PUC Minas

Pontifícia Universidade Católica de Minas Gerais
Bacharelado em Engenharia de Software – Campus Coração Eucarístico
Programação Modular - Semestre 2025/2
Prof. Daniel Kansaon
Atividade II

Atividade 1. Explique, com suas palavras, cada um dos cinco princípios SOLID e dê um exemplo de onde cada princípio seria útil em um projeto real.

Atividade 2. Explique, com suas palavras:

- O que é herança em orientação a objetos?
- O que é polimorfismo?
- Como a substituição de métodos (override) está relacionada ao polimorfismo?

Atividade 3. Explique o que é um enum em Java. Em que situações o uso de enum é mais adequado do que o uso de constantes (static final)?

Atividade 4. Explique para que serve a API Stream em Java. Cite 3 operações comuns usadas em streams e o que cada uma faz.

Atividade 5. Abaixo estão listados alguns dos principais padrões de projeto utilizados em desenvolvimento orientado a objetos. Para cada um deles, escreva uma descrição com suas palavras, incluindo qual problema o padrão resolve e um exemplo de situação em que o padrão pode ser aplicado.

A. Singleton
B. Factory Method
C. Strategy
D. Decorator
E. Observer
F. Chain of Responsibility
G. Facade
H. Adapter

Atividade 6. O Java Collections Framework oferece diversas estruturas de dados para armazenar e manipular conjuntos de objetos. Com base no que foi estudado em aula, escreva um pequeno texto sobre cada uma das coleções abaixo, explicando:

- Sua característica.
- Um exemplo de uso prático.
- Em que situação ela é mais adequada?

Atividade 7. Observe o código abaixo.

```
public class Usuario {
   public String nome;
   public String email;

   public void imprimirDados() {
      System.out.println("Usuário: " + nome + " - Email: " + email);
   }

   public void enviarEmail(String mensagem) {
      System.out.println("Enviando mensagem: " + mensagem + " para " + email);
   }
}
```

A) Você identifica algum problema de design ou violação de princípios de orientação a objetos neste código? Justifique com base nos conceitos aprendidos.

B) Quais melhorias poderiam ser feitas neste código, pensando em: Encapsulamento, Separação de responsabilidades e Reutilização de código.

Atividades Práticas (em Java)

Atividade 1. Implemente uma classe Estacionamento que simula o controle de entradas e saídas de veículos. Você deverá criar um enumerador chamado TipoOperacao, que define dois tipos de operação:

- ENTRADA: código "IN", descrição "Entrada no estacionamento"
- SAÍDA: código "OUT", descrição "Saída do estacionamento"

Cada operação registrada deve ser armazenada em um histórico contendo:

- Placa do veículo (String)
- Tipo de operação (usando o enum)
- Data e hora da operação

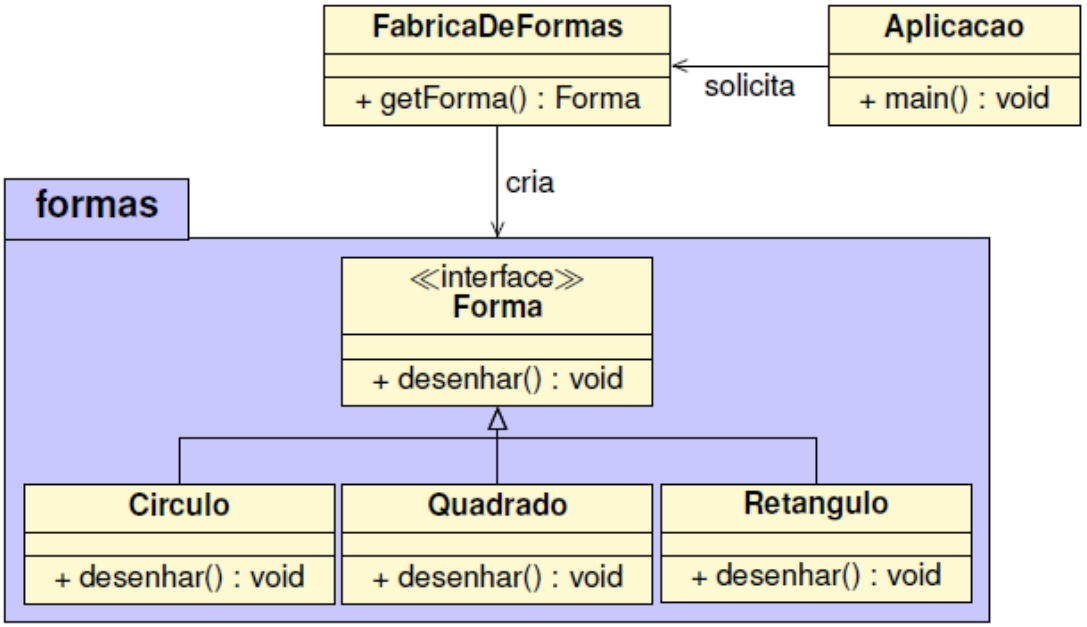
A classe Estacionamento deve ter:

