

#Tecnologias utilizadas:

Java 11
Gradle
IntelliJ
JUnit4

#Arquitetura

O padrão arquitetural escolhido para utilizar neste projeto foi o Model-View-Controller. Como todo software tende a evoluir, escolhi esta arquitetura por ser a mais utilizada na maioria das aplicações para a World Wide Web.

#Arrumação dos pacotes

Os pacotes estão separados da seguinte forma: Controller, Model e Core.

No Core, está a classe principal.

Na Model, encontram-se classes que apenas modelam minhas entidades. Algumas delas, têm uma responsabilidade básica onde, coloco alguma lógica.

No Controller, coloquei toda a minha lógica de negócio para conseguir encontrar o Hotel mais barato.

#Lógica das principais Classes

SimpleClientFactory: Utilizei uma fábrica simples para criação dos meus clientes.

ValidDates: É uma classe que valida apenas datas, podendo talvez, se tornar um singleton, uma vez que pode ser compartilhada por outras classes que não tenham o mesmo propósito do projeto supracitado.

FileReading: Faz apenas a leitura de arquivos presentes em um específico diretório.

ValidateFileLines: Valida as linhas presentes no arquivo.

FindCheapestHotel: Responsável por encontrar o hotel mais barato.

#Funcionamento do meu Projeto

Como já tratado no enunciado do problema, o programa faz a leitura de inputs colocados em um arquivo e devolve ao cliente o hotel mais barato. Neste projeto, é possível fazer a leitura de mais de uma linha, retornando o hotel mais barato para cada uma delas. Caso o usuário entre com cliente que não exista, uma exception será lançada.

#Leitura do Arquivo e modificação do mesmo

Será necessário colocar o arquivo, que já vem dentro da pasta resources, em seu /tmp.

#Testes Unitários

os testes que lidam com os arquivos: FileReadingTest e ValidateFileLinesTest, dependem, exclusivamente de como está seu arquivo atual. Se algo for mudado, esses dois testes irão falhar.

#Como rodar o projeto

Basta, no IntelliJ, ou em sua IDE de preferência, importar "existing files" e escolher o Gradle como gerenciador de dependências.