



Anderson Rodrigues dos Santos
Trabalho de POO
19/10/2023

Introdução

O programa consiste um sistema de agendamento de consultas de clinicas hospitalares, que facilita a organização do consultório, além disso, mantém os cadastros dos médicos e pacientes. Além da criação de cadastros de médicos, pacientes e consultas, também tem como removê-los e consultá-los.

Decisões de Implementação

A linguem escolhida, por ser um programa simples que não precisa muito de complexidade, é Java. A implementação necessitou de cinco classes: Pessoa, Medico, Paciente, Consulta e Consultório.

Classe Pessoa:

- Descrição: A classe Pessoa é uma classe base que contém atributos e métodos comuns a Medico e Paciente.
- Atributos:
 - nome (string): O nome da pessoa.
 - endereço (string): O endereço da pessoa.
 - sexo (char): O sexo da pessoa.
 - telefone (int): O número de telefone da pessoa.
 - identidade (int): O número de identidade da pessoa.
 - cpf (string): O CPF (Cadastro de Pessoas Físicas) da pessoa.
- Métodos:
 - Construtor (com valores para inicialização): Inicializa os atributos com os valores fornecidos.
 - Construtor (sem valores de inicialização): Permite criar uma instância da classe sem atribuir valores iniciais.
 - Getters e Setters: Métodos para acessar e modificar os atributos privados.
 - Imprimir(): Exibe os valores dos atributos privados.

Classe Medico:

- Descrição: A classe Medico é derivada da classe Pessoa e representa um médico.
- Atributos:
 - crm (int): O número de registro do médico no Conselho Regional de Medicina.
 - especialidade (String): A especialidade médica do médico.
- Métodos:
 - Construtor (com valores para inicialização): Inicializa os atributos com os valores fornecidos.
 - Construtor (sem valores de inicialização): Permite criar uma instância da classe sem atribuir valores iniciais.
 - Imprimir(): Exibe os detalhes do médico, incluindo seu nome, endereço, sexo, telefone, identidade, CPF, CRM e especialidade.
 - Getters: Métodos para acessar os atributos do médico, incluindo CRM e especialidade.
 - Setters: Métodos para definir os atributos do médico, como CRM e especialidade.

Classe Paciente:

- Descrição: A classe Paciente também é derivada da classe Pessoa e representa um paciente.
- Atributos adicionais:
 - relato (string): Descrição do relato do paciente.
 - medicacaoConsumidaRegularmente (string): Informação sobre a medicação que o paciente consome regularmente.
- Métodos adicionais:
 - Construtor (com valores para inicialização): Inicializa os atributos com os valores fornecidos.
 - Construtor (sem valores de inicialização): Permite criar uma instância da classe sem atribuir valores iniciais.
 - Imprimir(): Este método exibe os detalhes do paciente, incluindo seu nome, endereço, sexo, telefone, identidade, CPF, relato e informações sobre a medicação que consome regularmente.
 - Getters: Métodos para acessar os atributos do paciente.
 - Setters: Métodos para definir os atributos do paciente.

Classe Consulta:

- Atributos:
 - data (String): A data da consulta.
 - hora (String): A hora da consulta.
 - cpfPaciente (String): O CPF do paciente que agendou a consulta.
 - crmMedico (int): O CRM do médico responsável pela consulta.
 - id(int): É o código de identificação da consulta

- Métodos:
 - Construtor (com valores para inicialização): Inicializa os atributos com os valores fornecidos.
 - Construtor (sem valores de inicialização): Permite criar uma instância da classe sem atribuir valores iniciais.
 - Imprimir(): Exibe os detalhes da consulta, incluindo data, hora, CPF do paciente e CRM do médico.
 - Getters: Métodos para acessar os atributos da consulta.
 - Setters: Métodos para definir os atributos da consulta.

