

Trabalho POO

Este trabalho é uma atividade extra-classe, complementando os tópicos estudados em sala.

A escolha de tema/cenário, o desenvolvimento do código fonte, a pesquisa sobre o uso das bibliotecas necessárias (dependendo do tema escolhido) devem acontecer fora do horário de aula.

Datas importantes:

Defesa do código fonte: 12 (seg) e 14 (qua) /12

(de acordo com o cronograma disponível no moodle)

Deadline para postar código fonte, resumo, diagrama de classes:

Alunos que apresentam dia 12: postar até 13:00h do dia 12 (ou seja, antes do início da aula)

Alunos que apresentam dia 14: postar até 15:00h do dia 14 (ou seja, antes do início da aula)

O horário e data de apresentação de cada equipe ficará disponível no moodle.

Fique atento ao dia e horário!

Apresentações de segunda chamada terão no máximo 50% da nota.

Cenários são únicos e a prioridade na escolha do cenário, horário/data de apresentação é por ordem de envio de e-mail a professora.

luciana.rech@ufsc.br

Descrição do Trabalho:

Desenvolver um sistema (em Python) utilizando obrigatoriamente conceitos de Programação Orientada a Objetos estudados em aula.

Cada equipe (equipes de até 2 componentes) irá desenvolver um sistema (cenário exclusivo) que contemple os seguintes requisitos mínimos:

- uso de Estruturas de Seleção/Repetição;
- uso de Lista /e-ou/ ListaComposta /e-ou/ Matriz /e-ou/ Dicionário;
- exemplo/uso de Instanciação;
- exemplo/uso de Herança;
- exemplo/uso de Polimorfismo;
- (Agregação ou Composição, opcional);
- manipulação de uma lista de objetos (ações: inserir, alterar, consultar e excluir).

Há dois temas possíveis para realizar este trabalho. Abaixo há um resumo dos dois temas.

Você deve escolher **apenas um dos temas** para o trabalho, sendo que o trabalho pode ser desenvolvido em grupos de até 2 membros.

Tema 1 : Jogo

1) Descrição do Trabalho - Jogo

Neste trabalho você irá conhecer (pesquisar) um pouco sobre a biblioteca para desenvolvimento de jogos do Python, a **pygame**. Assim, o objetivo do trabalho é que você desenvolva um jogo utilizando os conceitos aprendidos na disciplina. O jogo não pode ser muito simples (como um jogo da velha), mas também não deve ser muito complexo (como um FIFA). Neste trabalho, você também deve pesquisar e utilizar conceitos de orientação a objetos em Python.

Links úteis

- [Introdução e primeiro jogo usando pygame](#)
- [Documentos e tutoriais sobre pygame](#)
- [Outros tutoriais sobre pygame](#)
- [Exemplo de jogo \(Space Shooters - Leonardo Gideão\)](#)
- [Editor de Mapas](#)
- Para orientação a objetos em Python, visite os tópicos da disciplina relacionados à orientação a objetos

Tema 2: Aplicação

2) Descrição do Trabalho - Aplicação

Neste trabalho, você deverá pesquisar e estudar orientação a objetos na linguagem Python e desenvolver uma aplicação qualquer, que possa ser útil no dia-a-dia de alguém. A aplicação não pode ser tão simples (como uma mera calculadora), mas também não precisa ser tão complexa (como um editor de imagens estilo Photoshop). Alguns temas válidos para esta aplicação são sistemas para video-locadoras, supermercados, padaria, posto de combustível, agenda para médicos/dentistas, hospital, etc.

TODOS os sistemas (**independente do tema escolhido**) deverão executar as seguintes ações: inserir, alterar, consultar e excluir de uma lista de objetos.

Portanto, cada equipe deverá escolher o tema e o cenário de seu sistema, entrar em contato com a professora para autorização do início da implementação (cenários não podem ser repetidos). É de responsabilidade do aluno verificar se todos os requisitos foram cumpridos.

A prioridade na aprovação do tema/cenário escolhido será a data de recebimento do e-mail solicitando aprovação do cenário).

Para a defesa (apresentação) do trabalho serão necessários:

- 1- Explicação do Diagrama de Classes (que deverá estar impresso para ser explicado no início da apresentação)
- 2- Apresentação do Resumo (também impresso no início da apresentação)
- 3- Apresentação da implementação do cenário (explicação de trechos do código, testes serão realizados no momento da apresentação).

Cada equipe deve chegar 30 minutos antes do horário agendado para ligar o computador, verificar se tudo está ok, e deixar o código pronto para realização dos testes (não ultrapassando assim o tempo de defesa)

Tempo previsto de defesa 30 minutos por equipe.

Um Modelo de Diagrama de Classes (inspirado em UML) ficará disponível no Moodle. Exemplos de softwares gratuitos para edição do Diagrama de Classes: Star UML, Astah, VPOOnline...