



UNIVERSIDADE ESTADUAL DE SANTA CRUZ - UESC

PRÓ-REITORIA DE GRADUAÇÃO – PROGRAD

DEPARTAMENTO DE CIÊNCIAS EXATAS - DCET

COLEGIADO DE CIÊNCIA DA COMPUTAÇÃO - COLCIC

Primeiro trabalho

Curso	Ciência da Computação
Disciplina	Linguagem de Programação III
Professor(es)	Otacílio José Pereira
Assunto	- Conceitos básicos de OO - Herança - Tópicos complementares

1. Apresentação inicial

O objetivo do trabalho é desenvolver um pequeno programa voltado para um cenário, em uma outra linguagem Orientada a Objetos para efeito de comparação e explorando tópicos diversificados para aprimorar a criação de programas.

Parte 1 : Implementar o cenário com conceitos básicos

Atividade 1) Compreenda o seu cenário e crie um programa com uma classe principal que apresente o menu de opções do usuário (em interface de comando / modo texto).

Atividade 2) Crie as classes do cenário aplicando todos os conceitos básicos de OO (classes, objetos, atributos, métodos, construtores, getters e setters, impressão e outros).

Robô: Classes Robô e Sala
Agricultura 4.0: Classes Sensor e Atuador
Comércio Eletrônico: Classes Compra e Produto
Computação gráfica: Classes Figura Geométrica e Categoria

Atividade 3) Implemente os métodos próprios do cenário, que dão a funcionalidade chave do cenário.

Robô: Movimentações
Agricultura 4.0: Leituras, ligar/desligar
Comércio eletrônico: Adicionar itens e calcular valor de compra
Computação gráfica: Calcular propriedades geométricas (área e volume)

Atividade 4) Experimentar o programa com as opções do menu para criar os objetos e realizar processamentos sobre os objetos.

Parte 2 : Usar herança e polimorfismo

Atividade 1) Criar uma classe derivada da classe base conforme cenário.

Robô: Robô 3D (analogia com Drone)
Agricultura 4.0: Atuador Georeferenciado (possui latitude e longitude)
Comércio Eletrônico: Produto de Terceiro (vendido por terceiros)
Computação gráfica: Figura-Objeto 3D

Atividade 2) Revise o menu e o programa para manipular também estes objetos. AO armazenar os elementos em lista, usar um vetor ou ArrayList de superclasses e adicionar e manipular as subclasses (polimorfismo).

Parte 3 : Comparar linguagem com o aprendizado em Java

Atividade 1) Crie uma planilha ou um quadro com os principais tópicos da Orientação a Objetos e nas colunas escreva trechos de código de como seria a escrita em Java e como foi na linguagem de seu trabalho.

Atividade 2) Faça comentários sobre as semelhanças e diferenças das duas linguagens. Explore pelo menos 6 tópicos relevantes de OO dentre estes (Classes, Objetos, Atributos, Métodos, Construtores, Encapsulamento, Herança, Polimorfismo e Classes Abstratas/Interfaces).

Exemplo de quadro (se fosse nossa exemplo de Conta):

Tópico	Java	Linguagem do trabalho
Classe, atributos e métodos	<pre>class Conta { int Numero; double Saldo; public void Sacar(double pValor) { this.Saldo -= pValor; } ... }</pre>	<pre>class Conta: numero = 1 saldo = 100.0 def sacar(self, pValor): self.saldo -= pValor ...</pre>
Comentário	<ul style="list-style-type: none">- No python não há a declaração explícita de tipo- Há o uso da palavra "self" na implementação do método- Semelhanças: como em Java, permite orientação a objetos com o uso de classes, atributos e métodos.	

Parte 4 : (Opcional, pontuação extra) Avançando com outros conhecimentos

Obs.: Basta realizar uma das atividades e o aluno pode ter até um ponto extra. Mais de uma atividade, permanece apenas 1,0 ponto extra.

Atividade 1) Caso o aluno queira, ele pode tentar usar alguma biblioteca de Interface Gráfica (no Javascript, o próprio HTML com CSS pode servir) e gerar uma versão gráfica do programa.

Atividade 2) Caso o aluno queira ele pode incrementar as funções do programa para salvar e ler de arquivo, e daí adicionar funções no Menu que salvam os objetos em arquivo e evita de reinserí-los em outras edições.

Atividade 3) Caso o aluno queira, ele pode tentar prever uma versão do trabalho com o uso de frameworks comerciais como (React, Node.js, Django, Biblioteca de Computação Gráfica e outras) e daí incorporar seus conceitos na codificação do trabalho.

3. Pontos, entregas e critérios

Valor:

6,0 pontos

(Crédito de prática)

Data de entrega final:

30/10

(Se apresentar 26/10 – 0,5 ponto extra)

Item	Descrição	Valor
Parte 1	Implementar cenário com conceitos básicos e em outra linguagem	2,0 pts
Parte 2	Implementar herança e polimorfismo	2,0 pts
Parte 3	Comparar as linguagens	2,0 pts
Parte 4	Avançar com conhecimentos em outros tópicos	1,0 pto (extra)

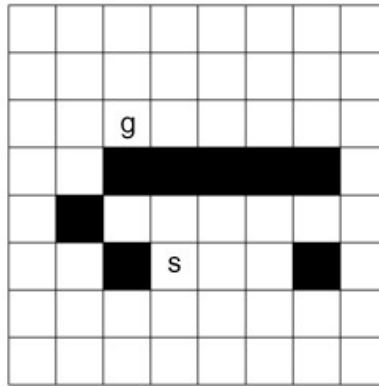
Pontos de atenção:

- Os grupos devem ter até 3 integrantes. / - Quem não apresentar o trabalho terá nota anulada
- Trabalhos copiados de colegas ou Internet terão a nota anulada
- Outros critérios como comprometimento, antecipação de dúvidas e discussões, qualidade de apresentação e outros que demonstrem boa atitude e profissionalismo podem ser avaliados também, com bonificação ou penalização.

