

Laboratório de programação II

Classes

Atributos e métodos

Edson Moreno

edson.moreno@pucrs.br

<http://www.inf.pucrs.br/~emoreno>

Introdução

- Classe

```
class Ponto {  
    public:  
        int x, y;  
        void set(int a, int b) {  
            x = a; y = b;  
        }  
};
```

Introdução

- Comparação Struct vs Classe

```
class Ponto {  
    public:  
        int x, y;  
        void set(int a, int b) {  
            x = a; y = b;  
        }  
};
```

```
typedef struct{  
    int x, y;  
}Ponto;
```

```
void set(Ponto *p, int x, int y) {  
    p->x = x; p->y = y;  
}
```

Introdução

- Manipulação de uma classe

```
#include <stdio.h>
```

```
class Ponto {  
    public:  
        int x, y;  
        Ponto(void) {  
            x = y = 0;  
        }  
        ~Ponto(void) {}  
};
```

```
...
```

```
int main (void){  
    Ponto p1;  
    Ponto p2();  
  
    Ponto *pp;  
    pp = new Ponto();  
  
    printf("%d %d ", p1.x, pp->x);  
    return 0;  
}
```

Introdução

- Nomenclatura e inicialização em C vs C++

C

```
Ponto p; //variável  
Ponto *p = (Ponto*) malloc(sizeof(Ponto));
```

C++

```
Ponto p; //Objeto  
Ponto *p = new Ponto();
```

Introdução

- Métodos/Funções em C e C++

C

```
Ponto p;  
inicia(&p, 10, 20);  
  
Ponto *p = (Ponto*)malloc(sizeof(Ponto));  
inicia(p, 19, 20);
```

C++

```
Ponto p;  
p.inicia(10, 20);  
Ponto p2(3,3);  
  
Ponto *p = new Ponto();  
p->inicia(10, 20);
```

Implementação de classes

- Organização da codificação em C++
 - Arquivos são separados em .h e .cpp

```
ponto.h  
  
#ifndef __PONTO_H__  
#define __PONTO_H__  
#include <stdio.h>  
  
class Ponto{  
Private:  
    int x, y;  
public:  
    Ponto();  
    void setX(int valor);  
    void setY(int valor);  
    void imprime();  
};  
  
#endif
```

Implementação de classes

- Organização da codificação em C++
 - Arquivos são separados em .h e .cpp

ponto.cpp

```
#include "ponto.h"

Ponto::Ponto() {
    x = y = 0;
}

Ponto::setX(int valor) {
    x = valor;
}

Ponto::setY(int valo) {
    y = valor;
}

void Ponto::imprime() {
    printf("\n Dados do ponto: (x=%d, y=%d) ", x, y);
}
```


Conceitos básicos

- Métodos básicos
 - Construtor
 - Não tem associado um tipo de retorno
 - Set e get
 - Usado para manipular cada uma das variáveis privadas/protegidas
- Tipos de proteção de atributos e métodos
 - Private:
 - somente a classe pode manipular o atributo diretamente
 - Protect:
 - somente a classe e seus decendentes podem manipular o atributo diretamente
 - Public:
 - Qualquer elemento pode manipular a classe

Compilação

- Organização da codificação em C++
 - Incluindo o arquivo .h na main.cpp e compilando o código

```
main.cpp  
  
#include "ponto.h"  
  
void main (void)  
{  
    Ponto p1(10, 40);  
    p1.setX(10);  
    p1.setY(40);  
    p1.imprime();  
}
```

- Compilação
 - `g++ ponto.cpp main.cpp -o run.x`