- Um método registrarOperacao(String placa, TipoOperacao tipo)
- Um método exibirHistorico() que imprime todas as operações

PUC Minas

Pontifícia Universidade Católica de Minas Gerais
Bacharelado em Engenharia de Software - Campus Coração Eucarístico
Programação Modular - Semestre 2025/2
Prof. Daniel Kansaon
Atividade II

Atividade 2. Você foi contratado para criar um sistema simples que permita gerar diferentes formas geométricas: Círculo, Quadrado e Triângulo. Implemente a Figura abaixo utilizando o Factory Method.



Atividade 3. Crie uma aplicação que gerencie os produtos de um estoque. Cada produto tem nome, quantidade e preço. Armazene os produtos em um Map<String, Produto>, onde a chave é o nome. Permita adicionar, remover, buscar e listar produtos. Use Collections.sort() para ordenar os produtos por preço.

Atividade 4. Implemente uma aplicação em Java que simule a fila de atendimento de uma clínica médica, respeitando a ordem de chegada dos pacientes (sem prioridade). Utilize a classe LinkedList para armazenar os pacientes. Implemente duas funcionalidades:

- Adicionar um novo paciente na fila.
- Atender o próximo paciente.

Atividade 5. A partir de uma lista de objetos <Pessoa> com nome, idade e cidade, use Streams para:

- Filtrar pessoas maiores de 18 anos;
- Agrupar por cidade;
- Calcular a média de idades;
- Imprimir os nomes em ordem alfabética.

Atividade 6. Você está desenvolvendo um módulo de análise para uma empresa que armazena dados de seus funcionários. Cada objeto Funcionario possui os seguintes atributos: nome (String), departamento (String), salario (double).

A partir de uma lista de funcionários (List<Funcionario>), use a API Stream para realizar as seguintes operações:

- Filtrar os funcionários com salário superior a R$ 5.000,00.
- Agrupar os funcionários por departamento (ex: "Financeiro", "TI", "RH").
- Calcular a média salarial geral da empresa.
- Imprimir todos os nomes em ordem alfabética.
- Imprimir, para cada departamento, o total de funcionários.

Atividade 7. Observe o código abaixo.

```
public class FuncionarioService {
   private SMSNotificador notificador = new SMSNotificador();

   public void calcularSalario(String tipo, double valorBase) {
      if (tipo.equals("CLT")) {
         System.out.println("Salário CLT: " + (valorBase - 0.1 * valorBase));
      } else if (tipo.equals("PJ")) {
         System.out.println("Salário PJ: " + valorBase);
      } else {
         System.out.println("Tipo desconhecido");
      }
   }

   public void gerarRelatorio(Funcionario f) {
      System.out.println("Relatório: " + f.getNome() + " - " + f.getCargo());
   }

   public void enviarNotificacaoSMS(String numero, String mensagem) {
      notificador.enviarSMS(numero, mensagem);
   }
}
```

PUC Minas

Pontifícia Universidade Católica de Minas Gerais
Bacharelado em Engenharia de Software – Campus Coração Eucarístico
Programação Modular - Semestre 2025/2
Prof. Daniel Kansaon
Atividade II

A) Liste quais princípios SOLID estão sendo violados no código acima e por quê.
B) Refatore o código, aplicando boas práticas e os princípios SOLID.

Contexto adicional: A empresa deseja, futuramente, calcular salários de estagiários e freelancers com regras específicas e também permitir que o envio de notificações possa ser feito também por e-mail. Além de que os relatórios poderão ser exportados em diferentes formatos, como PDF e CSV.