UNISINOS

Programação Orientada a Objetos

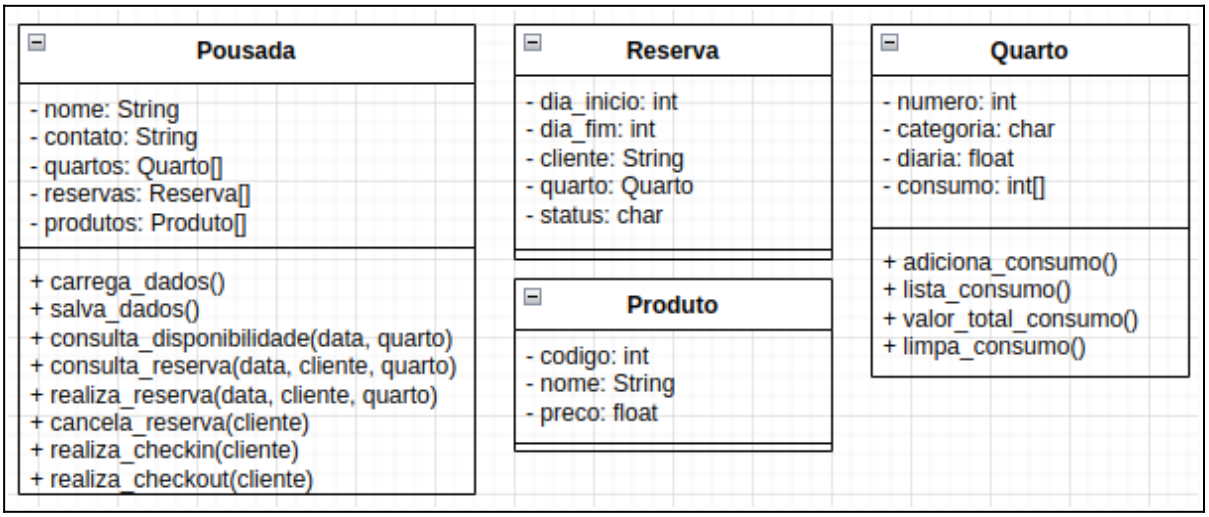
Trabalho Prático do Grau A

Profa. Andriele Busatto do Carmo

Sistema para Gerenciamento de uma Pousada

Objetivo: Desenvolver um programa para gerenciamento básico de uma pousada. Implementar persistência de dados em arquivos texto tabulados.

Estrutura de classes: A estrutura abaixo é uma proposta que serve como ponto de partida para o desenvolvimento do sistema. Adicione novos métodos e atributos caso seja necessário.



Observações:
Os status das reservas são: A/C/I/O - Ativa, Cancelada, Check-In, Check-Out.
As categorias dos quartos são: S/M/P - Standard, Master, Premium.
O array “consumo” da classe “Quarto” é uma lista com os códigos dos produtos.

Descrição:
Ao iniciar o programa, os dados devem ser carregados a partir dos arquivos “pousada.txt”, “quarto.txt”, “reserva.txt” e “produto.txt”. Em seguida, deve mostrar o menu conforme descrição abaixo.

Menu:

Código	Opção	Descrição
1	Consultar disponibilidade	Usuário informa uma data e o número de um quarto, e o programa verifica se o quarto está liberado naquela data específica. Caso esteja liberado, mostrar os dados do quarto. Caso contrário, mostrar mensagem informativa.
2	Consultar reserva	Usuário informa uma data e/ou o nome do cliente e/ou o número de um quarto e o programa deve verificar se existe alguma reserva ativa com os dados informados. Caso exista, mostrar a data inicial e final, nome do cliente e dados do quarto para cada reserva que satisfaça os parâmetros informados. Caso contrário, mostrar mensagem informativa.

3	Realizar reserva	Usuário informa uma data inicial e final, o nome do cliente e o número do quarto. O programa deve verificar a disponibilidade do quarto no período solicitado, e somente realizar a reserva caso o quarto esteja disponível e o cliente não tenha outra reserva ativa ou em check-in. Mostrar mensagem informativa de sucesso ou falha.
4	Cancelar reserva	Usuário informa o nome do cliente e o programa marca a reserva ativa do cliente como cancelada. Caso não exista reserva ativa para o cliente, mostrar mensagem informativa.
5	Realizar check-in	Usuário informa o nome do cliente e o programa verifica se existe reserva para ele. Caso exista, registrar o check-in e mostrar a data inicial e final, a quantidade de dias reservados, o valor total das diárias e os dados do quarto. Caso contrário, mostrar mensagem informativa.
6	Realizar check-out	Usuário informa o nome do cliente e o programa verifica se existe check-in ativo para ele. Caso exista, mostrar a data inicial e final, quantidade de dias, o valor total das diárias, a lista de consumo da copa (produto e valor), o valor total dos consumos e o valor final a ser pago (diárias + consumo). Em seguida, registrar o check-out nas reservas e limpar a lista de consumo da copa. Caso não exista reserva em check-in para o usuário, mostrar mensagem informativa.
7	Registrar consumo	Usuário informa o nome do cliente e o programa verifica se existe check-in ativo para ele. Caso exista, mostrar a lista dos produtos e valores disponíveis na copa. Usuário informa o produto desejado e o programa registra o consumo. Caso não exista reserva em check-in para o usuário, mostrar mensagem informativa.
8	Salvar	Salva os dados das reservas e quartos em arquivos texto tabulado. Deve excluir as reservas canceladas ou com check-out realizado.
0	Sair	Encerra o sistema. Caso algum dado não tenha sido salvo em arquivo, salvar automaticamente.

