Instruções de Execução

Reserva de Hotel – Philippe Magno Alves de Menezes

Solução para sistema de reservas de hostéis, a qual deve ser encontrada a melhor opção pelo melhor preço.

Versão do java: "1.8.0_77"
Compilação: mvn clean install

Execução:

caso esteja no diretório principal da aplicação, após realizar o comando acima (maven), executar a aplicação:

```
java target/hotel-reserva-1.0-SNAPSHOT.jar [input_dados]
```

- [input_dados]: o programa espera que os dados sejam enviados diretamente como no arquivo de especificação: Regular: 16Mar2015(seg), 17Mar2015(ter), 18Mar2015(qua) – digitar isso diramente no console.

- Não deve ser colocado tudo entre “ ” : “Regular: 16Mar2015(seg), 17Mar2015(ter), 18Mar2015(qua)”. Pois o programa espera que os dados sejam enviados como um conjunto de dados sobre os dias.

Design da aplicação:

Para o desenvolvimento da solução optei por separar ao máximo as responsabilidades conforme padrão em OO.

Para os **Hotéis** foi criada uma classe modelo (**Hotel.java**). Nela estão contidos todos os atributos dos hotéis (nome, estrelas, valores das diárias), afim de evitar o modelo anêmico de classes também foram inseridos nesta classe, alguns métodos para calcular o valor da diária para o tipo de cliente. Os valores dessas diárias, por não variarem tanto, foram armazenados em um array do tipo integer, que ocupam 4 posições:

1. valor do Dia de Semana, cliente regular
2. valor do Dia de Semana, cliente vip
3. valor do Fim de Semana, cliente regular
4. valor do Fim de Semana, cliente vip

TipoClienteEnum.java: Para representar os tipos de clientes dentro da solução, optei por criar um Enum que representasse os dois estados que este item pode ter: Regular e VIP. Através do método `getFromValue,` que recebe uma string que representa o tipo do cliente, é recuperado o enum referente ao cliente.

HotelRepository.java: Para a listagem dos 3 hotéis foi simulado um padrão chamado repository (também poderia ser chamado de DAO), onde são recuperados de alguma base de dados ou de algum serviço a lista de itens, neste caso a lista de hotéis já configurados.

AlgoritmoBusca.java: Esta é a parte responsável por encontrar o melhor preço de hospedagem entre os hotéis, ela é a responsável por verificar o valor de cada hotel e em caso de empate escolher o que possui a maior quantidade de estrelas. Para realizar este processamento o método

`pesquisarMelhorHotel` recebe o tipo de cliente e as datas de hospedagem deste cliente. Esta classe também possui dependência direta com a classe **HotelRepository**, onde recuperar a listagem dos hotéis.

Utils: No pacote **utils** temos duas classes responsáveis por auxiliar a solução. A classe **DiaSemanaUtil.java**, serve de apoio ao calculo de hospedagem. Nela está contida um **Map** que diz se o dia da semana é dia útil (retorna true) um ou final de semana (retorna false).

Já a classe **StringUtils.java** realiza conversões sobre os parâmetros passados para execução do programa, extraindo o tipo de cliente e convertendo-o para o tipo **TipoClienteEnum.java,** e também extraindo os dias da semana passados como parâmetros. Vale lembrar que não foi optado a conversão das datas para um tipo do Java, como Date ou Calendar, já que o mais importante era saber o nome do dia da semana (segunda, terça …). Sendo assim está classe somente extraí esse dia e retorna como uma lista do tipo String.

Testes: Afim de garantir a melhor qualidade da solução foram realizados vários testes nas classes desenvolvidas dentro do sistema. Para isto foi utilizado o famoso framework para teste de unidade em **Java**, conhecido como **JUnit**. Estes testes estão localizados no diretório **src/test/java** e são executados no momento da compilação da solução.