

Trabalho Final

Programação para Design – 2018.2

Prof. Anderson Lemos

1. Objetivo

Desenvolver um sistema robusto utilizando o paradigma da Programação Orientada a Objetos utilizando a linguagem de programação **TypeScript**. Esse sistema deverá ser escolhido pelo aluno e ter uma interface de interação com o usuário (não necessariamente gráfica, pode ser em console). Trabalhos com um fim social, que ajudem a comunidade de alguma forma, serão muito bem vistos.

2. Requisitos

O código do sistema desenvolvido pelo aluno deverá:

- Ter um diagrama de classes atualizado, modelando o sistema exatamente como ele foi/será implementado em código.
- Ter no mínimo dez classes.
- Ter interfaces/classes abstratas.
- Apresentar herança entre as classes.
- Apresentar algum tipo de polimorfismo.
- Apresentar encapsulamento correto nos métodos e atributos das classes.
- Ter pelo menos três tipos de exceções.
- Apresentar o uso inteligente de coleções (listas, conjuntos ou mapas), no sentido de escolha da coleção mais eficiente para ser utilizada.
- Disponibilizar ao usuário várias operações possíveis (não pode ser um sistema simples, que faça poucas coisas). Geralmente, deve ser possível fazer CRUD (inserção, busca, atualização e remoção) das entidades do sistema.
- Ter uma interface de fácil uso e intuitiva (mesmo que por linha de comando).

3. Considerações Finais

- As equipes serão formadas por no máximo **2 (DOIS)** alunos.
- Todos os requisitos do trabalho são **OBRIGATÓRIOS**, e devem ser preenchidos com coisas que façam sentido. Cada requisito não cumprido acarretará na diminuição da nota.
- As notas serão diferenciadas por aluno no momento da apresentação do trabalho em sala.
- A nota do trabalho fará parte da nota final da disciplina, valendo um terço da nota total.
- O trabalho deverá ser implementado em TypeScript.

4. Dicas

- Sejam criativos! Façam um sistema que vocês gostariam de usar. Funcionalidades a mais serão muito bem vistas, por exemplo:
 - Salvar os dados em arquivos para não serem perdidos quando o programa for fechado e serem carregados quando o programa for aberto.
 - Interface gráfica.
 - Se seu sistema utiliza alguma mídia (música, vídeo), executar a mídia.
- A interpretação do trabalho é uma forma de avaliação. Se você acha que algo não ficou bem explicado, tente entender como seria a maneira certa de fazê-lo e resolva o problema. Esse é um problema que vocês vão ter que lidar diariamente no mercado de trabalho.
- Deixem a parte mais difícil por último, concentrem-se em implementar as classes de acordo com a orientação a objetos. As funcionalidade podem ser implementadas depois.
- Procurem dividir o trabalho entre os integrantes.
- Sintam-se livres para implementar da forma que for mais conveniente. No entanto, respeitem a OO e convenções de código. Tornem o programa funcional.
- Façam o trabalho por partes: leiam com cuidado este documento e vão implementando à medida que forem entendendo.
- Tirem dúvidas com o professor, entre vocês e pela Internet.
- Utilizem os conceitos vistos em sala: herança, polimorfismo, abstração, encapsulamento, tratamento de erros, coleções, classes abstratas, etc.

Bom trabalho!