

Exercício (20 pontos)

Data de entrega: 20/10/2022

A entrega deve ser feita pelo SIGAA até as 23:59 do dia 20/10/2022

1. Um consultório solicitou a criação de um sistema para o seu consultório. A ideia é que o sistema permita armazenar os dados do consultório, paciente, médicos (já que mais de um médico irá atender no mesmo consultório) e as consultas marcadas. Após uma reunião com o analista de sistemas responsável, foi proposto o desenvolvimento do sistema representado pelas classes abaixo:

Classe Pessoa

Atributos:

nome: string,
sexo: char,
endereço:string,
cpf: string,
telefone: int,
identidade: int;

Métodos:

imprimir()
getters
setters

Classe Medico (classe filha da classe Pessoa)

Atributos:

crm: int,
especialidade: string;

Métodos:

imprimir()
getters
setters

Classe Paciente (classe filha da classe Pessoa)

Atributos:

relato: string,
medicacaoConsumidaRegularmente: string;

Métodos:

imprimir()

Classe Consulta

Atributos:

data: String
hora: string
cpfPaciente: string,
crmMedico: int

Metodos:

Imprimir()
Getters
Setters

Classe Consultorio

Atributos:

medicos:lista,
paciente:lista,
consulta:lista,
contPaciente: int,
telefone: int,
endereco: string,
nome: string

Metodos

cadastrarPaciente(Paciente p)
removerPaciente(string cpf)
cadastrarMedico(Medico m)
removerMedico(int crm)
cadastrarConsulta(Consulta C)
removerConsulta(int cpfPaciente, int crmMedico)
imprimirListaPacientes()
imprimirListaMedicos()
imprimirListaCosultas()

Faça um diagrama de classes que represente o sistema que você irá implementar.

O sistema deve utilizar listas alocadas dinamicamente

Observação 1: O main deve ser implementado de forma que ele permita que o usuário digite:

- 1 para cadastrar um médico.
- 2 para cadastrar um paciente.
- 3 para cadastrar uma consulta.

- 4 para imprimir dados do médico.
- 5 para imprimir consultas.
- 6 para imprimir dados do paciente
- 7 para imprimir data, hora, paciente e médico de uma consulta em específico
- 9 para remover médico
- 10 para remover paciente
- 11 para remover consulta
- 12 para sair

Observação 2: Caso seja necessário adicionar novos métodos ou atributos eles podem ser adicionados e o diagrama deve ser corrigido.

Observação 3: Deve ser apresentada uma documentação do programa contendo:

- Introdução
- Decisões de implementação
- Casos de teste
- Resultados dos testes realizados
- Análise dos resultados