

A3 Write-Up

Design Patterns I Made Use Of:

Single Responsibility Principle

- Created classes with single responsibility such as SimilarityFinder, UserUserMatrixPopulator, and SimScoreCalculator.

Abstract Classes

- Shown in the Matrix class.
- This is so the matrices must be either a UserUserMatrix or UserMovieMatrix, and not just a matrix in general.

Generics

- Used in the Matrix classes (and subclasses).
- This is so the matrices used can hold data of any type.

Iterator

- Shown in the MatrixRowIterator class.
- Created this class to iterate through a row in a matrix given a matrix and a row number.

Unit Testing

- Created test cases for each of the classes.

Refactoring:

- Broke down code in Cfiltering into various classes.
- Matrix, UserUserMatrix, and UserMovieMatrix take care of all the matrix related functions, such as holding and printing matrix data.
- MatrixRowIterator provides an easier way to iterate through a matrix's rows.
- Created a class called SimilarityFinder to look store all the data related to the most similar and dissimilar users in the matrix.
- Created a class called SimScoreCalculator to handle any sort of calculation (like distance) for the similarity scores between to users.
- Created a class called UserUserMatrixPopulator to handle the population of the UserUserMatrix.
- Went back to Cfiltering, and changed the methods so that they are simply calling simple, small methods in other classes, rather than having the entire code in each of the large functions in Cfiltering.