

Relatório - Mini EP 6 - Aplicando *design patterns*

Guilherme Schützer - NUSP 8658544

Ricardo Lira - NUSP 8536131

18/11/2014

1 Preparação

Inicialmente, nós tivemos que decidir em algum programa/código e um Design Pattern para aplicar. Após várias decisões e escolhas, chegamos em um projeto que fizemos ano passado em MAC122: Um IMDB (Internet Movie DataBase) simplificado, que recebe arquivos de texto como entrada com vários filmes e informações adicionais como "nota" e "ano que foi lançado". Junto com essa escolha decidimos utilizar do Design Pattern *Strategy*, focando em uma das características bacanas do programa: Os diferentes meios de ordenação da lista que foram/podem ser implementados.

O programa original possuía duas opções de *sort*, que eram pra ser implementadas por nós: *MergeSort* e *QuickSort*. Com o padrão *Strategy* na cabeça nós decidimos deixar de um jeito mais eficiente a utilização de novos *sorts* no programa, e especialmente caso seja preciso adicionar novos.

O primeiro desafio foi a linguagem. Nosso exercício programa foi feito inicialmente em C, que por não ser uma linguagem orientada a objetos, dificulta um pouco aplicar de forma concreta os conhecimentos e funcionalidades do padrão *Strategy* que vimos em sala de aula. Decidimos então primeiro traduzir o código para uma linguagem que melhor podemos usufruir das vantagens de *design pattern*, e que condiz com nosso curso atual: Python.