Prova II

Disciplina: Laboratório de Programação Orientada a Objeto - POO

Período: 3°

Valor da Prova: 07 Pontos

Professor: Diego Ramos Inácio

Curso de Engenharia de Software Município: Saquarema – RJ

Definições:

Desenvolvimento de Sistema em Programação Orientada a Objeto

Introdução:

A programação orientada a objetos (POO) é fundamental no desenvolvimento de software, oferecendo uma estrutura modular e reutilizável. Ao representar entidades do mundo real através de classes e objetos, facilita a organização, implementação e manutenção do código. Conceitos como encapsulamento e abstração promovem a coesão e escalabilidade, evitando dependências desnecessárias. Técnicas avançadas como herança, abstração e encapsulamento, além da relação com interfaces, otimizam o desempenho, permitindo a reutilização e adaptação de código. Em resumo, a POO melhora a organização, desempenho e escalabilidade do software, resultando em sistemas mais robustos e flexíveis, trazendo benefícios a longo prazo para projetos e usuários.

ATENÇÃO: Produtos para serem entregues os arquivos .py e .java como forma de anexo no AVA.

Demanda: Você foi convidado(a) a apresentar uma ideia de desenvolvimento de um sistema utilizando o paradigma de programação orientada a objetos (POO) em uma importante reunião de equipe, com foco no contexto de um sistema de interfone. Esse convite é uma prova do reconhecimento do seu trabalho e da confiança depositada em você pela equipe.

O desenvolvimento de um sistema em POO desempenha um papel crucial no sucesso de um interfone moderno. Ele pode influenciar diretamente a eficiência na comunicação entre moradores e visitantes, a integração com sistemas de segurança, o gerenciamento de acessos, além de facilitar a expansão do sistema para novas funcionalidades, como registro de chamadas ou controle remoto via aplicativos.

Ao apresentar uma proposta de desenvolvimento de sistema bem estruturada e eficiente, você contribuirá para a criação de um interfone robusto e escalável, garantindo uma comunicação mais ágil e segura, além de proporcionar uma experiência moderna e prática para os usuários finais.

Questão 1: Criação da Classe Base e Subclasse para o Sistema de Interfone (2 pontos) Python

Você deverá desenvolver um sistema para um interfone inteligente utilizando os conceitos de herança e abstração no paradigma de programação orientada a objetos (POO).

Requisitos:

1. **Classe Base Interfone (Abstrata)** Crie uma classe base abstrata chamada Interfone, que será utilizada como modelo para diferentes tipos de interfones. Esta classe deve conter:
    - **Atributos:**
        - **modelo:** uma string que identifica o modelo do interfone.
        - **endereco:** uma string com o endereço do local onde o interfone está instalado.
    - **Métodos Abstratos:**
        - **ligar_para_proprietario():** Um método abstrato que deve ser implementado por subclasses para simular o interfone ligando para o celular do proprietário.
        - **registrar_chamada():** Um método abstrato que registra o horário e a duração da chamada realizada.
2. **Subclasse InterfoneResidencial** Crie uma subclasse chamada InterfoneResidencial, que herda de Interfone, e implemente os métodos abstratos:
    - **Atributos**
        - **nome_proprietario:** uma string que identifica o nome do proprietário
            1. *proprietário deve ter duas opções para ligar*
        - **telefone_proprietario:** uma string que identifica o número dos proprietários
            1. *proprietário deve ter duas opções para ligar*
    - **metodo_cadastro():** Criar um método de cadastro usando @classmethod para instanciar a class.
    - **ligar_para_proprietario():** Exiba uma mensagem simulando a ligação para o celular do proprietário que mostre a escolha do proprietário. Por exemplo:
        - "Ligando para o número cadastrado do proprietário “nome_Proprietário” no endereço “endereço_do_proprietario”".
    - **registrar_chamada():** Simule o registro do horário e da duração da chamada, exibindo uma mensagem com essas informações. Por exemplo:
        - "Chamada registrada: Horário 14:35, Duração: 02 minutos e 15 segundos.".

Questão 2: Instanciando as Classes e Realizando Interações com Várias Opções de Cadastro (2 pontos) Python

Nesta questão, você deve expandir o sistema de interfone criado na Questão 1 para incluir a possibilidade de cadastrar números para serviços de emergência, como polícia, bombeiros e SAMU, além de simular a interação para realizar uma ligação. Instanciando as classes criadas na Questão 1.