Classe Consultorio:

- Atributos:
 - medicos (list): Uma lista de médicos disponíveis no consultório.
 - pacientes (list): Uma lista de pacientes cadastrados no consultório.
 - consultas (list): Uma lista de consultas agendadas.
 - id(int): O código que é atribuído na classe consulta
- Métodos:
 - Construtor (sem valores de inicialização): Permite criar uma instância da classe sem atribuir valores iniciais.
 - cadastrarPaciente(Paciente p): Adiciona um paciente à lista de pacientes do consultório.
 - removerPaciente(String cpf): Remove um paciente da lista com base no CPF.
 - cadastrarMedico(Medico m): Adiciona um médico à lista de médicos do consultório.
 - removerMedico(int crm): Remove um médico da lista com base no CRM.
 - VerificarId(): Verifica se o id chegou o numero limite(700), se sim atribua o valor de id igual zero, se não acrescente mais um no valor atual.
 - cadastrarConsulta(Consulta c): Agenda uma consulta e adiciona à lista de consultas.
 - removerConsulta(int cpfPaciente, int crmMedico, int id): Remove uma consulta com base no CPF do paciente , no CRM do médico e o id da Consulta.
 - imprimirListaPacientes(): Exibe a lista de pacientes cadastrados.
 - imprimirListaMedicos(): Exibe a lista de médicos disponíveis.
 - imprimirListaConsultas(): Exibe a lista de consultas agendadas.
 - imprimirConsulta(String cpfPaciente, int crmMedico, int id): Exibe uma consulta especifica na lista de consultas.

Classe APP:

É onde o método main se encontra e é o ponto de entrada do programa, onde a execução do programa começa. É aqui que você pode instanciar objetos das outras classes, como Medico, Paciente, Consulta e Consultório, e chamar métodos para interagir.

Casos de teste e Resultados dos testes realizados

```

1. Cadastrar Médico
2. Cadastrar Paciente
3. Cadastrar Consulta
4. Imprimir Dados dos Médico
5. Imprimir Consultas
6. Imprimir Dados dos Paciente
7. Imprimir Dados uma Consulta Especifica
8. Remover Médico
9. Remover Paciente
10. Remover Consulta
11. Sair

Escolhe uma opcao: █

---

Escolhe uma opcao: 1

Nome: Anderson
Endereco: cefet
Cpf: 00000
Especialidade: Engenheiro
Identidade: 000
Sexo (M/H): H
Telefone: 3788
CRM: 100

Medico Cadastrado

1. Cadastrar Médico
2. Cadastrar Paciente
3. Cadastrar Consulta
4. Imprimir Dados dos Médico
5. Imprimir Consultas
6. Imprimir Dados dos Paciente
7. Imprimir Dados uma Consulta Especifica
8. Remover Médico
9. Remover Paciente
10. Remover Consulta
11. Sair

Escolhe uma opcao: █

```

É o menu principal quando é inicializado o programa. O usuário escolherá um opção que é classificado por um número.

Ex.: 1 no Cadastrar Médico

Encolhendo a opção 1, direciona o usuário para preencher o questionário de Cadastrar Médico com as informações como nome, sexo e cfp e quando o usuário digitar algo errado que é diferente do que pede, o programa retorna na linha errada para o usuário repetir o processo. Quando finaliza o Cadastro, o sistema salva os dados na lista de médicos e volta para o menu principal.

```
Escolhe uma opcao: 2

Nome: Lucas
Endereco: Divinopolis
Cpf: 11111
Relato: EStudante
Medicação Consumida Regulamente: Remedio para gripe
Identidade: 111
Sexo (M/H): h
Telefone: tt
Telefone: 2345

Paciente Cadastrado

1. Cadastrar Médico
2. Cadastrar Paciente
3. Cadastrar Consulta
4. Imprimir Dados dos Médico
5. Imprimir Consultas
6. Imprimir Dados dos Paciente
7. Imprimir Dados uma Consulta Especifica
8. Remover Médico
9. Remover Paciente
10. Remover Consulta
11. Sair

Escolhe uma opcao: █
```

