

[Lista de estúdios de videojogos em Portugal](#)
[Lista de videojogos feitos em Portugal](#)
[Make contact :\)](#)

POO, conceitos (cap 2)

Posted in [programação \(C plus plus\)](#) - 15/10/2018 - 0 Comment

Operadores de saída e entrada:

Operador de saída: << Operador de entrada: >>

podemos usar o namespace para organizar ou evitar conflitos, inclusive podemos criar o nosso próprio exemplo:

```
namespace exemplo {
// definições de funções, variáveis,
// classes, etc.
}
```

e para evitar o uso dos operadores :: aquando da biblioteca iostream

```
std::cout
```

```
faz-se:
```

```
using namespace std;
```

Bibliotecas a usar:

```
#include <iostream>
```

```
#include <string>
```

Cast: conversão explícita de tipo:

static_cast, de ponteiros do tipo void* para ponteiros de outros tipos

exemplo:

```
double d = 0.0;
```

```
int x = d;
```

```
x = static_cast(d);
```

```
const_cast,
```

```
reinterpret_cast,
```

dynamic_cast, Converte um ponteiro para um uma classe, num ponteiro para uma classe derivada, derivada da derivada

auto, o compilador determina automaticamente o tipo de uma variável

exemplo:

```
double val1 = 1;
```

```
double val2 = 2;
```

```
auto x = val1 + val2;
```

Ciclo for range-based, para quando todos os elementos de uma colecção (array, vector, ...) devem ser processados

```
for(auto & elemento: tabela)
```

exemplo:

```
int a[10];
```

```
for(auto & elemento: a){
```

```
elemento = 4;
```

```
}
```

A constante nullptr em C++11, A constante nullptr pode ser usada para inicializar ponteiros nulos.

Referências, pode:

ser passada a uma função

PWEB TP Final

31/12/2018

POO TP Intercalar

**entrega do TP
Intercalar**
25/11/2018

POO TP Final

entrega do TP
01/01/2019

SO TP 2 meta

02/12/2018

SO TP Final

01/01/2019

Blender

- [Blender Guru](#)
- [blender tutorials](#)
- [cgmasters – blender](#)
- [creativeblog – Blender](#)

Bolsa de trabalho TI

- [itjobs PT](#)
- [OIMT \(portugal\)](#)

consolas - hardware

- [cabos modernos](#)

modelação - blender

- [blender total](#)

programar - livros

Privacidade - Termos de Utilização

ser retornada por uma função
ser criada de forma independente.

Ponteiro como parâmetro de uma função

exemplo:

```
void f(int *n);

int main() {
    int i = 0;
    f(&i);
    cout << "\nNovo valor de i:" << i << endl;
    return 0;
}

void f(int *n) {
    *n = 100;
}
```

Referência como parâmetro de uma função

exemplo:

```
void f(int &n);

int main() {
    int i = 0;
    f(i);
    cout << "\nNovo valor de i:" << i << endl;
    return 0;
}

void f(int &n) {
    n = 100;
}
```

Referência como retorno de uma função

exemplo:

```
int &f( );

int x = 0; //tem que ser global

int main( ){
    f( ) = 100;
    cout << x << endl;
    return 0;
}

int &f( ){
    return x;
}
```

Constantes, uma constante deve ser inicializada, já que não é possível fazer-lhe atribuições.

A palavra const antes do * impede qualquer alteração ao objecto apontado através do ponteiro em causa

A palavra const depois do * torna constante o ponteiro.

A palavra const antes e depois do * torna impossível alterar o objeto apontado

Quando usar: quando uma função não deve alterar uma variável

Parâmetros do tipo referência para constante

_ganhar eficiência de passar a referência ao invés do valor

exemplo conclusão

```
1 #include <iostream>
2
3 void t(int*) {}
4
5 void u(const int* cip) {
6     // *cip = 2; // ERRO: modifica valor
7     int i = *cip; // OK: copia valor
8     // int* ip2 = cip; // ERRO: ip2 não é pont. para const
9 }
```

- [97 Things Every Programmer Should Know](#)
- [Building Skills in Programming](#)
- [Essential Coding Theory](#)
- [Foundations of Programming](#)
- [How to Design Programs](#)
- [How to Think Like a Computer Scientist](#)
- [The Codeless Code](#)
- [The Little Introduction To Programming](#)

tools

- [Download videos](#)
- [Rufos – iso to USB](#)

Video Tutorials - making vide games

- [curso: heartbeat \[GM\]](#)
- [Shaun Spalding – Video Tutorials](#)
- [Tom Francis – Video Tutorials](#)

videojogos - blogs

- [blog – Unseen64: Beta and Cancelled Videogames](#)
- [blog – Vida Extra](#)

videojogos - conferencia

- [Agenda eventos videojogos](#)
- [DevGAMM Conference](#)
- [IndieCade Europe](#)
- [International Conference on Entertainment Computing](#)

videojogos - Devblogs

- [Ganbatte Devblog](#)

videojogos - framework

- [Cocos2DX](#)
- [createjs](#)
- [flixel](#)
- [game froot](#)
- [Godot – opensource](#)
- [kiwijs](#)
- [Marmalade SDK](#)
- [phaser](#)
- [pico 8](#)
- [RPG Maker](#)
- [Torque](#)
- [wmi5](#)

```
10
11  const char* v() {
12  // Retorna o endereço de uma string:
13  return "result of function v()";
14  }
15
16
17  void main()
18  {
19  int x = 0;
20  int* ip = &x;
21  const int* cip = &x;
22  t(ip);
23  //t(cip);//Erro:t() nao pode receber ponteiro para constante
24  u(ip); // OK u() pode receber ponteiro sem restricoes
25  u(cip); // OK u() pode receber ponteiro para constante
26  //char* cp = v(); // Erro: nao deve ser possivel alterar
27  // a string retornada por v()
28  const char* ccp = v(); // OK
29  system("pause");
30  }
```

faltou:

referências a lvalue e rvalue

Tags : [cap02](#), [POO](#)

Profile

[Sign in with Twitter](#) [Sign in with Facebook](#)

or

Comment

Name

Email

Not published

Website

Post It

This site uses Akismet to reduce spam. [Learn how your comment data is processed](#).

No gamedev mas gamificação, que também é interessante (em gráficos)

POO, conceitos (cap 3)

videojogos - gameMaker

- [Game Maker Tutorials](#)
- [gamemaker studio book](#)

videojogos - investigação

- [European Conference on Games Based Learning – conferência](#)
- [game studies](#)
- [international journal game based learning journal](#)
- [revista push start – entrevistas](#)

videojogos - noticias

- [game from scratch](#)
- [GameDev map](#)
- [Games Industry](#)
- [indie db](#)

videojogos - parking

- [game jolt](#)
- [itch.io](#)

videojogos - recursos

- [Cubebrush](#)
- [free assets – kenney.nl](#)
- [Game AI Pro](#)
- [Game Design Quest](#)
- [Game designing tools](#)
- [game-development by david.arcila](#)
- [gameart2d – free](#)
- [gamedev by intel](#)
- [gamedev resources](#)
- [open game art](#)
- [Poly 3D](#)
- [Textos para as narrativas](#)

videojogos - retro

- [BBS – Artisanal Retromodding](#)
- [retro collect](#)

videojogos - tools

- [bug me not \(user/pass\)](#)
- [MagicaVoxel](#)
- [Paint of Persia](#)
- [piskel](#)
- [texture packer](#)