Resposta da 1 e 2

```
import time
from datetime import datetime, timedelta


# Questão 1
# Classe Base
class Interfone:
    def __init__(self, endereco):
        self.modelo = "Intelbras"
        self.endereco = endereco

    # Métodos abstratos
    def ligar_para_proprietario(self):
        raise NotImplementedError("Este método deve ser implementado pelas subclasses.")

    def registrar_chamada(self):
        raise NotImplementedError("Este método deve ser implementado pelas subclasses.")

# Questão 2
# Subclasse para Interfones Residenciais
class InterfoneResidencial(Interfone):
    def __init__(self, endereco, nome_proprietario, telefone_proprietario):
        super().__init__(endereco)
```

```
        self.nome_proprietario = nome_proprietario
        self.telefone_proprietario = telefone_proprietario

    @classmethod
    def metodo_cadastro(cls):
        endereco = input("Digite o endereço do interfone: ")
        apto = input("Digite o numero do apto: ")
        proprietario1 = input("Digite o nome do proprietário 1: ")
        proprietario2 = input("Digite o nome do proprietário 2: ")
        telefone1 = input("Digite o número do proprietário: ")
        telefone2 = input("Digite o número do proprietário: ")
        return cls(endereco, {apto:[proprietario1, proprietario2]}, {apto:[telefone1, telefone2]})

    def ligar_para_proprietario(self, apto, n):
        try:
            proprietarios = self.nome_proprietario[apto]
            telefones = self.telefone_proprietario[apto]
            if 1 <= n <= len(proprietarios):
                print(f"Ligando para {proprietarios[n-1]} no telefone {telefones[n-1]} (Apto: {apto}, Endereço: {self.endereco}).")
            else:
                print("Número do proprietário inválido.")
        except KeyError:
            print(f"Apto {apto} não encontrado.")

    def registrar_chamada(self):
        # Registrar o horário de início
        horario_inicio = datetime.now()
        print(f"Início da chamada: {horario_inicio.strftime('%H:%M:%S')}")
```

```
        # Simular a duração da chamada
        print("Ligação em andamento...")
        time.sleep(5)  # Aguarda 5 segundos para simular a ligação

        # Registrar o horário de término
        horario_fim = datetime.now()
        print(f"Término da chamada: {horario_fim.strftime('%H:%M:%S')}")

        # Calcular a duração
        duracao = horario_fim - horario_inicio
        minutos, segundos = divmod(duracao.total_seconds(), 60)
            print(f"Chamada registrada: Horário {horario_inicio.strftime('%H:%M:%S')}, Duração: {int(minutos)} minutos e {int(segundos)} segundos.")

# Questão 3
# Subclasse para Interfones Emergenciais
class InterfoneEmergencial(Interfone):
    def __init__(self, modelo, endereco, tipo_servico, numero_servico):
        super().__init__(endereco)
        self.modelo = modelo  # Agora a classe aceita o modelo como argumento
        self.tipo_servico = tipo_servico
        self.numero_servico = numero_servico

    @classmethod
    def metodo_cadastro(cls):
        modelo = input("Digite o modelo do interfone: ")
        endereco = input("Digite o endereço do interfone: ")
        tipo_servico = input("Digite o tipo de serviço (Polícia, Bombeiros, SAMU, etc.): ")
```

```
        numero_servico = input("Digite o número do serviço: ")
        return cls(modelo, endereco, tipo_servico, numero_servico)

    def ligar_para_proprietario(self):
        print(f"Ligando para o serviço de emergência {self.tipo_servico} no número {self.numero_servico}.")

    def registrar_chamada(self):
        # Registrar o horário de início
        horario_inicio = datetime.now()
        print(f"Início da chamada: {horario_inicio.strftime('%H:%M:%S')}")

        # Simular a duração da chamada
        print("Ligação em andamento...")
        time.sleep(5)  # Aguarda 5 segundos para simular a ligação

        # Registrar o horário de término
        horario_fim = datetime.now()
        print(f"Término da chamada: {horario_fim.strftime('%H:%M:%S')}")

        # Calcular a duração
        duracao = horario_fim - horario_inicio
        minutos, segundos = divmod(duracao.total_seconds(), 60)
        print(f"Chamada registrada: Horário {horario_inicio.strftime('%H:%M:%S')}, Duração: {int(minutos)} minutos e {int(segundos)} segundos.")

# Uso das classes com inputs
if __name__ == "__main__":
    # Cadastro do interfone residencial
    print("\nCadastro de Interfone Residencial")
```

interfone_residencial = InterfoneResidencial.metodo_cadastro()

interfone_residencial.ligar_para_proprietario('201', 1)

interfone_residencial.registrar_chamada()

 # Cadastro de interfones emergenciais

print("\nCadastro de Interfone Emergencial - Polícia")

interfone_policia = InterfoneEmergencial.metodo_cadastro()

interfone_policia.ligar_para_proprietario()

interfone_policia.registrar_chamada()

Questão 3: Implementação do Sistema de Interfone em Java (2 pontos) Java

Nesta questão, você deverá replicar a funcionalidade do sistema de interfone inteligente criado em Python nas Questões 1 e 2, agora utilizando a linguagem Java.