Encolhendo a opção 2, direciona o usuário para preencher o questionário de Cadastrar Paciente com as informações como nome, sexo e relato e quando o usuário digitar algo errado que é diferente do que pede, o programa retorna na linha errada para o usuário repetir o processo, como é representado linha Telefone. Quando finaliza o Cadastro, o sistema salva na lista de pacientes e volta para o menu principal.

```
Escolhe uma opcao: 3

Data: 10/10
Hora: 10:10
Cpf do Paciente: 10
Crm do Médico: 11

Medico ou Pacinete não Cadastrado
- - - - -

Escolhe uma opcao: 3

Data: 10/10
Hora: 10:00
Cpf do Paciente: 11111
Crm do Médico: 100

Consulta Cadastrado
ID: 0
Data: 10/10
Hora: 10:00
CPF do Paciente:11111
CRM do Medico: 100
```

Na opção 3, direcionara o usuário para preencher o questionário de Cadastrar Consulta com as informações com data, hora , cpf e Crm. Se o cpf ou Crm informado do paciente ou medico não estiver cadastrado no sistema o cadastramento não ocorrera e voltara para o menu principal. Este ocorrera também quando os informações do Cadastro já tenha sido cadastrado e, além disso, quando a hora e data do medico já tenha sido utilizado.

Se caso o cpf ou Crm informado do paciente ou medico estiverem cadastrado no sistema, o cadastro ocorrera e salvara na lista de consultas

Escolhe uma opcao: 4

Nome: Anderson
Sexo: H
CPF: 00000
Endereco: cefet
Telefone: 3788
Identidade: 0
CRM: 100
Especialidade: Engenheiro

Escolhe uma opcao: 5

ID: 0
Data: 10/10
Hora: 10:00
CPF do Paciente:11111
CRM do Medico: 100

Escolhe uma opcao: 6

Nome: Lucas
Sexo: h
CPF: 11111
Endereco: Divinopolis
Telefone: 2345
Identidade: 111
Relato: Estudante
Medicação Consumida Regulamente: Remedio para gripe

Escolhe uma opcao: 7

Digite o CPF do Paciente: 11111
Digite o CRM do Médico: 100
Digite o Id da consulta: 0

ID: 0
Data: 10/10
Hora: 10:00
CPF do Paciente:11111
CRM do Medico: 100

Escolhe uma opcao: 7

Digite o CPF do Paciente: 35
Digite o CRM do Médico: 43
Digite o Id da consulta: 6

Consulta não encontrado!

Nas opções 4, 5 e 6 exibirá os médicos, consultas e pacientes cadastrados. E quando qualquer uma estiver vazia, o Sistema informara.

Na opção 7, o usuário respondera as perguntas para encontrar a uma consulta especifica. Se o dados forem forem digitados corretamente, mostrar as informações da consulta, se não exibira a mensagem "Consulta não encontrada"

```
Escolhe uma opcao: 8
Digite o Crm do Médico: 100
Medico Removido

Escolhe uma opcao: 9
Digite o Cpf do Paciente: 11111
Paciente Removido

Escolhe uma opcao: 10
Digite o Cpf do Paciente: 11111
Digite o Crm do Médico: 100
Digite o Id da consulta: 0
Consulta Removido
```

Nas opções 8, 9 e 10, quando o usuário responde as respectivas questões, excluirá as informações da opção desejada. Caso o usuário informar os parâmetros errados, exibirá uma mensagem que não foi possível encontrar.

```
Escolhe uma opcao: 11
```

E a opção 11 sairá do programa.

Os testes funcionaram corretamente sem erros ou falhas.

Análise dos Resultados

Um dos principais motivos que teve um resultado satisfatório foi a implementação do tratamento de exceção no entradas das variáveis de números, como por exemplo no recebimento no CRM do medico, por ser um int, é sucessível ao erro quando o usuário digitar qualquer coisa diferente de numero.