



Trabalho Prático 1 - Polígonos

Tipos Abstratos de Dados / Alocação Dinâmica

Valor: 0,5 pontos (5% da nota total)

Data de entrega: 05/04/2010

Documentação em Latex: 0,1 pontos extra

Interface gráfica: até 0,3 pontos extra

Impressão frente-verso: 0,05 pontos extra

Objetivos

Consiste em rever conceitos básicos de programação e alocação de memória bem como explorar os conceitos de Tipos Abstratos de Dados (TADs). Ainda serão necessários conhecimentos em geometria analítica para realização do trabalho prático.

Descrição

Você deverá implementar um tipo abstrato de dados `TPonto` para representar pontos no espaço \mathcal{R}^2 . Esse tipo abstrato deverá armazenar as coordenadas x e y do ponto.

As operações que devem ser realizadas pelo seu TAD são:

1. Ler um ponto (leia as coordenadas);
2. Criar um ponto (forneça as coordenadas);

Você deverá implementar também um TAD `TPoligono` para armazenar um conjunto de pontos que descrevam o polígono (a quantidade de pontos varia de polígono para polígono). Observe que um polígono é formado por pelo menos 3 pontos.

As operações que devem ser realizadas por este TAD são:

1. Criar um polígono com três pontos;
2. Inserir um ponto no polígono;
3. Remover o i -ésimo ponto do polígono;
4. Calcular o perímetro do polígono;
5. Calcular a área do polígono;
6. Dados dois polígonos A e B , determinar um polígono C que seja a interseção de A e B ;
7. Dados dois polígonos A e B , determinar um polígono C que seja a união de A e B .

Você deverá implementar ainda um TAD `TFigura` para armazenar um conjunto de polígonos.

Esse TAD deve implementar as seguintes operações:

1. Inicializar uma figura vazia;
2. Inserir um polígono na figura (observe que a figura não permite interseção de polígonos);
3. Remover o i -ésimo polígono da Figura;
4. Calcular o perímetro total da figura;



5. Calcular a área total da figura.

A determinação das declarações das operações (ou assinaturas das funções) faz parte da avaliação do trabalho e é de inteira responsabilidade sua.

Implemente seus TADs em arquivos separados do programa principal (TPonto.c e TPonto.h, TPoligono.c e TPoligono.h e TFigura.c e TFigura.h). Se necessário, você pode criar outras funções auxiliares em seu TAD. Suas funções devem executar testes de consistência.

Uma vez criado os TADs, você deverá criar um programa *main* para manipular esses TADs de forma que seja possível testar a consistência das implementações.

O que deve ser entregue

- Código fonte do programa em C ou C++ (bem indentado e comentado).
- Documentação do trabalho. Entre outras coisas, a documentação deve conter:
 1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
 2. Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. **Muito importante**: os códigos utilizados nas implementações devem ser inseridos na documentação.
 3. Análise de Complexidade: estudo da complexidade de tempo e espaço das funções implementadas e do programa como um todo (notação O).
 4. Listagem de testes executados: os testes executados devem ser apresentados e analisados e discutidos, quando convier.
 5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
 6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso. Uma referência bibliográfica deve ser citada no texto quando da sua utilização
 7. Em Latex: Caso o trabalho seja elaborado/escrito em latex, ganha-se 0,1 pontos.
 8. Formato: mandatoriamente em PDF (<http://www.pdf995.com/>).

Obs: Veja modelo de como fazer o trabalho em latex:

<http://www.decom.ufop.br/menotti/ed1101/tps/modelo.zip>

Como deve ser feita a entrega:

A entrega DEVE ser feita via Moodle (www.decom.ufop.br/moodle) na forma de um **único** arquivo zipado, contendo o código, os arquivos e a documentação. Também deve ser entregue a documentação impressa na próxima aula teórica após a data de entrega do trabalho.



Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Biológicas – ICEB
Departamento de Computação – DECOM
Disciplina: Estruturas de Dados I – BCC202
Professor: David Menotti (menottid@gmail.com)
Professora: Emilianara Mara Lopes Simões (simoes.eml@gmail.com)



Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- Clareza, identificação e comentários no programa também serão avaliados;
- O trabalho é individual (grupo de UM aluno);
- Trabalhos copiados (e FONTE) terão nota zero. Devido a recorrentes problemas com cópias de trabalhos (plágios), os autores de trabalhos copiados também terão a maior nota nos testes teóricos levada a zero, como forma de punição e coação ao plágio acadêmico;
- Trabalhos entregues em atraso serão aceitos, todavia a nota atribuída ao trabalho será zero;
- Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.