C++

Templates

Softblue
cursos online

O problema: redundância

- Redundância baseada exclusivamente no tipo de dado

```
class ExibeInt {
public:
    int a;
}
```

```
class ExibeFloat {
public:
    float a;
}
```

```
class ExibeChar {
public:
    char a;
}
```

```
class Exibe {
public:
    <___> a;
}
```

A solução: templates

- Torna universal o tipo de dado redundante
- Generalização do tipo de dados, similar ao recurso ***generics*** do Java
- Permite criar uma classe única e informar o tipo de dados dinamicamente
- Extremamente prático para operações iguais com diferentes tipos de dados
- Simples implementação
- Atenção nos cabeçalhos de funções

Implementando templates

Softblue

.h

```
template <class T>

class Exemplo
{
public:
    T minhaVar;
    Exemplo(T tmpVar);
    void exibir();
}
```

.cpp

```
#include ".h"

template <class T>

Exemplo<T>::Exemplo(T tmpVar)
{
    minhaVar = tmpVar;
}

void Exemplo<T>::exibir()
{
    cout << minhaVar;
}
```

Templates com dois tipos de dados

Softblue

.h

```
template <class T1, class T2>

class Exemplo
{
public:
    T1 minhaVar;
    T2 outraVar;
    Exemplo(T1 va, T2 vb);
    void exibir();
}
```

.cpp

```
#include ".h"

template <class T1, class T2>

Exemplo<T1, T2>::Exemplo(T1 va, T2 vb)
{
    minhaVar = va;
    outraVar = vb;
}

void Exemplo<T1, T2>::exibir()
{
    cout << minhaVar << outraVar;
}
```

Instanciando um template

Softblue

Arquivo main

```
// template <class T1, class T2> class Exemplo

Exemplo<int, double> *a = new Exemplo<int, double>();

Exemplo<int, int> *b = new Exemplo<int, int>();

Exemplo<char, float> *c = new Exemplo<char, float>();
```

Considerações gerais

- Generalização de classes
- Simplicidade de uso
- Geralmente utilizado para:
  - Generalizar coleções de objetos
  - Algoritmos de processamento
- Arquitetura dependente do desenvolvedor
  - struct x classes x templates
- Praticar, praticar e... praticar!

Aulas práticas e manuais on-line

Assista agora as aulas práticas.

Clique aqui para visualizar as aulas práticas disponíveis