Aula 11

1) Registro (STRUCT)

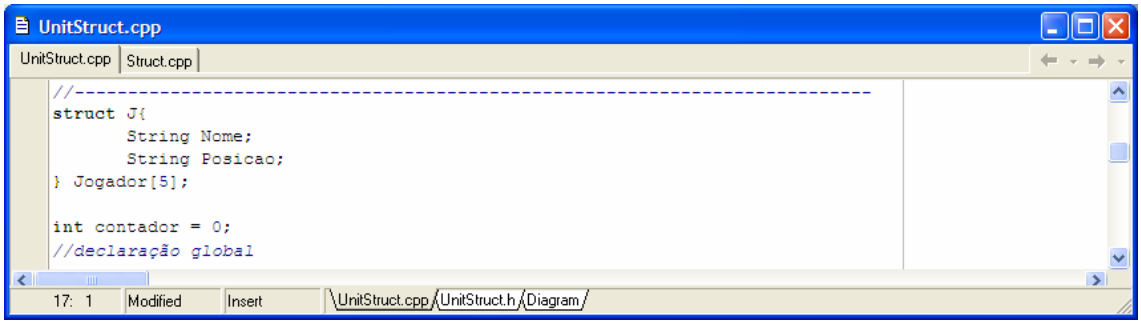
São conjuntos de dados logicamente relacionados, mas de tipos diferentes (numérico, string, lógico). O conceito de registro visa facilitar o agrupamento que não são do mesmo tipo, mas que guardam estreita relação lógica. O conjunto de variáveis que compõem a estrutura do registro é também denominado de registro lógico. A referência ao conteúdo de um componente do registro é indicada pela notação:

Identificador do registro. Identificador do componente

Exemplo:

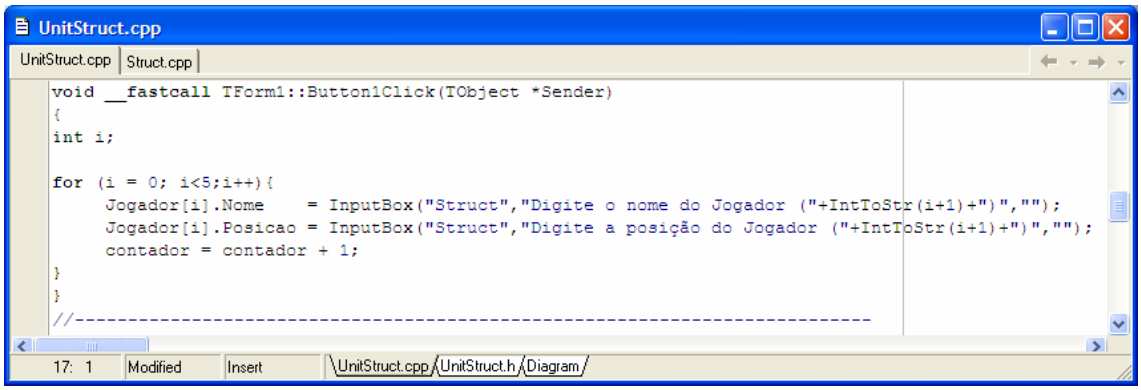
Layout da aplicação.

Passo 1: Declaração global da estrutura de dados e variáveis...



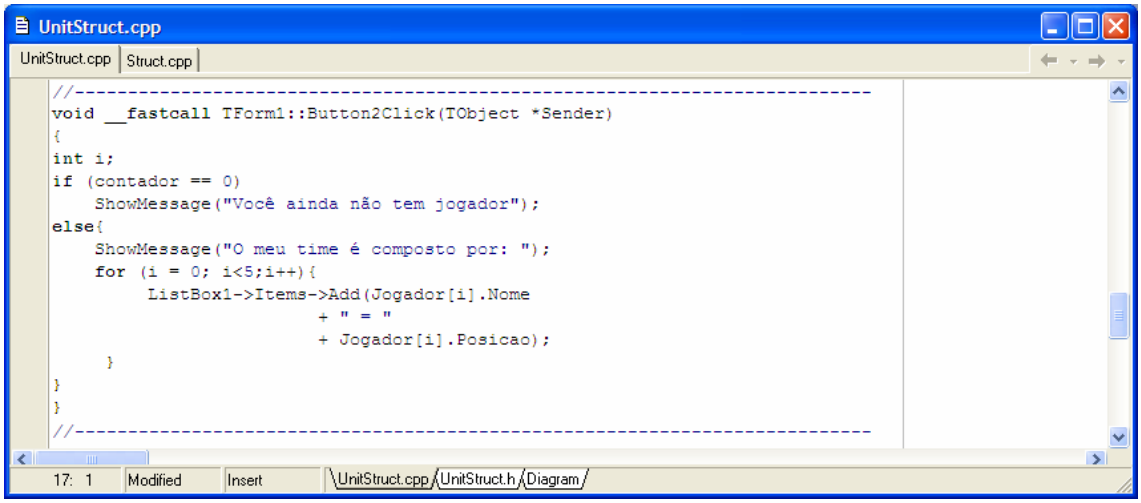
Implementando Struct.

