



Curso: ADS – ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS

Turna: INF006

Disciplina: Estrutura de Dados (Prática)

Data: ____/____/____

Conceito:

Os tipos abstratos de dados (TADs) são estruturas de dados capazes de representar os tipos de dados que não foram previstos no núcleo das linguagens de programação e que, normalmente, são necessários para aplicações específicas. Essas estruturas são divididas em duas partes: os dados e as operações. A especificação de um TAD corresponde à escolha de uma boa maneira de armazenar os dados e à definição de um conjunto adequado de operações para atuar sobre eles.

A característica essencial de um TAD é a separação entre conceito e implementação, ou seja, existe uma distinção entre a definição do tipo e a sua representação, e a implementação das operações. Um TAD é, portanto, uma forma de definir um novo tipo de dado juntamente com as operações que manipulam esse novo tipo de dado. As aplicações que utilizam esse TAD são denominadas clientes do tipo de dado.

Como exemplo, um tipo de dados interessante e do qual necessitamos frequentemente é o tipo Data. Seguidamente precisamos fazer cálculos envolvendo datas e nem sempre isto é fácil. Por exemplo, *quantos dias existem entre 24 de abril e 2 de fevereiro?* Apesar de ser possível efetuar este cálculo, ele não é trivial, já que os meses têm número de dias diferentes. Assim, parece uma boa ideia implementar um TAD para Data.

Formalmente, um TAD pode ser definido como um par (v,o) , onde v representa um conjunto de valores e o representa o conjunto de operações aplicáveis sobre esses valores. Por exemplo, um TAD para representar uma data pode ser dado pelo par (v,o) , onde:

- v – é uma tripla formada por dia-mês-ano; e
- o – são as operações aplicáveis sobre o tipo data, como verificar se uma data é válida, calcular o dia da semana de uma determinada data, calcular a data do Carnaval de um determinado ano, entre outras.

Uma vez que o TAD foi caracterizado, o próximo passo é escolher sua estrutura de representação, ou seja, a estrutura de dados propriamente dita que irá suportar as operações definidas. A representação será uma coleção de campos primitivos ou mesmo uma estrutura complexa formada por vários campos primitivos. Considerando o TAD Data, podemos criar uma estrutura com três campos do tipo inteiro para representar, respectivamente, o dia, o mês e o ano, como mostrado a seguir:

Data = registro

Dia: inteiro

Mês: inteiro

Ano: inteiro

fim registro

ATIVIDADE PRÁTICA:

Questão 1 Especifique um TAD que seja capaz de armazenar uma data composta por dia, mês e ano e implemente o seguinte conjunto de operações para manipular esse tipo Data:

- uma função que recebe como parâmetro o dia, o mês, o ano e uma estrutura do tipo Data, onde o resultado deve ser armazenado, e retorna verdadeiro se a data estiver válida; caso contrário, retorna falso. A validação da data pode ser feita da seguinte forma:
 - meses com 30 dias: 04, 06, 09 e 11;
 - meses com 31 dias: 01, 03, 05, 07, 08, 10, 12;
 - cálculo de ano bissexto: cada ano divisível por 4 é um ano bissexto(E); cada ano divisível por 100 não é bissexto (OU); todo ano divisível por 400 sempre é um ano bissexto;
- uma função que recebe como parâmetro uma data do tipo string (no formato DD/MM/AAAA) e uma estrutura do tipo Data, onde o resultado deve ser armazenado, e retorna verdadeiro se a data estiver válida; caso contrário, retorna falso;
- uma função que recebe como parâmetro uma estrutura do tipo Data e um número inteiro, e retorna a soma do número de dias da data recebida;
- uma função que recebe como parâmetro uma estrutura do tipo Data e um número inteiro, e retorna a subtração do número de dias da data recebida;
- uma função que recebe como parâmetro uma data e escreve essa data por extenso. Por exemplo, 25/03/2007 deve ser escrito como *25 de março de 2007*. A função deve retornar verdadeiro se a operação for realizada com sucesso e falso, caso contrário;
- uma função que receba como parâmetro o ano e retorne a data da Páscoa. O domingo de Páscoa é o primeiro domingo depois da primeira lua cheia do outono. Utilize o seguinte algoritmo, criado em 1800 pelo matemático Carl Gauss:
 - suponha que y seja o ano (como 1800 ou 2001);
 - divida y por 19 e chame o resto de a. Ignore o quociente;
 - divida y por 100 para obter o quociente b e o resto c;
 - divida b por 4 para obter o quociente d e o resto e;
 - divida $8 * b + 13$ por 25 para obter o quociente g. Ignore o resto;
 - divida $19 * a + b - d - g + 15$ por 30 para obter o quociente h. Ignore o resto;
 - divida c por 4 para obter o quociente j e o resto k;
 - divida $a + 11 * h$ por 319 para obter o quociente m. Ignore o resto;
 - divida $2 * e + 2 * j - k - h + m + 32$ por 7 para obter o resto r. Ignore o quociente;
 - divida $h - m + r + 90$ por 25 para obter o quociente n. Ignore o resto;
 - divida $h - m + r + n + 19$ por 32 para obter o resto p. Ignore o quociente.

Questão 2 Considere uma aplicação para armazenar os seguintes dados de uma pessoa em uma agenda de endereços: nome, endereço e telefone. Especifique um TAD para armazenar os dados das pessoas e as operações necessárias para inserir, consultar e excluir os dados das pessoas.

Questão 3 Considere uma aplicação para armazenar os seguintes dados de carros para uma garagem: placa, marca/modelo e cor. Especifique um TAD para armazenar os dados dos carros e as operações necessárias para inserir, consultar e excluir os dados das pessoas.

Questão 4 Considere uma empresa que precisa armazenar os seguintes dados de um cliente:

- nome completo;
- ano de nascimento;
- renda mensal do cliente.

Especifique um TAD para armazenar os dados de um cliente e as operações necessárias para inserir, consultar e excluir os dados dos clientes. Especifique também operações para exibir o número de clientes com renda mensal acima da média, e exibir o número de clientes que nasceram entre 1980 e 2000.