

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

POO, conceitos (Cap-05, Memoria dinamica)

Posted in [programação \(C plus plus\)](#) - 07/11/2018 - 0 Comment

#arrays de objetos

os arrays de objetos

```

1  #include "pch.h"
2  #include <iostream>
3  #include <string>
4
5  using namespace std;
6
7  class Ponto {
8      int x, y;
9  public:
10     Ponto(int x0 = 0, int y0 = 0);
11     ~Ponto();
12 };
13
14 int main()
15 {
16     //ou indicamos o construtor que queremos usar
17     Ponto a[] = { Ponto(1,3), Ponto(2,4), Ponto(5,7) };
18     Ponto b[] = { {1,3}, {2,4},{5,7} };
19
20     //ou usamos o construtor por omissão
21     Ponto c[3];
22     Ponto d[3] = { Ponto(1,3) };
23
24     system("pause");
25 }

```

```

1  //main
2  #include "Relogio.h"
3
4  int main()
5  {
6      {
7          Relogio a[10]; //criar um array de 10 objetos
8      }
9      system("pause");
10 }

```

```

1  //Relogio.cpp
2  #include "Relogio.h"
3
4  Relogio::Relogio(int hora, int minuto, int segundo)
5  {
6      cout << "Relogio criado" << endl;
7  }
8  Relogio::~Relogio()
9  {
10     cout << "Relogio destruido" << endl;
11 }

```

```

1  //Relogio.h
2  #pragma once
3  #include "pch.h"
4  #include <iostream>
5  #include <string>
6
7  using namespace std;
8
9  class Relogio
10 {
11 public:
12     Relogio(int hora = 0, int minuto = 0, int segundo = 0);
13     ~Relogio();
14 };

```

#variaveis em memoria dinâmica

o uso do operador new permite criar um objecto até ser explicitamente destruido com o operador delete

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](#)
- [creativebloq – Blender](#)

Bolsa de trabalho TI

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

consolas - hardware

- [cabos modernos](#)

modelação - blender

- [blender total](#)

programar - livros

Privacidade - Termos de Utilização

```
pVar = new tipo; //ponteiro para um endereço
delete pVar;
```

ou podemos usar da seguinte forma

```
int *v1 = new int(4); //valor inicial a 4
delete v1;
```

#objectos em memória dinâmica

ou aplicar a uma classe

```
1  #include "pch.h"
2  #include <iostream>
3  #include <string>
4
5  using namespace std;
6
7  class Ponto {
8      int x, y;
9  public:
10     Ponto() {
11         x = 0; y = 0;
12         cout << "construindo" << endl;
13     };
14     ~Ponto() { cout << "destruindo" << endl; };
15
16     void imprimir() { cout << "valor x=" << x << " valor de y=" << y <<
17 };
18
19 int main()
20 {
21     {
22         //usar memoria dinamica
23         Ponto *p = new Ponto;
24         p->imprimir();
25         delete p;
26     }
27
28     system("pause");
29 }
```

#arrays dinamicos de objetos

para criar um é necessário fazer:

```
pVar = new tipo[dim];
```

para libertar a memória:

```
delete [] pVar;
```

```
1  #include "pch.h"
2  #include <iostream>
3  #include <string>
4
5  using namespace std;
6
7  class Ponto {
8      int x, y;
9  public:
10     Ponto() {
11         x = 0; y = 0;
12         cout << "construindo" << endl;
13     };
14     ~Ponto() { cout << "destruindo" << endl; };
15
16     void imprimir() { cout << "valor x=" << x << " valor de y=" << y <<
17 };
18
19 int main()
20 {
21     {
22         //usar arrays dinamico de objectos
23         Ponto *p = new Ponto[4];
24         for (int i = 0; i < 4; i++) {
25             p[i].imprimir();
26         }
27         delete [] p;
28     }
29
30     system("pause");
31 }
```

Tags : [arrays de objetos](#), [cap05](#), [Memória dinamica](#), [POO](#)

Profile

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

- [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](#)