UNIVERSIDADE FEDERAL DO CEARÁ

CK0109 2019.2 T02 - ESTRUTURAS DE DADOS

AULA 13 - 2019-09-09

ITERADORES

1.0 PONTEIRO THIS: DENTRO DE UMA FUNÇÃO-MEMBRO, A EXPRESSÃO THIS DENOTA UM PONTEIRO PARA A ESTRUTURA A PARTIR DA QUAL A FUNÇÃO FOI CHAMADA.

EXEMPLO:

```
#INCLUDE <IOSTREAM>
USING STD::COUT;
STRUCT TipoQQ
  {
    INT CAMPO_QQ;
    //TipoQQ
    TipoQQ* MEU_ENDERECO () { RETURN THIS;}
  };
INT MAIN ()
  {
    TipoQQ a,b;
    COUT << "ENDEREÇO DE a: "
         << a.MEU_ENDERECO() << '\n';
    if( &a != a.MEU_ENDERECO() ||
        a.MEU_ENDERECO()== b.MEU_ENDERECO() )
      { COUT << "ABSURDO!\n"; }
  } //MAIN
```

2. REFERÊNCIAS: EM C++, UMA REFERÊNCIA É OUTRO NOME PARA UM OBJETO NA MEMÓRIA. ELAS COMPARTILHAM PARTE DA UTILIDADE DOS PONTEIROS, POR EXEMPLO A POSSIBILIDADE DE SE ACESSAR UMA VARIÁVEL NUM CONTEXTO EM QUE ELA NÃO PODE SER REFERIDA PELO NOME, E COM VANTAGENS, COMO A NÃO NECESSIDADE DO USO DO OPERADOR * PARA O ACESSO. POR OUTRO LADO, REFERÊNCIAS SÃO MAIS RESTRITAS QUE PONTEIROS, NO SENTIDO DE QUE, ENQUANTO PONTEIROS PODEM SER REDIRECIONADOS, UMA REFERÊNCIA SEMPRE APONTA PARA O MESMO ENDEREÇO, A SABER, O ENDEREÇO DA INICIALIZAÇÃO.

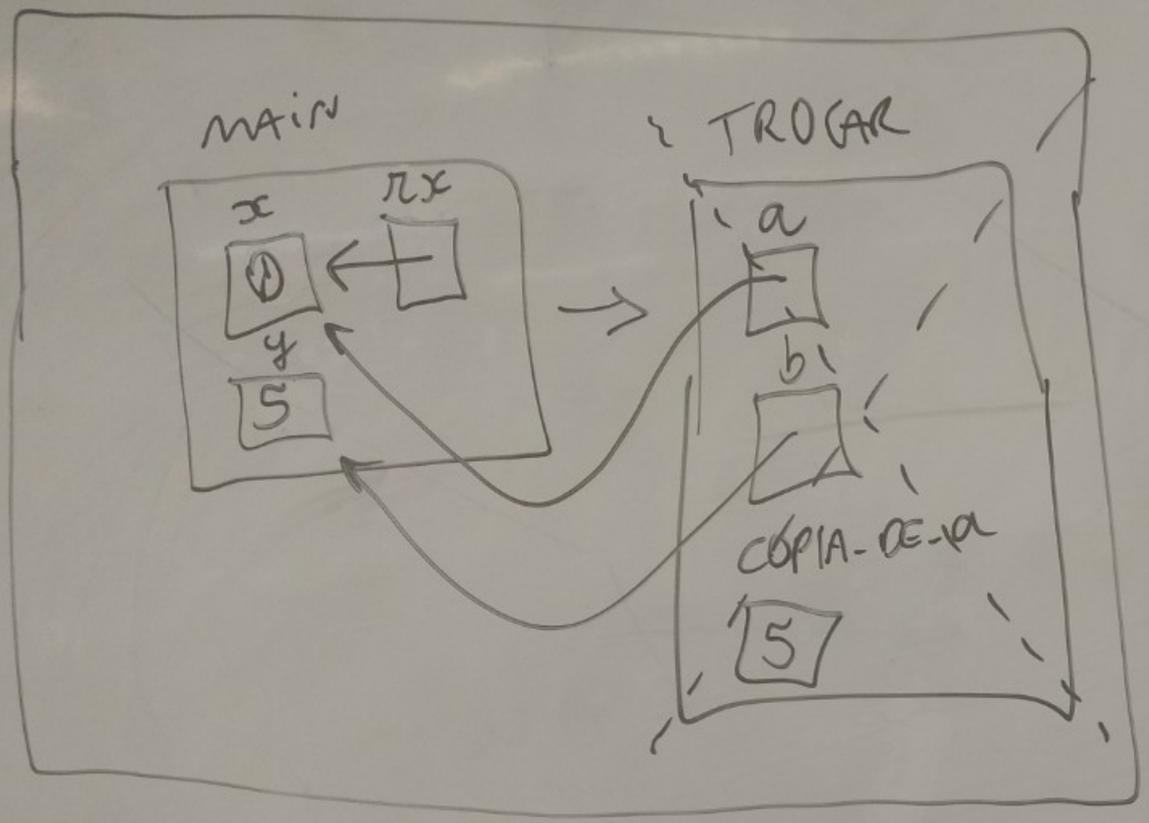
EXEMPLO:

```
VOID TROCAR (INT &a, INT &b)
  {
    INT COPIA_DE_a = a;
    a = b;
    b = COPIA_DE_a;
  }

INT MAIN ()
  {
    INT x = 10; INT &rx = x;
    rx = 5; //x == 5
    INT y = 0;
    TROCAR(x, y);
    // x == 0 && y == 5
  }
```

