



Universidade Federal do ABC
Centro de Matemática, Computação e Cognição

Templates

Monael Pinheiro Ribeiro, D.Sc.

Exemplo

- Suponha que você deve implementar uma função que faça a soma de dois valores.

Exemplo

- Suponha que você deve implementar uma função que faça a soma de dois valores.
 - $2+5 = 7$
 - $3.1 + 6.4 = 9.5$
 - $UF + ABC = UFABC$
- O que você faria?

Exemplo

- Solução:

```
int soma(int a, int b)
{
    return a+b;
}
```

```
double soma(double a, double b)
{
    return a+b;
}
```

```
std::string soma(std::string a, std::string b)
{
    return a+b;
}
```

Exemplo

- Solução:

```
int main()
{
    int a=2, b=5;
    double c=3.1, d=6.4;
    std::string f="UF", g="ABC";

    std::cout << soma(a, b) << std::endl;
    std::cout << soma(c, d) << std::endl;
    std::cout << soma(f, g) << std::endl;

    return 0;
}
```

Exemplo

- Solução:

```
int main()
{
    int a=2, b=5;
    double c=3.1, d=6.4;
    std::string f="UF", g="ABC";

    std::cout << soma(a, b) << std::endl;
    std::cout << soma(c, d) << std::endl;
    std::cout << soma(f, g) << std::endl;

    return 0;
}
```



7
9.5
UFABC

Exemplo

- Solução:

```
int soma(int a, int b)
{
    return a+b;
}
```

```
double soma(double a, double b)
{
    return a+b;
}
```

```
std::string soma(std::string a, std::string b)
{
    return a+b;
}
```

Perceba que só os tipos mudam, a lógica é exatamente a mesma.

Templates

- Define uma família de classes e funções.
- Com templates é possível parametrizar os tipos de dados que serão resolvidos em tempo de compilação.
- Essa técnica é conhecida como Programação Genérica.

Exemplo com Templates de Funções

- Solução:

```
template <typename T> T soma(T a, T b)
{
    return a+b;
}
```

Exemplo com Templates de Funções

- Solução:

```
template <typename T> T soma(T a, T b)
{
    return a+b;
}
```

- O símbolo T é chamado de parametro de tipo.
- Trata-se de um handle que é trocado pelo tipo ou classe no momento que a função é invocada.
- Agora o template da função soma() funciona para todos os tipos cujo operador + esteja definido.

Exemplo com Templates de Funções

- Solução:

```
int main()
{
    int a=2, b=5;
    double c=3.1, d=6.4;
    std::string f="UF", g="ABC";

    std::cout << soma(a, b) << std::endl;
    std::cout << soma(c, d) << std::endl;
    std::cout << soma(f, g) << std::endl;

    return 0;
}
```



7
9.5
UFABC

Templates de Classes

- São como os templates de funções, porém gerando classes genéricas no lugar das funções.

- Sintaxe:

```
template <class T>
class Classe
{
    ...
};
```

Templates de Classes

- São como os templates de funções, porém gerando classes genéricas no lugar das funções.

- Sintaxe:

```
template <class T>
class Classe
{
    ...
};
```

- Podendo ter mais de um tipo parametrizado, inclusive incluindo tipos primitivos:

```
template <class T, int n>
class Classe
{
    ...
};
```

Templates de Classes

- Os membros uma classe template podem usar o template da classe.

```
template <class T>
class ClasseExemplo
{
    private:
        T atributo;
    public:
        T soma(T a, T b)
        {
            return a+b;
        }
};
```