Anexo: Descrição dos cenários

Cenário Robô (Vide exemplo de sala)

- Imagine que um robô precisa se mover em uma indústria.
- Na sala, os espaços em branco são áreas livres e os espaços pretos são obstáculos.
- Considere que a sala é uma matriz $N \times N$ com a posição (0,0) representada no canto superior esquerdo e que a posição do robô é uma tupla (i,j), linha e coluna da matriz.



As seguintes funções devem ser previstas:

- Posicionar robô
 - Estabelece a posição do robô e do alvo de destino
- Posicionar obstáculos
 - Posiciona os diversos obstáculos na indústria.
 - Deve ser uma collection de posições em que se encontram os obstáculos
- Mover robô (direita, esquerda, para cima e para baixo)
 - Permite movimentar o robô nas diversas posições. Caso haja obstáculo o robô não é movimentado. Ao chegar ao alvo sinalizar mensagem de sucesso.
- Mostrar situação atual da sala
 - Mostra como está a sala com o posicionamento do robô, do alvo e dos obstáculos.

Para herança e polimorfismo

- Considere que o robô em vez de se movimentar apenas no chão (plano 2D) ele pode também agora se movimentar na vertical em um espaço 3D, portanto existe o Robo3D que herda do Robô original, possui mais uma coordenada e mais dois movimentos.

Cenário Agricultura 4.0 (Vide exemplo de sala)

- Imagine que em um cenário existe um sensor de umidade do solo e uma bomba. Com base nos dados de umidade a bomba deve ser ligada.

- Veja slides das aulas iniciais para maiores detalhes e qualquer dúvida solicite esclarecimentos ao professor.

As seguintes funções devem ser previstas:

- a) Inicializar sensor de umidade
 - Estabelece o ID, descrição e valor de leitura do sensor
- b) Inicializar bomba
 - Estabelece os dados iniciais da bomba, ID, descrição e status (ligada/desligada)
- c) Ler sensor
 - Altera os dados do sensor e conforme valor aciona ou não a bomba
- d) Controle automático
 - Repetidamente solicita dados do sensor (como se fizesse a leitura automática) e daí atualiza os dados gerais do cenário.
- e) Visualizar cenário
 - Mostra todos os dados do cenário.

Para herança e polimorfismo

- Considere que o sensor pode ser georreferenciado para se saber a posição dele na lavoura (com latitude e longitude) que são inseridos manualmente.

- Além disso considere que não há mais apenas um sensor mas um conjunto de 3 sensores georreferenciados ou não. A bomba liga ou desliga com base na média dos valores dos sensores que são armazenados em um vetor.

Cenário Comércio Eletrônico

- Imagine que neste cenário uma compra em um site da Internet vai ser emulado. Uma determinada loja pode vender até 5 tipos de produtos que possui código, descrição e valor.

- Para realizar uma compra o usuário solicita a criação de um carrinho ou adiciona um item na compra e o carrinho é criado automaticamente. A partir daí o usuário escolhe novos itens ou edita itens do carrinho e por fim finaliza a compra.

As seguintes funções devem ser previstas:

- a) Visualizar produtos
 - Mostra a lista dos 5 produtos que podem ser vendidos na loja.
 - A lista de produtos pode ser fixa
- b) Criar carrinho
 - Criar um objeto da classe carrinho que contém uma Lista de produtos e um campo de total.
- c) Adicionar produto
 - Conforme o ID do produto selecionado, adiciona o item ao carrinho
- d) Visualizar carrinho
 - Mostra os itens do carrinho
- e) Retirar item
 - Retira item do carrinho
- f) Concluir compra
 - Mostra um relatório (cupom fiscal) da compra com os itens e total

Para herança e polimorfismo

- Imagine que a compra (carrinho) acima é para entrega em casa mas que o cliente deseja pegar na loja. Neste caso, crie uma classe CarrinhoLoja ou CompraLoja que contém também uma data em que o cliente vai pegar os itens e que calcula o total com um desconto de 10% (que seria usado para frete).

Cenário Computação Gráfica

- Imagine que os equipamentos em uma fábrica podem ser representados por figuras geométricas 3D cubo, paralelepípedo e cilindros. Cada figura tem portanto suas propriedades que permite calcular a área e o volume e contem também uma descrição do equipamento (se preferir pode fazer um outro objeto equipamento relacionado).

As seguintes funções devem ser previstas:

- a) Visualizar figuras geométricas / equipamentos
 - Mostra a lista de até 5 equipamentos registrados na indústria
- b) Registrar equipamento
 - Cria um novo equipamento. Ele solicita o tipo (cilindro, cubo ou paralelepípedo) e depois as suas propriedades (lado ou raio).
- c) Calcular área e perímetro de um equipamento
 - Seleciona um equipamento e obtem os dados
- d) Calcular a área total dos equipamentos
 - Calcula o total das áreas dos equipamentos
- e) Retirar equipamento
 - Retira o equipamento da industria

Para herança e polimorfismo

- Na versão simples, o aluno pode trabalhar com apenas um tipo de figura por exemplo, com cubos.

- Na versão com herança, o aluno precisa criar uma classe mais alta (Objeto 3D) com as características comuns e depois as figuras específicas conforme o enunciado. Estas diferentes figuras podem ser armazenadas no mesmo vetor (polimorfismo).