5 segments of any dimensions (> 9.0) within the constructor and implement the following methods:

d) The method `public boolean findSegmentForCrate(GoodsCrate crate)` that looks for the best storage space for a crate in any segment. The algorithm should look for the best space that fits the crate as best as it can. If a shelf is found, `true` should be returned. `false` otherwise. 2