

Programação Orientada a Objetos em C++

Construtores e Destrutores

Agostinho Brito

2020

Construtores de uma classe

- O que acontece quando um objeto é criado? Qual o seu estado inicial?
- O estado inicial de qualquer variável é indefinido, por isso elas devem ser inicializadas.
- Em C++, o estado de um objeto pode ser inicializado usando **Métodos Construtores**.
- Os construtores são **funções-membro** que possuem **O mesmo nome da classe**.

```
1 class MinhaClasse{  
2 private:  
3     // membros privados  
4 public:  
5     MinhaClasse(); // Construtor PADRAO  
6 };
```

- Pode haver mais de uma sobrecarga para os construtores.

Construtores de uma classe

- O que acontece quando um objeto é criado? Qual o seu estado inicial?
- O estado inicial de qualquer variável é indefinido, por isso elas devem ser inicializadas.
- Em C++, o estado de um objeto pode ser inicializado usando **Métodos Construtores**.
- Os construtores são **funções-membro** que possuem O mesmo nome da classe.

```
1 class MinhaClasse{
2 private:
3     // membros privados
4 public:
5     MinhaClasse(); // Construtor PADRAO
6 };
```

- Pode haver mais de uma sobrecarga para os construtores.

Destruítores da classe

- Tá, mas e quando o objeto está para sumir? É possível realizar alguma ação?
- E, se sim, **POR QUE?**
- Muitas vezes, o construtor aloca recursos do sistema que precisam ser liberados antes do fim do ciclo de vida do objeto.
- Nestes casos, é papel do **Método destrutor** realizar essa liberação de recursos.
- Em C++, o destrutor é **ÚNICO** e possui o **MESMO NOME DA CLASSE**, precedido pelo caractere `~`.

```
1 class MinhaClasse{
2     private:
3         // membros privados
4     public:
5         ~MinhaClasse(); // Destruítor
6     };
```


Destruítores da classe

- Tá, mas e quando o objeto está para sumir? É possível realizar alguma ação?
- E, se sim, **POR QUE?**
- Muitas vezes, o construtor aloca recursos do sistema que precisam ser liberados antes do fim do ciclo de vida do objeto.
- Nestes casos, é papel do **Método destrutor** realizar essa liberação de recursos.
- Em C++, o destrutor é **ÚNICO** e possui o **MESMO NOME DA CLASSE**, precedido pelo caractere `~`.

```
1 class MinhaClasse{
2     private:
3         // membros privados
4     public:
5         ~MinhaClasse(); // Destruítor
6 };
```

Destruutores da classe

- Tá, mas e quando o objeto está para sumir? É possível realizar alguma ação?
- E, se sim, **POR QUE?**
- Muitas vezes, o construtor aloca recursos do sistema que precisam ser liberados antes do fim do ciclo de vida do objeto.
- Nestes casos, é papel do **Método destrutor** realizar essa liberação de recursos.
- Em C++, o destrutor é **ÚNICO** e possui o **MESMO NOME DA CLASSE**, precedido pelo caractere **~**.

```
1  class MinhaClasse{  
2  private:  
3      // membros privados  
4  public:  
5      ~MinhaClasse(); // Destruitor  
6  };
```



Obrigado