Passo 2: No botão inserir...



Implementação da inserção de valores na estrutura de dados.

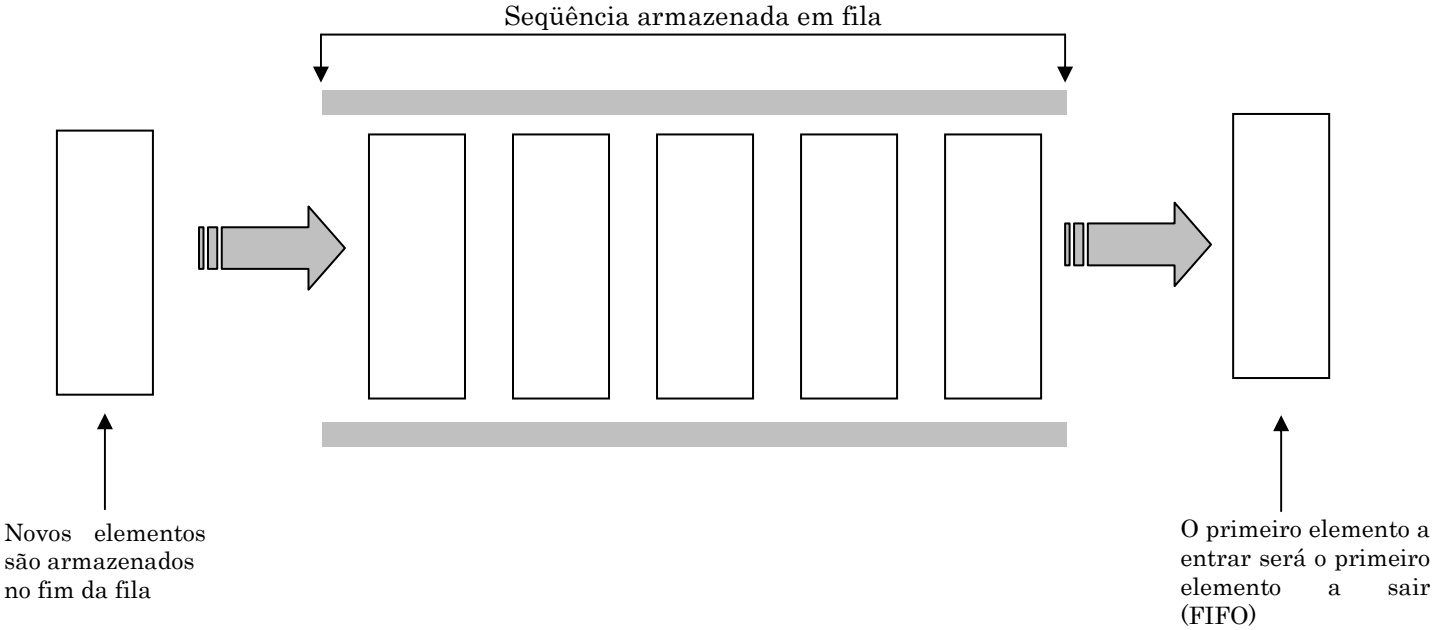
Passo 3: No botão listar...



Implementação da exibição dos elementos armazenados na estrutura de dados.