Simulação

x, rx

Memória



3. Sobrecarga de Operadores: em C++,

podemos dar "novos significados" aos operadores.

Exemplo:

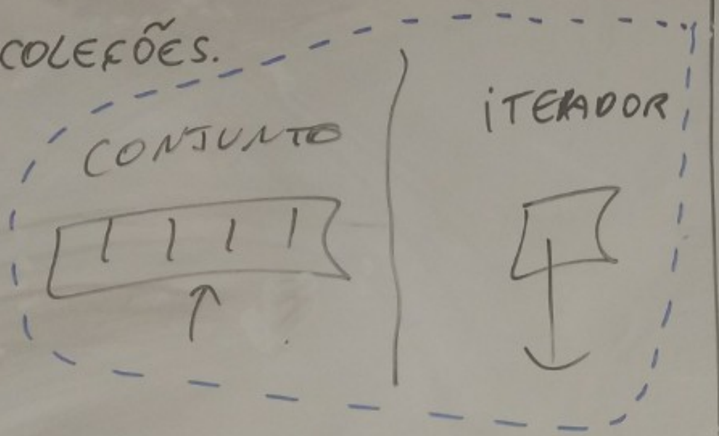
```
struct Vetor2D
  {
    double x, y;

    Vetor2D operator+ (const Vetor2D &v)
    {
      Vetor2D novo;
      novo.x = x + v.x;
      novo.y = y + v.y;
      return novo;
    }
};
```

a+b def.= a.operator+(b)

```
int main()
  { Vetor2D a,b, c = a+b; }
```

↳ a.operator+(b)

4. ITERADORES: DÃO UMA MANEIRA NÃO-INTRUSIVA DE SE PERCORRER COLEÇÕES.



EXEMPLO:

```
...
INT MAIN( )
  {
   CONJUNTO<DOUBLE> C;
   ...POVOAR C...
   AUTO FIM = C.FIM( );
   FOR(AUTO I = C.INICIO( ); I != FIM; ++I)
      COUT << ' ' << *I;
   C.TERMINAR( );
  }
```

IMPLEMENTAÇÃO:

$--(++i)$

$i++$

```
...
//CONJUNTO
STRUCT ITERADOR
  {
    T *P;

    BOOL OPERATOR!= (CONST ITERADOR &J)
      {
       RETURN ( P != J.P);
      }

    T& OPERATOR*( ) { RETURN *P;}

    ITERADOR& OPERATOR++( )
      {
        ++P;
        RETURN *THIS;
      }
  }; //ITERADOR
```