Observações importantes:

1. Todas as opções do menu devem ser implementadas. A não implementação de alguma opção acarretará um desconto na nota final do grupo.

2. Os nomes de classes, atributos e métodos especificados acima na estrutura das classes devem ser mantidos na implementação do código (ou seja, não os renomeie). Novos métodos e atributos devem ser nomeados de acordo com a sua respectiva função.

3. Se necessário, implemente métodos *getters* e *setters* para os atributos das classes. Métodos *setters* devem verificar o valor sendo setado ao atributo quando necessário.

4. Para padronizar a persistência de dados, implementar os métodos “serializar()” e “deserializar()”.

5. A conversão de um objeto em texto para visualização do usuário deve ser feita no método “toString()”.

Fique atento(a) aos seguintes itens:

a) Comente o seu código, informando no mínimo o que cada método de suas classes faz. Não esqueça de colocar seu nome em TODOS os arquivos que você for implementar.
b) Indente seu código. Códigos não indentados são mais difíceis de entender, difíceis de encontrar erros, além de receberem desconto na nota final.
c) Utilize impressões na tela para informar o que está acontecendo no programa.
d) Escreva seu código com clareza.

Outras informações:

- O trabalho deverá ser implementado utilizando o Visual Studio Code. Para utilizar outra IDE, fale **ANTES** com a professora, para verificar a possibilidade ou não.
- Deve ser realizado **em dupla**.
- Trabalhos copiados de colegas e/ou não realizados pelo aluno receberão zero (todos os envolvidos).

Entrega e apresentação:

- Você deve enviar um ZIP com o projeto do trabalho pelo Moodle até as **19h** do dia **02/05/2024** e apresentar o trabalho somente após a entrega no Moodle.
- Apenas um integrante da dupla realiza a entrega.
- Identifique claramente a dupla no nome do projeto (nome completo).
- Trabalhos em atraso não serão aceitos para avaliação.
- A apresentação será realizada em aula, para a professora, no dia **02/05/2024**.
- O tempo e a ordem das apresentações serão definidos na semana anterior à data de entrega.

Avaliação:

- Antes de mais nada:

```
if código executa com erros :
      print(“Sem avaliação...”)
      nota = zero
else
      print(“O que você fez será avaliado!”)
```

- O programa deve ser todo orientado a objetos.
- A função `main` deve conter apenas a manipulação do menu.
- O código-fonte deve estar corretamente identado e comentado.
- Os nomes dos atributos, métodos, parâmetros e variáveis utilizados devem ser autoexplicativos, utilizando a notação “snake_case” e “PascalCase”.
- Devem ser utilizados todos os comandos e conceitos especificados na definição desse trabalho (incluindo orientação a objetos, vetores, manipulação de arquivos, strings, comandos de seleção e comandos de repetição).
- Todas as funcionalidades do programa contidas nesta definição devem ser implementadas.
- Todos os componentes do grupo devem explicar uma parte do trabalho durante a apresentação.

- Os itens do trabalho serão avaliados da seguinte forma:
  - *Código comentado*: **desconto de 1,0 ponto por classe** não comentada.
  - *Código indentado*: **desconto de 1,0 ponto por classe** não indentada.

- *Apresentação para a professora*: apresentações confusas, mal explicadas, erradas e/ou que demonstrem que não foi o(a) aluno(a) que desenvolveu o trabalho, acarretarão em **descontos na nota final**.

- *Classes conforme enunciado*: **2,5 pontos**.
- *Interface com o usuário (mensagens e criatividade)*: **1,5 pontos**.
- *Corretude do programa (lógica, ações, resultados)*: **6,0 pontos**.

Dicas:

- Faça um trabalho simples, mas correto. Não é necessário criar um programa com centenas de linhas de código. Mais vale um programa que funciona e é simples do que um que é demasiadamente complexo e apresenta falhas.
- Implemente e vá testando suas classes. Não deixe para verificar se funcionam no final de toda a implementação. Se existirem erros, serão mais difíceis de serem encontrados.
- Pense no programa antes de começar a implementar. Comece a codificar somente quando entender exatamente como ele deve funcionar.
- Não invente regras. Elas estão todas claras no decorrer deste documento.
- Não se assuste com a primeira impressão do trabalho. Você tem plenos conhecimentos e total capacidade de implementá-lo. Basta calma e atenção.
- Não deixe para fazer na última semana.
- Reserve um tempo por dia para pensar no trabalho e implementar alguma parte.

Boa sorte e bom trabalho!

Créditos: descrição para o Sistema de Gerenciamento de Pousada contruída com base na descrição original do Prof. Márcio Garcia Martins (Unisinos).