Linguagem C++

1.Origem e Influência
2.Classificação:
2.1.Alto Nível
2.2.Paradigma Orientada a Objetos
2.3.Estática
3.Expressividade:
3.1.Sobrecarga de Métodos
4.Códigos Representativos

1.Origem e Influência

C++ é uma linguagem de programação criada por Bjarne Stroustrup no início da década de 1980. Baseado em C e em Simula, Hubbard,em 2003,acrescentou que é atualmente uma das linguagens mais populares para programação orientada a objetos. Foi padronizada em 1998 pelo American National Standards Institute (ANSI) e pela International Standards Organization (ISO). Possui o mecanismo classe/objeto, permite herança simples e herança múltipla e sobrecarga de operadores e funções.

O C++ tem uma enorme variedade de códigos, pois além de seus códigos, pode contar com vários da linguagem C. Esta variedade possibilita a programação em alto e baixo níveis. O C++ apresenta grande flexibilidade, embora seja bom, este fato faz com que a programação seja muito mais cuidadosa para não terem erros.

Alguns fatos sobre C++:

- O C++ é uma linguagem criada para ser tão eficiente quanto o C, porém com novas funções.
- É uma linguagem que suporta múltiplos paradigmas
- A linguagem da liberdade para o programador escolher as opções, mesmo sendo a opção errada.
- Muitos códigos podem ser transferidos para C facilmente, pois o C++ foi criado para ter compatibilidade com o C.
- A linguagem não tem privilégios para alguns grupos de programadores, os comandos são feitos para todas as especialidades de programadores
- Não é necessário um ambiente de desenvolvimento muito potente para o desenvolvimento de C++.

2.Classificação

2.1.Alto Nível

– São linguagens voltadas para o ser humano. Em geral utilizam sintaxe mais estruturada, tornando o seu código mais fácil de entender.

– São linguagens independentes de arquitetura.

• Um programa escrito em uma linguagem de alto nível, pode ser migrado de uma máquina a outra sem nenhum tipo de problema.

– Permitem ao programador se esquecer completamente do funcionamento interno da máquina.

• Sendo necessário um tradutor que entenda o código fonte e as características da máquina.

– Vantagens:

• Por serem compiladas ou interpretadas, têm maior portabilidade, podendo ser executados em várias plataformas com pouquíssimas modificações.

• Em geral, a programação é mais fácil.

– Desvantagens:

• Em geral, as rotinas geradas (em linguagem de máquina) são mais genéricas e, portanto, mais complexas e por isso são mais lentas e ocupam mais memória.

2.2.Paradigma Orientado a Objetos

– Tratam os elementos e conceitos associados ao problema como objetos;

– Objetos são entidades abstratas que embutem dentro de suas fronteiras, as características e operações relacionadas com a entidade real;

– Sugere a diminuição da distância entre a modelagem computacional e o mundo real:

• O ser humano se relaciona com o mundo através de conceitos de objetos;

• Estamos sempre identificando qualquer objeto ao nosso redor;

• Para isso lhe damos nomes, e de acordo com suas características lhes classificamos em grupos;

– Sistemas são vistos como coleções de objetos que se comunicam, enviando mensagens, colaborando para dar o comportamento global dos sistemas.

– Uma aplicação é estruturada em módulos (classes) que agrupam um estado (atributos) e operações (métodos) sobre este;

– A classe é o modelo ou molde de construção de objetos. Ela define as características e comportamentos que os objetos irão possuir.

– A orientação a objetos permite que classes possam "herdar" as características e métodos de outra classe para expandi-la ou especializá-la de alguma forma.

– Vantagens:

- Organização do código;
- Aumenta a reutilização de código;
- Reduz tempo de manutenção de código;
- Ampla utilização comercial;

– Desvantagens:

- Menos eficientes;

Exemplo:

```
   class Circle {                    // classname
private:
      double radius;            // Data members (variables)
      string color;
public:
      double getRadius();       // Member functions
      double getArea();
}
```

2.3.Estática

A definição básica da tipagem estática que uma linguagem de programação pode ter como característica é que há uma verificação dos tipos usados em dados e variáveis para garantir que sempre está sendo usado um tipo que é esperado em todas as situações. Esta verificação é feita no código fonte pelo processo de compilação. Esta análise ajuda na chamada **segurança de tipos** na

utilização dos dados pelo programa permitindo que o programador **se preocupe menos** com esta questão. O compilador fornece garantias que alguns problemas não poderão ocorrer após o programa passar por esta verificação, ou seja, erros são detectados logo, antes do programa ser efeticamente executado.

Uma variável não pode mudar seu tipo.

No entanto a tipagem estática pode causar uma falsa sensação de segurança. Só uma parte dos erros podem ser descobertos antecipadamente.

Exemplo:

```
int num, sum;

num = 5;

sum = 10;

sum = sum + num;
```

Alguns dos mais conhecidos programas são feitos em C++, ou parte dos seus códigos são nessa linguagem.Alguns deles são:

- Adobe Photoshop;
- MySQL;
- Mozilla Firefox;
- Internet Explorer;
- Microsoft Windows,etc.

3.Expressividade

3.1.Sobrecarga de Métodos

Sobrecarga de método permite a existência de vários métodos de mesmo nome, porém com assinaturas levemente diferentes ou seja variando no número,tipo de argumentos ,no valor de retorno e até variáveis diferentes. Ficará a cargo do compilador escolher de acordo com as listas de argumentos os procedimentos ou métodos a serem executados.

Exemplo:

1 - C++

```
#include <iostream>

using namespace std;


int soma(int x, int y) {
```

```
  return x + y;
}

double soma(double x, double y) {
  return x + y;
}

int main() {
  int myNum1 = soma(8, 5);
  double myNum2 = soma(4.3, 6.26);
  cout << "Int: " << myNum1 << "\n";
  cout << "Double: " << myNum2;
  return 0;
}
```

2 - Java

```
public class Soma {

   public static int soma(int x,int y){
      return x + y;
   }

   public static double soma(double x,double y){
     return x + y;
   }

   public static void main(String[] args){
      System.out.println("Int: " + soma(8,5));
      System.out.println("Double: " + soma( 4.3 , 6.26 ));
   }
```

4.Códigos Representativos

Comparações:

C

```
#include<stdio.h>
  int main()
  {
     printf("Olá Mundo!\n!");
}
```

C++

```
#include<iostream>
int main()
{
    std::cout<<"Olá Mundo!"<<"\n";
}
```

Python

```
    print "Hello World"
```

Bibliografia:

https://www.infoescola.com/informatica/cpp/

http://edirlei.3dgb.com.br/aulas/clp/CLP_Aula_02_Classificacao_Linguagens_Programacao_2015.pdf

https://pt.wikipedia.org/wiki/Sobrecarga_de_m%C3%A9todo