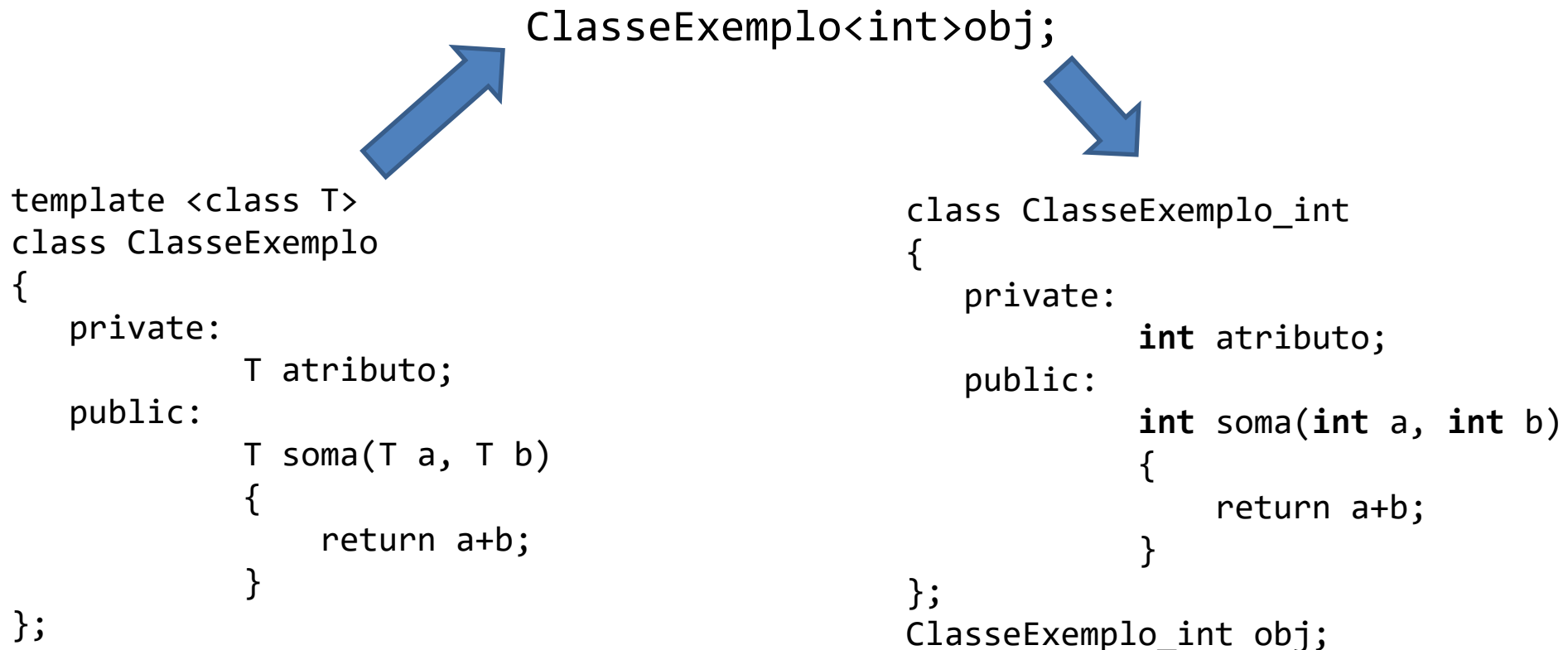
Templates de Classes

- O compilador faz a troca do parâmetro de tipo pelo informado na declaração do objeto gerando uma classe e um objeto específico a partir do template.

```
ClasseExemplo<int>obj;
```

Templates de Classes

- O compilador faz a troca do parâmetro de tipo pelo informado na declaração do objeto gerando uma classe e um objeto específico a partir do template.



Exemplo Templates de Classes

- Template de uma Pilha Estática:

```
template <class T>
class Pilha
{
    private:
        T *itens;
        int topo;
    public:
        Pilha();
        Pilha(int);
        void setTopo(int);
        int getTopo();
        void setItens(int, T);
        T getItens(int);
        void empilha(T);
        void desempilha();
        T topo();
        bool cheia();
        bool vazia();
};
```

Exemplo Templates de Classes

- Template de uma Pilha Estática:

```
template <class T>
Pilha<T>::Pilha()
{ }
```

```
template <class T>
Pilha<T>::Pilha(int n)
{
    this->itens = new T[n];
    this->setTopo(-1);
    this->setTamanho(n);
}
```

```
template <class T>
void Pilha<T>::setTamanho(int n)
{
    this->tamanho = n;
}
```

```
template <class T>
int Pilha<T>::getTamanho()
{
    return this->tamanho;
}
```

```
template <class T>
void Pilha<T>::setTopo(int t)
{
    this->topo = t;
}
```

```
template <class T>
int Pilha<T>::getTopo()
{
    return this->topo;
}
```

```
template <class T>
void Pilha<T>::setItens(int i, T it)
{
    this->itens[i] = it;
}
```

```
template <class T>
T Pilha<T>::getItens(int i)
{
    return this->itens[i];
}
```

Exemplo Templates de Classes

- Template de uma Pilha Estática:

```
template <class T>
void Pilha<T>::empilha(T it)
{
    if(!this->cheia())
    {
        this->setTopo(this->getTopo()+1);
        this->setItens(this->getTopo(), it);
    }
}

template <class T>
void Pilha<T>::desempilha()
{
    if(!this->vazia())
    {
        this->setTopo(this->getTopo()-1);
    }
}

template <class T>
T Pilha<T>::top()
{
    if(!this->vazia())
    {
        return this->getItens(this->getTopo());
    }
}

template <class T>
bool Pilha<T>::cheia()
{
    return this->getTopo() == this->getTamanho()-1;
}

template <class T>
bool Pilha<T>::vazia()
{
    return this->getTopo() == -1;
}
```

Templates em JAVA

Templates em JAVA



Templates em JAVA



~~Templates~~ em JAVA

- Template em JAVA se chama Generics.

Generics em JAVA

- Template em JAVA se chama Generics.
- JAVA não usa o cabeçalho template, assim a

instrução:

```
template<class T>
class Pilha
{
    ...
};
```

- Em JAVA é:

```
public class Pilha<T>
{
    ...
}
```


Exemplo de Generics em JAVA

- Pilha Estática Generic

```
class Pilha<T>
{
    private T[] itens;
    private int topo;
    private int tamanho;

    public Pilha()
    { }
    public Pilha(int n)
    {
        this.itens = (T[])new Object[n];
        this.setTopo(-1);
        this.setTamanho(n);
    }
    public void setTopo(int t)
    {
        this.topo = t;
    }
    public int getTopo()
    {
        return this.topo;
    }
}
```

```
    public void setTamanho(int t)
    {
        this.tamanho = t;
    }
    public int getTamanho()
    {
        return this.tamanho;
    }
    public void setItens(int i, T it)
    {
        this.itens[i] = it;
    }
    public T getItens(int i)
    {
        return this.itens[i];
    }
    public T top()
    {
        return this.getItens(this.getTopo());
    }
}
```

Exemplo de Generics em JAVA

- Pilha Estática Generic

```
public boolean cheia()
{
    return this.getTopo() == this.getTamanho()-1;
}
public boolean vazia()
{
    return this.getTopo() == -1;
}

public void empilha(T it)
{
    if(!this.cheia())
    {
        this.setTopo(this.getTopo()+1);
        this.setItens(this.getTopo(), it);
    }
}
public void desempilha()
{
    if(!this.vazia())
    {
        this.setTopo(this.getTopo()-1);
    }
}
}
```