Requisitos

1. **Classe Base Interfone (Abstrata):**
    - **Crie uma classe abstrata chamada Interfone com os seguintes atributos:**
        - **modelo (String):** Representa o modelo do interfone.
        - **endereco (String):** Indica o endereço onde o interfone está instalado.
    - **Inclua os métodos abstratos:**
        - **void ligarPara(String nome, String numero):** Simula uma ligação.
        - **void registrarChamada(String nome, String horario, String duracao):** Registra o horário e a duração da chamada.
2. **Subclasse InterfoneResidencial:**
    - **Implemente a classe InterfoneResidencial, que herda de Interfone.**
    - **Adicione o método void cadastrarContato(String nome, String numero),** que exibe uma mensagem confirmando o cadastro do contato.
    - **Implemente os métodos abstratos:**
        - ligarPara(String nome, String numero): Exibe uma mensagem simulando a ligação para o contato.
        - registrarChamada(String nome, String horario, String duracao): Exibe detalhes

da chamada.

3. **Simulação no Método main:**
    - Instancie a classe InterfoneResidencial.
    - Cadastre os seguintes contatos:
        - Proprietário.
        - Polícia (190).
        - Bombeiros (193).
        - SAMU (192).
    - Realize ligações para cada contato e registre as chamadas.

Resposta da Questão 3

```
// Classe Base Interfone (Abstrata)
abstract class Interfone {
    // Atributos
    protected String modelo;
    protected String endereco;

    // Construtor
    public Interfone(String modelo, String endereco) {
        this.modelo = modelo;
        this.endereco = endereco;
    }

    // Métodos Abstratos
    public abstract void ligarPara(String nome, String numero);
    public abstract void registrarChamada(String nome, String horario, String duracao);
}
```

```
// Subclasse InterfoneResidencial
class InterfoneResidencial extends Interfone {
    // Atributos para os contatos e números
    private String contato1;
    private String numero1;

    private String contato2;
    private String numero2;

    private String contato3;
    private String numero3;

    private String contato4;
    private String numero4;

    // Construtor
    public InterfoneResidencial(String modelo, String endereco) {
        super(modelo, endereco);
        // Inicializa os contatos e números como vazios
        this.contato1 = "";
        this.numero1 = "";
        this.contato2 = "";
        this.numero2 = "";
        this.contato3 = "";
        this.numero3 = "";
        this.contato4 = "";
        this.numero4 = "";
    }
```

```
// Método para cadastrar contatos
public void cadastrarContato(int index, String nome, String numero) {
    if (index == 1) {
        this.contato1 = nome;
        this.numero1 = numero;
    } else if (index == 2) {
        this.contato2 = nome;
        this.numero2 = numero;
    } else if (index == 3) {
        this.contato3 = nome;
        this.numero3 = numero;
    } else if (index == 4) {
        this.contato4 = nome;
        this.numero4 = numero;
    }
    System.out.println("Contato cadastrado: " + nome + " - Número: " + numero);
}

// Implementação do método ligarPara
@Override
public void ligarPara(String nome, String numero) {
    System.out.println("Ligando para " + nome + " no número " + numero);
}

// Implementação do método registrarChamada
@Override
public void registrarChamada(String nome, String horario, String duracao) {
    System.out.println("Chamada registrada: ");
```

```
        System.out.println("Nome: " + nome);
        System.out.println("Horário: " + horario);
        System.out.println("Duração: " + duracao);
    }

    // Método para realizar chamadas para todos os contatos
    public void realizarChamadas() {
        if (!contato1.isEmpty()) {
            ligarPara(contato1, numero1);
            registrarChamada(contato1, "10:00:00", "5 minutos");
        }
        if (!contato2.isEmpty()) {
            ligarPara(contato2, numero2);
            registrarChamada(contato2, "10:05:00", "4 minutos");
        }
        if (!contato3.isEmpty()) {
            ligarPara(contato3, numero3);
            registrarChamada(contato3, "10:10:00", "3 minutos");
        }
        if (!contato4.isEmpty()) {
            ligarPara(contato4, numero4);
            registrarChamada(contato4, "10:15:00", "6 minutos");
        }
    }
}

// Classe principal
public class SistemaInterfone {
    public static void main(String[] args) {
```

```
        // Instanciando o interfone residencial
        InterfoneResidencial interfone = new InterfoneResidencial("Intelbras", "Rua Exemplo, 123");

        // Cadastrando contatos
        interfone.cadastrarContato(1, "Proprietário", "1234-5678");
        interfone.cadastrarContato(2, "Polícia", "190");
        interfone.cadastrarContato(3, "Bombeiros", "193");
        interfone.cadastrarContato(4, "SAMU", "192");

        // Realizando chamadas para os contatos cadastrados
        interfone.realizarChamadas();
    }
}
```