

Atividade de Finalização – Unidade de Programação Orientada a Objetos

Curso Técnico em Desenvolvimento de Sistemas – Senac

Objetivo da Atividade

Consolidar os conceitos estudados na unidade de Programação Orientada a Objetos (POO) por meio do desenvolvimento de um projeto prático em equipe. A proposta é criar um aplicativo que utilize de forma adequada os princípios de classes, atributos, métodos, herança, polimorfismo e encapsulamento, aplicando-os em um sistema funcional com interface via terminal.

Formação das Equipes

- O projeto poderá ser realizado por equipes de 1 a 3 integrantes.
- A equipe deverá idealizar um tema e conceito do software e apresentá-lo ao professor antes de iniciar o desenvolvimento.

Requisitos Técnicos do Projeto

1. Classes e Estrutura

- O projeto deve conter pelo menos 3 classes no total.
- Pelo menos 1 classe pai (superclasse) e 1 classe filha (subclasse).
- Cada classe deve conter:
 - No mínimo 3 atributos.
 - No mínimo 2 métodos próprios (não incluindo getters e setters e sem contar métodos herdados).
 - Getters e Setters para todos os atributos, garantindo encapsulamento.

2. Herança

- A classe filha deve herdar atributos e métodos da classe pai.
- Pode incluir atributos e métodos adicionais exclusivos.

3. Polimorfismo

- Sempre que possível, utilize polimorfismo para que métodos herdados de uma classe pai possam ser redefinidos (sobrescritos) na classe filha, alterando seu comportamento para atender necessidades específicas.

- Exemplo: Um método `exibirInformacoes()` na classe pai pode ser implementado de forma genérica, mas na classe filha ser reescrito para incluir mais detalhes específicos.

4. Aplicativo

- O sistema deverá ser acessível via terminal com um menu de opções.
- O menu deve permitir executar operações que modifiquem e/ou utilizem os dados armazenados nas classes.
- O aplicativo deve fazer uso adequado dos métodos das classes para acesso e modificação de informações (evitar manipulação direta de atributos fora das classes).

5. Boas Práticas

- Código organizado e comentado.
- Nomes de classes, atributos e métodos devem seguir convenções de programação.
- Uso coerente de POO: encapsulamento, herança, polimorfismo.

Entregáveis

- Código-fonte completo do projeto.
- Descrição textual breve do tema e funcionalidades do aplicativo (máx. 1 página).
- O professor poderá solicitar demonstração do funcionamento.

Critérios de Avaliação

- Atendimentos aos requisitos técnicos (classes, herança, atributos, métodos).
- Uso correto de polimorfismo quando aplicável.
- Organização e clareza do código.
- Funcionamento do menu e operações.
- Uso correto de POO (encapsulamento, herança, abstração, polimorfismo).

Exemplo de Projeto

Tema: Sistema de Gerenciamento de Biblioteca

Descrição: O sistema permite gerenciar livros e usuários, permitindo cadastrar, listar e emprestar livros.

Classes sugeridas:

1. Classe Pai: `ItemBiblioteca`

- Atributos: `titulo`, `ano`, `disponivel`
- Métodos: `emprestar()`, `devolver()`

- Getters/Setters para todos os atributos.
- 2. **Classe Filha: Livro** (herda de `ItemBiblioteca`)
 - Atributos adicionais: `autor`, `genero`
 - Métodos adicionais: `detalhesItem()` (redefinição polimórfica de `detalhesItem()` da classe pai para mostrar informações específicas de livros), `resumo()`
- 3. **Classe Auxiliar: Usuario**
 - Atributos: `nome`, `idUsuario`, `livrosEmprestados`
 - Métodos: `cadastrarUsuario()`, `listarLivrosEmprestados()`

Menu via terminal:

- Cadastrar usuário
- Cadastrar livro
- Listar livros
- Emprestar livro
- Devolver livro
- Sair

Exemplo de uso do polimorfismo:

A classe `ItemBiblioteca` possui um método `detalhesItem()` que exibe apenas título e ano. Na classe `Livro`, esse método é sobrescrito para também mostrar autor e gênero.