Linguagens de Programação

Paradigmas de Programação
Introdução a C++

Aula Teorica 2

Topicos

I. Paradigmas de Programação
II. Introdução a linguagem C++

Paradigma- Conceito

Palavra de origem grega, que significa exemplo a ser seguido ou modelo padrão. Conjunto de acções ou elementos na mesma estrutura.

Paradigma de Programação e um conjunto de características agregadas em uma linguagem de programação, relacionada com a organização ou identidade da própria linguagem, na resolução de problemas.

Tipos de Paradigmas

Diversos paradigmas existem para os diversos tipos de problemas, similar as inúmeras linguagens de programação.

A escolha de um paradigma depende do tipo de problema a ser solucionado, por isso deve-se escolher o paradigma que permite uma solução optima.

Tipos de Paradigmas

No entanto, certos problemas não permitem uma solução concisa com a utilização de apenas uma paradigma, isto é, necessitam de agregar diferentes paradigmas.

Tipos de Paradigmas

1. Paradigma Imperativo- caracterizada por procedimentos que servem de mecanismos de accao,denomindo de procedural pois inclui sub-rotinas ou procedimentos.

Ex: Fortran, Pascal, Python, C, ALGOL, Assembler, Ada;

2. Paradigma Estruturado- dividido em três estruturas: sequencia, decisão, e iteração.

Ex: C++, Pascal, Cobol;

Tipos de Paradigmas

3. Paradigma Funcional- enfatiza a aplicação de funções, com foco em funções matematicas.
Ex: LISP, Scheme, Haskell;

4. Paradigma Logico- utilizado em aplicações de inteligência artificial.
Ex: QLISP, Planner, Prolog;

5. Paradigma Orientado a Objectos- programa composto por objetos com propriedades e operações que podem ser executados.
Ex: Ruby, C++, Object Pascal, Java, C#, Oberon;

Tipos de Paradigmas

6. Paradigma Orientado a Eventos- baseado na ocorrência de eventos externos, apropriado no uso de interfaces graficas.
Ex: Delphi, Visual Basic, C#, Python, Java

7. Paradigma Multiparadigma- suporta mais de um paradigma de programação.

Historia do C++

A linguagem C++ foi desenvolvida inicialmente por Bjarne Stroustrup na AT&T, de 1979 a 1983, à partir da linguagem C, tendo como idéia principal a de agregar o conceito de classes. Razão porque inicialmente chamava-se de “C com classes”. Bjarne procurou tanto quanto possível manter compatibilidade com C, de modo que programas em C pudessem ser compilados por um compilador C++ com um mínimo de alterações. Entretanto, encarar C++ como um superconjunto de C é um erro, e C++ deve ser vista como uma “outra linguagem”.

Historia do C++

Nem todo o programa escrito em C é compilado em C++ sem erros, e pode nem mesmo gerar o mesmo resultado, já que a sintaxe e a semântica de algumas construções diferem. Além disso, C++ oferece um conjunto de mecanismos básicos que não estao presentes em C.

Finalmente, os mecanismos de C++ devem inspirar a programação segundo o paradigma de orientação a objetos e, portanto, não se deve programar em C++ como se faz em C.

Historia do C++

Pode-se dizer que C++ foi a única linguagem entre tantas outras que obteve sucesso como uma sucessora à linguagem C, inclusive servindo de inspiração para outras linguagens como Java e C♯.

Conceitos

- Possui sintaxe própria e rígida
  - Conjunto de palavras reservadas
  - Bibliotecas padrão de funções
  - Ambiente de desenvolvimento
- Permite declaração de variáveis
- Permite definições de blocos de comandos
- Permite organização de funções e módulos
- Utilizada para a implementação de algoritmos

Tipo de Dados em C++

A declaração do tipo de dados em C++ é necessária para informar ao computador como alocar e manipular os dados em memória, sejam dados do tipo inteiro, flutuante, ou uma sequência de caracteres.

Tipo de Dados em C++

Existem os tipos de dados básicos que são:

- **char:** Caracter: O valor armazenado é um caractere. Caracteres geralmente são armazenados em códigos (usualmente o código ASCII).
- **int:** Número inteiro.
- **float:** Número em ponto flutuante de precisão simples. São conhecidos normalmente como números reais.
- **double:** Número em ponto flutuante de precisão dupla.
- E outros...

Declaração de Variáveis

```
int a; // Declara um inteiro a

const int c; // Declara inteiro constante C

int f(); // Declara que existe uma função f()

struct S; // Declara que existe uma estrutura S

class C; // Declara que existe uma classe C
```

Declaração com Inicialização

Consiste em declarar uma variavel, ao mesmo tempo, atribuir um valor.

Exemplo:

```
int x = 7;    // O mesmo que int x(7);
float y = 5.2;     // Inicializa y com o valor 5.2
int r(30);        // Inicializa r com o valor 30
```

Palavras-Reservadas do C++

Uma linguagem de programação faz uso extensivo de determinadas palavras, denominadas palavras-chave. Essas palavras foram definidas para a linguagem C++ e são utilizadas pelo programador com algum objectivo.

Palavras-chave, também conhecidas como palavras reservadas da linguagem, são palavras que não podem ser usadas como identificadores, ou seja, não podem ser usadas para representar variáveis, classes ou nomes de métodos.