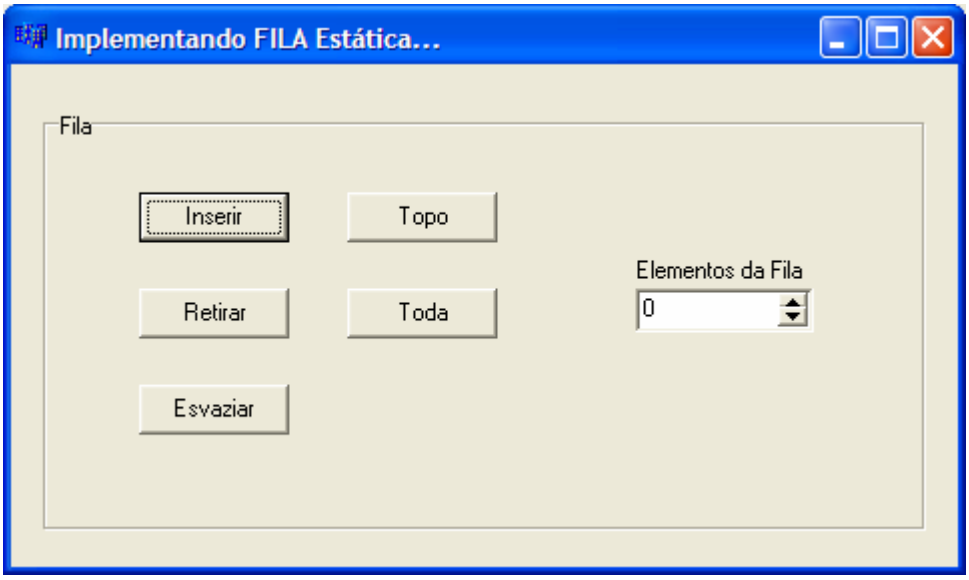
2) Fila

É possível afirmar que todos nós já ficamos em uma fila. Fila para comprar ingressos para shows, pegar dinheiro no banco e, às vezes, até para comprar o pãozinho da manhã. O conceito de fila em programação é o mesmo dessas filas em que esperamos para ser atendidos em ordem: o primeiro elemento a entrar na fila será o primeiro elemento a sair. Esse conceito é conhecido como 'First In, First Out' ou FIFO, expressão conhecida em português como PEPS ou 'Primeiro que entra, Primeiro que sai'. Então no conceito de fila, os elementos são atendidos, ou utilizados, seqüencialmente na ordem em que são armazenados. As filas (*queues*) são conjuntos de elementos (ou listas) cujas operações de inserção são feitas por uma extremidade e de remoção, por outra extremidade.Como exemplo pode-se implementar uma fila de impressão, em que os arquivos a ser impressos são organizados em uma lista e serão impressos na ordem de chegada, à medida que a impressora estiver disponível.

