104824 – INTRODUCTION TO SYSTEMS PROGRAMMING – Spring 2025 ..... Practice Sheet
Practice Sheet 5

This practice sheet is not meant for submission, and will not have any grade. You can ask me about any questions you have, including checking your answers.

Warning Even though this guide will not have any grade, it is important to avoid copying/plagiarism. Avoid the usage of LLMs (such as ChatGPT, DeepSeek, etc). When solving a graded assignment, this warnings will be actual rules that must be followed.

Task 1 Consider project *better-ticket-machine* from chapter 2 of the book *Objects First with Java: A Practical Introduction using BlueJ (Barnes & Kölling)*. Modify the class `TicketMachine` to include pre and postconditions for all methods. Add those specifications as part of the documentation and using exceptions instead of assertions. Propose a class invariant, add it as part of the documentation and implement its validation as a `repOk` method.

Task 2 Consider project *clock-display* from chapter 3 of the book *Objects First with Java: A Practical Introduction using BlueJ (Barnes & Kölling)*. Modify classes `ClockDisplay` and `NumberDisplay` with pre and postconditions for all methods. Add these specifications as part of the documentation, using exceptions instead of assertions to check them. Propose a class invariant for both classes, add them as part of the documentation and implement its validation as a `repOk` method.

Task 3 Consider project *music-organizer-v2* from chapter 4 of the book *Objects First with Java: A Practical Introduction using BlueJ (Barnes & Kölling)*. Modify classes `MusicOrganizer` and `Track` with pre and postconditions for all methods. Add these specifications as part of the documentation, using exceptions instead of assertions to check them. Propose a class invariant for both classes, add them as part of the documentation and implement its validation as a `repOk` method.

Arrays and loops

Task 4 Consider the accompanying code *utils*. Complete the implementations for all methods.

Task 5 Consider if the class should be divided to increase cohesion, do it if you consider it necessary.

Task 6 Should methods require the creation of an object? Justify your answer.

Task 7 If the answer for the previous question was negative, made any necessary modifications.

Task 8 Make a new class to test all methods, try difference scenarios for each.