Palavras-Reservadas do C++

Tipos	bool, char, wchar_t, short, int, long, float, double, long double, void
Modicadores de tipos	auto, const, extern, mutable, register, signed, static, typedef, unsigned, volatile
Controle	break, case, continue, default, do, else, for, goto, if, return, switch, while
Logicos	and, and_eq, bitand, bitor, compl, false, not, not_eq, or, or_eq, true, xor, xor_eq
Memoria	delete, new
Diversos	Diversos asm, class, enum, explicit, export, friend, inline, namespace, operator, sizeof, struct, template, this, typeid, typename, union, using, virtual
Conversoes	const_cast, dynamic_cast, reinterpret_cast, static_ cast

Regras e Boas Praticas

Todas as regras devem ser aplicadas de forma coerente ao longo de todo o programa.

Nomes de variaveis ou classes devem ser significativos, levando em consideracao a existencia de palavras chaves que fazem parte do processador e não podem ser usadas;

Todos os nomes de variáveis começam por letras de (a-z, A-Z) ;

Regras e Boas Praticas

- Todas instruções terminam com ponto e virgula (;);
- Comentários podem ser agrupados como se segue abaixo:

// comentário para uma linha

/* abrir comentario para um bloco e fechar*/

- Maiúsculas e minúsculas fazem diferença!!!
- *Em geral, nomes de classes são iniciados por maiúsculas.*

Regras e Boas Praticas

A linguagem C++ difere maiúsculas de minúsculas, ou seja, AA e diferente de aa.

Embora ' ' possa ser utilizado no inicio de um nome, ele deve ser evitado.

Caracteres validos { a-z A-Z 0-9 + - * / = , . : ; ? n “

Caracteres

Os caracteres '\n'(nova linha), '\a' (beep) e '\t' (tabulação) são conhecidos como caracteres de escape. Um carácter de escape e composto de uma barra invertida e uma letra que indica a acção a ser executada.

“endl”

'\f’

'\n’

Estrutura de Programa em C++

Um programa em C++ é constituído por:

- Várias funções, das quais uma obrigatoriamente tem que se chamar main.

  A função main(), como qualquer outra é definida :

  - por um cabeçalho constituído por:
    - » tipo de dados que a função devolve.
    - » o seu nome.
    - » parâmetros formais que recebe.
  - por um corpo (definido entre “{ }”) com :
    - » declarações
    - » definições
    - » instruções
    - » comentários

Estrutura de programa em C++

Exemplo de estrutura geral

```
#include <stdio.h>- BIBLIOTECA
/* impressão de uma mensagem simples
*/COMENTÁRIO
int main()
{    ---------------------- BLOCO PRINCIPAL
    Cout<<“ Mensagem inicial em c++! \n”;
    return 0;
}
```

Operadores Relacionais

> Maior

< Menor

>= Maior igual

<= Menor igual

= = Igual

!= Diferente

Operadores Lógicos

Os operadores lógicos estabelecem uma linguagem protocolar, clara e precisa. O retorno de um operador lógico é denominado “Valor Lógico”, que somente pode ser de dois tipos: verdadeiro ou falso.

- OR: Disjunção não exclusiva
- END: Conjunção
- NOT: Negação

Operadores de Atribuição

= comando de atribuição

- Menos

+ Mais

* Multiplicação

/ Divisão

% Resto da Divisão

-- Decremento

++ Incremento

Entrada e Saida de Dados

O objeto **cout** representa o ***stream* de saída** no C++. Este *stream* é uma espécie de seqüência (fluxo) de dados a serem impressos na tela.

cout « “primeiro programa em c++";

Entrada e Saida de Dados

O objeto **cin** representa o ***stream* de entrada** no C++. Ele realiza a leitura de uma seqüência de dados, sem espaços e sem tabulações, vindas do teclado.

Entrada e Saida de Dados

```
#include <iostream>
using namespace std;
int main(){
  int Num1;    int Num2;
  cout << "Lendo o primeiro número...: ";
  cin >> Num1;      cout << endl;
  cout << "Lendo o segundo número....: ";
  cin >> Num2;      cout << endl;
  return 0;   }
```

Entrada de Strings

Em determinadas ocasiões, deseja-se dados que contenham *strings* com tabulações, espaços em branco e/ou novas linhas. Frases são exemplos onde este tipo de situação ocorre.
Para englobar essas situações, C++ oferece o uso da função-membro **getline**. Esta função remove o delimitador do *stream* (isto é, lê o caractere e o descarta) e armazena o mesmo em uma variável definida pelo usuário.

Exercícios

1. Faca um programa em que pede ao usuário para introduzir o nome e a idade exibindo na tela.
2. Elabore um programa que calcula a media de 3 testes introduzidos pelo usuário.
3. Elabore um programa que calcule a soma, a razão e multiplicação de 2 valores introduzidos pelo teclado.

Exercícios

4. Faça um programa que peça para o usuário informar: Nome completo, sexo, idade, endereço, telefone e curso, na saida imprima estes dados na tela.

Exercícios

5. Elabore um programa para calcular a area de um triângulo e dizer se e equilatero, isoscele ou escaleno.

6. Faça um programa que determina o máximo e o mínimo de um conjunto de n números inteiros armazenados num vector A de 10 elementos.