Developer documentation

Project name: Cinema Management Software

Author: Vlad Fintina

Programming language: C#

Target framework: .NET 6.0

Every class is stored in one file. The usage of each class is the same as the usage of the file with the same name.

Domain folder – for entities:

Movie.cs – contains Movie: entity class with necessary info of a movie.

Exception folder – developer`s own exceptions

ExistanceExceptions.cs - exceptions regarding whether an entity exists or not in memory.

ValidationException.cs – exception regarding an entity that is not valid.

Repository folder – for repository layer:

InMemoryRepo folder – for repositories that work only with the non-persistent memory.

Repository.cs – contains Repository interface with default methods every other repository needs to implement (addElement, getElements, modifyElement)

InMemoryRepo.cs – contains a repository that implements Repository and works only with the memory.

FileRepo folder – for repositories that work with files:

FileRepository.cs - contains FileRepository:InMemoryRepo which adds some extra feature to InMemoryRepo, especially special methods to read and write data to/from the source file.

Service folder – for service layer:

ServiceInterface.cs – contains ServiceInterface interface which defines every method a service should implement (addMovie, removeMovie, updateMovie, searchMovie, getMovie...)

Service.cs – contains the Service class which implements the ServiceInterface.

Validator folder – for validating entities:

Validator.cs – functional interface with method validate.

BaseValidator.cs – abstract class which implements Validator; validate method is abstract. In addition, it has general methods any validator can use.

MovieValidator.cs – class which extends BaseValidator and validates Movie entity.

Ui folder – for UI layer:

Ui.cs – functional interface with method showUI

Console.cs – implements Ui and creates a console interface

Program.cs – contains class with main method