Proiect

Modelare software - laborator

60% PROJECT- FINAL PRESENTATION

40% LAB

- 7 LABS (3 ABSENCES ALLOWED)
- 4 CHECKPOINTS FOR THE PROJECT * 1P EACH
  - 30% PRESENCE
  - 70% ACTIVITY

Objectives

- Make use of UML to preview the architecture of your system
- Correctly identify the layers of the application
- Learn more OOP
- Apply the learnt design patterns
- Choose the right design pattern
- Respect an adequate coding style

Project #1

Design a parking lot on the following assumptions:

1. The parking lot has multiple levels
2. In the parking lot you can park motorcycles, cars, and buses
3. The parking lot has small spots, compact spots, and large spots
4. A motorcycle can park in any spot
5. A car can park in either a single compact spot or a single large spot
6. A bus can park in five large spots that are consecutive and within the same row

Project #2

Implement your own file system given the following specifications:

1. It should support ten different commands (e.g. referring to file permissions, pwd, cd, ls, add/edit/delete users, etc)
2. Commands that are to be executed will be given as input from a file, one by one on a different line
3. The output of these command will be written to another file

Project #3

Implement an application that emulates the behaviour of a calculator given the following specifications:

1. It should support the following operations: **+, -, *, /, ^, log**, **sqrt, -** (negative numbers)
2. It can contain parentheses that change the order of the operations

Project #4

Design your own Youtube system given the specifications:

1. Users can register the system
2. Users can login or log out the system
3. Users can share, upload, view, comment videos in the system
4. Users can like or dislike the videos, under this condition, the system should be kept a number of likes, dislikes, comments, views to present these number to users

Project #5

Design a caching system for text files based on the specifications:

1. Path to the file is the key; its content is the value
2. You have to control the cache dimension
3. You have to implement a cache replacement policy
4. You should be able to evaluate the performance: cache hit, cache miss