

Trabalho Final de Programação Orientada a Objetos

André de Freitas Smaira

Objetivo

Realizar um projeto prático em C++ que explore os principais conceitos de Programação Orientada a Objetos (POO), como herança, polimorfismo, encapsulamento, composição, sobrecarga de operadores e tratamento de exceções. O trabalho será feito em grupos de 2 ou 3 integrantes, permitindo que vocês apliquem a teoria a um projeto criativo, funcional e desafiador.

Instruções Gerais

1. **Formação dos Grupos:** Cada grupo deve ser composto por 2 ou 3 alunos. Organizem-se para dividir as tarefas e documentem a contribuição de cada membro no relatório.
2. **Proposta do Projeto:** Cada grupo deve definir e enviar através do formulário abaixo uma proposta de trabalho, que será avaliada e o aluno será contactado para aprovação ou não da mesma.

<https://forms.gle/755SMiSQuYFh8PPEA>

A proposta deve incluir:

- Título do projeto;
- Descrição geral do tema e justificativa;
- Objetivo do projeto e funcionalidades principais

3. Entrega Final:

- **Código-Fonte:** Todo o código deve ser entregue em arquivos separados e adequadamente comentado.
- **Documentação e Relatório:** Entregar um relatório em formato PDF com as seguintes seções:
 - **Descrição do Projeto:** Explicar o que foi feito e qual problema o projeto busca resolver.
 - **Planejamento e Estrutura:** Detalhar as decisões de design, justificando o uso de herança, composição e os relacionamentos entre as classes.
 - **Diagramas de Classes e Casos de Uso:** Diagrama UML mostrando as relações entre as classes e diagramas de casos de uso, se aplicável.
 - **Implementação e Funcionalidades:** Explicar como o projeto foi implementado, descrevendo as funcionalidades e as decisões de design específicas.
 - **Guia de Uso:** Instruções sobre como compilar, executar e usar o programa, incluindo exemplos.
 - **Desafios e Soluções:** Liste as dificuldades encontradas e como foram resolvidas.

Requisitos do Projeto

Para garantir que o projeto explore os conceitos de POO aprendidos em aula, o trabalho deve seguir as seguintes restrições e requisitos técnicos:

1. Mínimo de Classes e Herança:

- O projeto deve conter ao menos 5 classes, com no mínimo uma delas servindo de classe base para outras derivadas.
- As classes derivadas devem especializar a classe base, adicionando ou estendendo suas funcionalidades.

2. Uso de Herança e Polimorfismo:

- Implementar ao menos uma classe abstrata com um ou mais métodos virtuais puros.
- Incluir funções virtuais em classes base para demonstrar polimorfismo e permitir flexibilidade na manipulação dos objetos.

3. Sobrecarga de Operadores:

- Incluir a sobrecarga de ao menos dois operadores distintos de maneira que seja relevante ao contexto da classe.
- Justificar o uso de cada operador sobrecarregado, descrevendo como ele facilita o uso da classe e melhora a legibilidade do código.

4. Encapsulamento e Acessibilidade:

- Todos os atributos devem ser privados ou protegidos, acessíveis apenas por métodos getters e setters.
- Evitar o uso de atributos públicos. Qualquer caso de exceção deve ser justificado no relatório.

5. Construtores e Destrutores:

- Cada classe deve ter ao menos um construtor com parâmetros e, se necessário, outros construtores, como o construtor de cópia e de movimento.
- Implementar destrutores apropriados quando necessário.

6. Composição:

- Pelo menos uma classe deve usar composição, ou seja, incluir outra classe como um de seus atributos.
- Justificar a escolha de composição em vez de herança para a modelagem do projeto.

7. Armazenamento de Dados:

- Incluir funcionalidade de leitura e/ou gravação em arquivos ou em banco de dados externo para salvar e carregar dados, como um sistema de persistência de dados para o projeto.

8. Documentação e Diagrama de Classes:

- Documentar cada classe e método, incluindo comentários explicativos.
- Apresentar um diagrama UML mostrando as relações e hierarquias entre as classes, além de diagramas de casos de uso relevantes para demonstrar como o sistema funciona.

9. Criatividade e Originalidade:

- O projeto deve ser único e inovador, sendo uma criação original do grupo.
- Reproduções ou cópias não serão aceitas.

Critérios de Avaliação

O projeto será avaliado com base nos seguintes critérios:

1. **Planejamento e Organização** (10%): Concordância entre a proposta inicial e o trabalho realizado.
2. **Cumprimento dos Requisitos de POO** (30%): Uso correto de herança, polimorfismo, encapsulamento, sobrecarga de operadores, e tratamento de exceções. Implementação de construtores, destrutores e composição.
3. **Documentação e Relatório** (30%): Qualidade e clareza da documentação. Completude do relatório, incluindo diagramas, explicações das escolhas, manual de utilização, exemplos de uso, dificuldades enfrentadas e referências utilizadas.
4. **Originalidade e Criatividade** (10%): Nível de inovação e exclusividade do projeto.
5. **Funcionalidade e Apresentação** (20%): O programa deve funcionar corretamente e cumprir os requisitos propostos. Apresentação e usabilidade do software (interface textual ou gráfica, se aplicável).

Dicas Finais

- **Divisão de Trabalho:** Dividam as responsabilidades entre os membros do grupo para garantir que todos contribuam e entendam o funcionamento do projeto.
- **Teste e Validação:** Testem o código regularmente e garantam que todos os casos de uso funcionem corretamente.
- **Documentação:** Comentem o código conforme forem avançando, pois isso ajudará na compreensão e no desenvolvimento do relatório.

Entrega

Submeter o código comprimido em um arquivo .zip e o relatório em formato .pdf no Google Classroom.

Boa sorte e bom trabalho!