

Primeira Lista de Exercícios

Estruturas de Dados 1

Prof. Paulo Henrique Ribeiro Gabriel

Tipos Abstratos de Dados

1. Sobre Tipo Abstrato de Dados (TAD), responda:
 - (a) O que é um TAD?
 - (b) Qual a característica fundamental na sua utilização de um TAD?
 - (c) Quais as vantagens de se programar com TADs?
2. Desenvolva um TAD Ponto que represente um ponto em \mathbb{R}^2 com as seguintes operações:
 - (a) $P \leftarrow criar(x, y)$: Criação de um ponto P com coordenadas x e y (aqui também é feita a alocação de memória – note que a variável ponto é “retornada” pela função);
 - (b) $destruir(P)$: Liberação da memória alocada;
 - (c) $acessar(P, x, y)$: Retorna as coordenadas de um ponto (note que os valores de x e y são instanciados **dentro da função** – o valor “retornado” estará nessas variáveis);
 - (d) $atribuir(P, x, y)$: Atribui novos valores às coordenadas de um ponto;
 - (e) $distancia(P_1, P_2)$: Calcula a distância entre dois pontos.

Note que as coordenadas x e y são valores reais. Para revisar conceitos sobre pontos e distâncias, acesse este [link](#). Faça, inicialmente, uma *especificação textual* do TAD e de suas operações. Em seguida, crie um programa em C que implemente esse TAD e faça diversos testes (não se esqueça de implementar a função `main`, ou seja, o usuário do TAD).

3. Utilizando o TAD Ponto criado no exercício anterior, defina o TAD Círculo, com as seguintes operações:
 - (a) $C \leftarrow criar(P, r)$: cria um círculo com centro no ponto P e raio r (aqui também é feita a alocação de memória – note que a variável círculo é “retornada” pela função);

- (b) *destruir*(P): libera a memória alocada por um círculo;
- (c) *area*(C): calcula a área do círculo;
- (d) *interior*(P, C): verifica se um dado ponto P está dentro do círculo C .

Note que algumas das operações solicitadas dependem de operações do TAD Ponto. Dessa maneira, é possível exercitar a criação de diferentes módulos em um programa. Faça, inicialmente, uma *especificação textual* do TAD e de suas operações. Em seguida, crie um programa em C que implemente esse TAD e faça diversos testes (não se esqueça de implementar a função `main`, ou seja, o usuário do TAD).

4. Um número complexo é escrito na forma $x + iy$, sendo que x é a parte real do número, y é a parte imaginária, e $i = \sqrt{-1}$. Tanto a parte real quanto a imaginária são representadas por valores reais. Crie um TAD que represente os números complexos com as seguintes operações:

- (a) Criação de um número complexo;
- (b) Destruição um número complexo;
- (c) Soma de dois números complexos;
- (d) Subtração de dois números complexos;
- (e) Multiplicação de dois números complexos;
- (f) Divisão de dois números complexos.

Além disso, não necessárias operações específicas para exibir (“retornar”) a parte real e a parte imaginária do número. Faça, inicialmente, uma *especificação textual* do TAD e de suas operações. Em seguida, crie um programa em C que implemente esse TAD e faça diversos testes. (Um bom material para relembrar os conceitos de números complexos está disponível neste [link](#).)