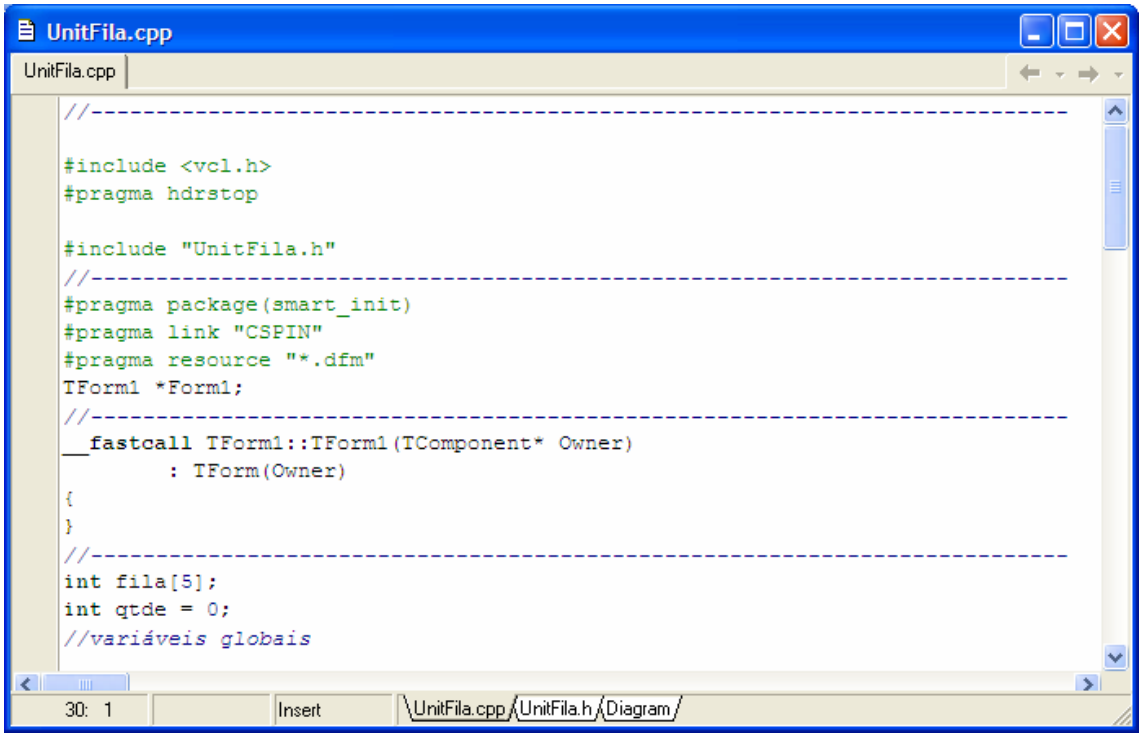


Conceito de fila

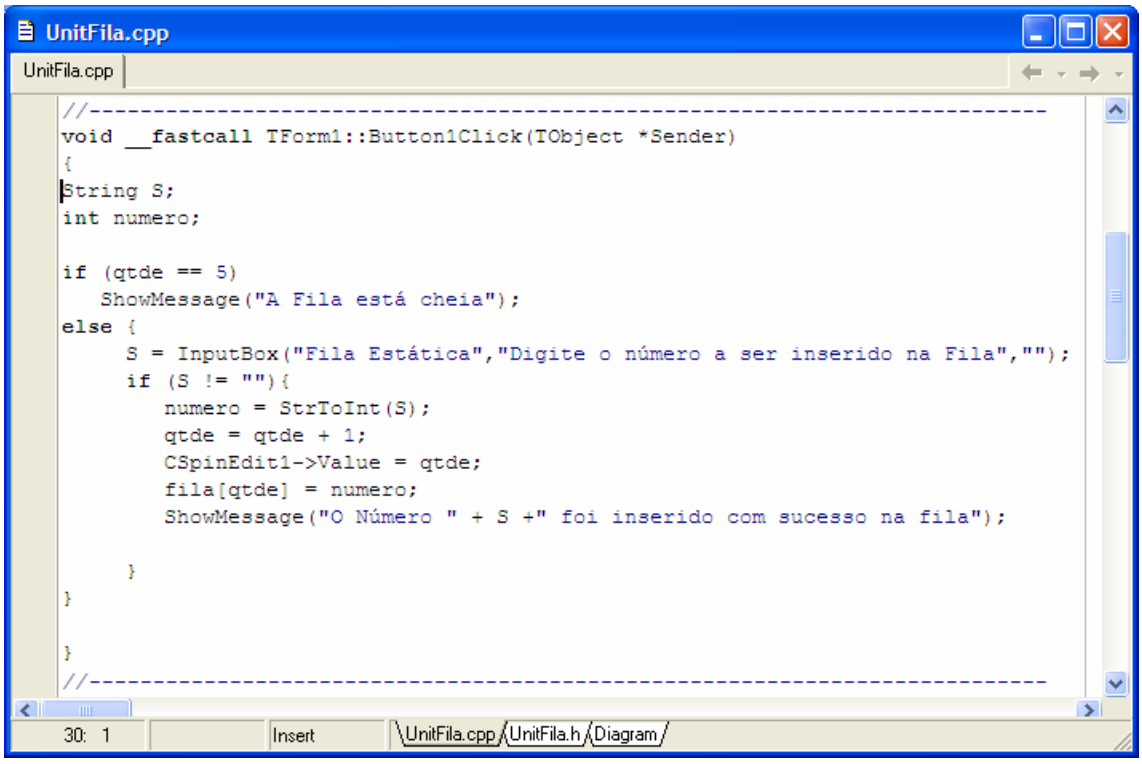
Exemplo:



