DATA STRUCTURES AND ALGORITHMS
Extra reading 5

Lect. PhD. Oneț-Marian Zsuzsanna

Babeș - Bolyai University
Computer Science and Mathematics Faculty

2023 - 2024

Content

- This extra reading is actually a problem statement. Inspired from a real-life situation, I have created the requirements for a container. Your task is to think about how you would implemented this container (and if you feel like it, try to actually implement it).

- My solution for the problem will be presented in the next extra reading.

- My washing machine has two buttons (*Turn on* and *Start washing*), a program selector knob and a digital display (and some other buttons below the digital display, which are not important now).
- The washing machine came with a list of predefined programs, that can be used when washing, for example: *Eco 40-60*, *Cotton*, *Synthetics*, *Towels*, *Outdoor*, etc.

Context II

- When the washing machine is turned on, the digital display shows the name of the current program together with details about the program (water temperature, how many times it will rinse the clothes, etc.).

- If I turn the knob to the right, it will show me the details of the next program. If I turn the knob to the left, it will show me the details of the previous program.

- If I keep turning the knob to the right, when it gets to the end of the program list, it will restart from the first one. We have a similar behavior if I keep turning the knob to the left.

- The special part regarding the washing machine is that the order in which the programs are shown is not constant. The first program, shown when the washing machine is turned on, is the *Eco 40-60* program, but the order for the other ones changes: programs used more frequently are at the beginning of the list.

- If I press the *Start* button, the washing starts.

The task

- Assume that you want to implement a *software* for such a washing machine. How would we store the names of the programs?

- **Obs:** This is not a classical *container*. We assume that we have the list of possible programs right from the beginning and no add or remove operations will happen (which are so characteristic for containers).

- The focus of the problem is storing these existing programs in the right order and simulating the behaviour of the program selector knob and the two buttons.

- Let's formally describe how that software should look like.

- We want to implemented the class *WashingMachine*, which stores a fixed list of programs for the washing machine, and when the machine is turned on has a current program which can be displayed.

- Class *WashingMachine* has the following operations:
    - *init* - the constructor. It receives a list of programs, which are the programs that this washing machine has.
    - *turnOn* - operation which turns on the washing machine. At this moment the *Eco 40-60* program is the current program.
    - *start* - start the current program (this will influence the frequency of using the current program)
    - *turnLeft* - turn the program selector knob to the left, which will change the current program to the previous one. If the current program was the first one, it will become the last program.
    - *turnRight* - turn the program selector knob to the right, which will change the current program to the next one. If the current program was the last one, it will become the first program.
    - *getCurrentProgram* - returns the current program (so that it can be displayed).

- The order in which the programs are considered has to be based on how many times the program was started since the creation of the washing machine.

- If two or more programs were started the same number of times, they can appear in any order.

- Assume that you have all the containers (ADTs) discussed in Lecture 3 and 4 already implemented. Which one would you use for implementing the *WachingMachine* and how would you implement each operation?

- Assume that you cannot use containers, you need to work directly with a data structure. How would you implement the *WashingMachine* on a dynamic array and how would you implement it on a linked list? What would be the complexity of the operations?