

UTP – Curso de Bacharelado em Ciência da Computação
Disciplina: Programação Orientada a Objetos – Prof. Diógenes Furlan

Lista Prática – 1º Bimestre – 2021

Sobre avaliação:

- A atividade vale 30 pontos da nota bimestral (em 100 pontos).
 - O trabalho deve ser feito individualmente.
-

Faça 5 projetos em linguagem C++, usando o Devcpp, o CodeBlocks, o CodeLite ou outro IDE de sua preferência:

Não se esqueça de informar no programa (comentário) a IDE utilizada na construção do protótipo.

Programas

(1) INTRODUÇÃO

- Crie uma classe chamada **CAIXA**:

Caixa
altura: double largura: double profundidade: double peso: double empilhamento: int
area(): double volume(): double numCaixasSala(alt,larg,prof): int

- Tornar atributos private, criando getters e setters para todos eles.
- Crie um campo Código (sem Setter)
- Treine a questão do encapsulamento fazendo outras checagens legais nos métodos setters.
 - altura, largura e profundidade não podem ser negativos
 - aceite peso em grama ou em quilos e converta para quilos
- Crie um método mostra(), que mostra os dados da caixa.
- Adicionar um destrutor, que imprime “caixa excluída”.

- Crie três construtores diferentes:
 - um com 3 parâmetros (altura, largura e profundidade).
 - um sem parâmetros: que inicia com tudo com ZERO.
 - um com 5 parâmetros.

(2) COMPOSIÇÃO

- Crie uma classe chamada **PONTO**.
 - com 2 atributos: X e Y
- Crie dois construtores para esta classe:
 - um, sem parâmetros, que inicializa o ponto com 0,0
 - outro com 2 parâmetros
- Crie dois métodos chamados distancia:
 - um que recebe dois valores reais: X e Y
 - um que recebe outro Ponto
- Crie uma classe **RETA** que tem como atributos dois objetos da classe Ponto.
- Crie 2 construtores para esta classe:
 - um que inicializa os pontos a partir de 2 pontos passados por parâmetro
 - um que inicializa os pontos a partir de 4 coordenadas reais passadas por parâmetro
- Quando seu programa ficar pronto acrescente função membros para esta reta, tais como:
 - coeficiente angular (inclinação) $\rightarrow a$
 - coeficiente linear (corta eixo y) $\rightarrow b$
- Faça uma rotina que mostra a reta na forma
 - $y=ax + b$

- Crie uma classe **TRIÂNGULO** composta por três objetos de tipo Ponto.
- Crie rotinas para:
 - calcular os 3 lados do triangulo
 - classificar o triangulo (equilátero, isósceles, escaleno)
 - calcular a altura do triangulo

(3) HERANÇA

Defina uma classe chamada **CONTA**. Esta classe deve armazenar o nome do cliente, o código da conta e o valor monetário.

Inclua:

- a) dois construtores: o primeiro sem argumentos, e o outro que recebe 2 parâmetros.
- b) como fazer o código ser gerado automaticamente a cada nova conta criada? Comece com o código 0001 e avance de 1 a cada nova conta criada.
- c) métodos para realizar depósito, saque e saldo

Vamos modelar algumas classes derivadas de Conta.

- a) classe **contaCorrente**
 - fazer validações no saque
 - impedir saques maiores que o armazenado
 - impedir saques superiores a 2000
- b) classe **contaEspecial**
 - deve ter “limite” de crédito
 - deve ter “cobrança de taxa” (método)
- c) classe **contaPoupança**
 - deve ter uma “taxa de reajuste” mensal
 - impedir saques maiores que o armazenado

4) //OPERATORS

Crie uma classe chamada **COMPLEXO**, para representar números complexos, e escreva um programa para testá-la.

- a) Escolha uma representação para os números complexos, usando a forma retangular ou a forma polar.
- b) Forneça três construtores que permitam que objetos dessa classe sejam inicializados ao serem alocados na memória:
 - um construtor sem parâmetros que inicializa o objeto como zero
 - um construtor com um parâmetro representando a parte real; a parte imaginária será zero
 - um construtor com dois parâmetros representando as partes real e imaginária
- c) Defina operações para obter a parte real, a parte imaginária, o módulo (valor absoluto) e o ângulo de um número complexo.
- d) Forneça a operação para determinar o inverso aditivo de um número complexo.
- e) Forneça as operações aritméticas básicas com números complexos: adição, subtração, multiplicação e divisão.
- f) Forneça as operações relacionais que permitem comparar dois números complexos.
- g) Defina a operação toString para converter um número complexo em string. Utilize o formato (a;b), onde a é a parte real e b é a parte imaginária.
- h) Escreva um aplicativo de teste que demonstra as capacidades da classe Complexo.

(5) GERAL

Um **GRAFO** é uma estrutura de dados muito comum em computação, e os algoritmos sobre grafos são fundamentais para a área.

Um grafo $G = (V; A)$ consiste em:

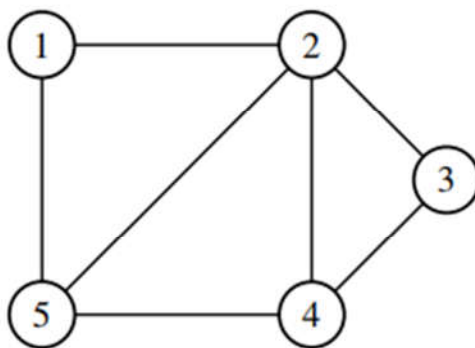
- um conjunto finito de pontos V . Os elementos de V são chamados de vértices de G .
- um conjunto finito A de pares não ordenados de V , que são chamados de arestas de G . Uma aresta a em A é um par não ordenado $(v; w)$ de vértices v, w em V , que são chamados de extremidades de a .

Uma aresta a em A é chamada de incidente com um vértice v em V , se v for uma extremidade de a .

Um vértice v em V diz-se vizinho de outro vértice w em V se existir uma aresta a em A incidente com v e w .

Um grafo pode ser representado por listas de adjacência ou por uma matriz de adjacência, como é ilustrado na figura abaixo:

(a) Grafo



(b) Listas de adjacências de um grafo

vértice	lista de adjacência
1	2, 5
2	1, 5

3	2, 4
4	2, 5, 3
5	4, 1, 2

(c) Matriz de adjacência do grafo

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

Escreva uma classe para representar grafos. Escolha entre a representação por listas de adjacência ou por matriz de adjacência. A classe deve oferecer uma operação para determinar se dois vértices são vizinhos, e outra operação para determinar a lista de todos os vértices que são vizinhos de um dado vértice. Considere que cada vértice é representado por um número inteiro.

Escreva um aplicativo para testar a classe. Crie pelo menos 3 objetos diferentes.

Entrega

- Data: até 10/maio/2021.
- Colocar nome completo dos integrantes dentro dos arquivos (usando comentário).
- Fazer o UPLOAD no link específico, na página do professor no site do Gerds
 - mandar pelo email: nestredidi@yahoo.com.br

Tabela de Avaliação

Item	Peso
Projeto 1	20
Projeto 2	20
Projeto 3	20
Projeto 4	20
Projeto 5	20
Total	100

BOA DIVERSÃO!!!