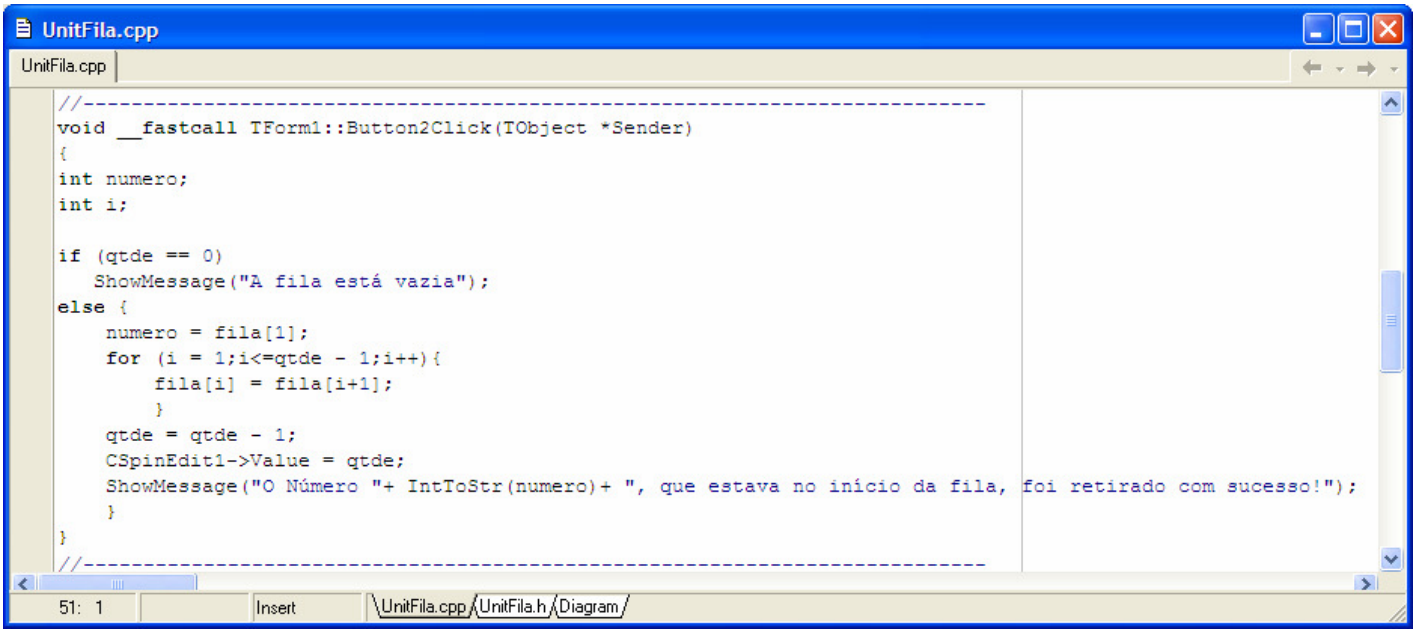
Layout da aplicação



Declaração das variáveis globais.



Implementação do botão inserir.



Implementação do botão retirar.

UnitFila.cpp

```
//---------------------------------------------------------------------------
void __fastcall TForm1::Button5Click(TObject *Sender)
{
if (qtde == 0)
   ShowMessage(" A fila já está vazia");
else {
    qtde = 0;
    CSpinEdit1->Value = qtde;
    ShowMessage("A fila foi esvaziada com sucesso!");
    }
}
//---------------------------------------------------------------------------
```

93: 1 Insert UnitFila.cpp UnitFila.h Diagram

Implementação do botão esvaziar.

UnitFila.cpp

```
//---------------------------------------------------------------------------
void __fastcall TForm1::Button3Click(TObject *Sender)
{
if (qtde == 0)
   ShowMessage("A Fila está vazia");
else
   ShowMessage("O Número do início da fila é o número: "+IntToStr(fila[1]));
}
//---------------------------------------------------------------------------
```

69: 1 Insert UnitFila.cpp UnitFila.h Diagram

Implementação do botão topo.

UnitFila.cpp

```
//---------------------------------------------------------------------------
void __fastcall TForm1::Button4Click(TObject *Sender)
{
String texto;
int i;

if (qtde == 0)
   ShowMessage("A fila está vazia");
else {
    texto = "";
    for (i = 1;i<=qtde;i++){
        texto = texto + "   "+ IntToStr(fila[i]);
    }
ShowMessage("A fila toda: "+ texto);
}
}
//---------------------------------------------------------------------------
```

77: 1 Insert UnitFila.cpp UnitFila.h Diagram

Implementação do botão toda.