Funções Amigas

Funções Amigas

- Uma **função amiga** de uma classe é uma função definida completamente fora da classe
  - Porém, possui acesso aos membros públicos e não públicos de uma classe;
  - Funções isoladas ou mesmo classes inteiras podem ser declaradas como amigas de outra classe.
  - Funções amigas podem melhorar a performance de uma aplicação, e também são utilizadas na **sobrecarga de operadores** e na criação de **iteradores**.
- Podemos também declarar uma **classe amiga**
  - Todos os métodos terão acesso aos membros da outra classe.

Funções Amigas

- Para declararmos uma função amiga, utilizamos a palavra ***friend*** antes do protótipo da função dentro da classe
  - Note que a função não será um método.
- Se uma classe *ClasseUm* será declarada como amiga de uma classe *ClasseDois*, dentro de *ClasseUm* declaramos

friend class ClasseDois;

- **Atenção!**
  - Estas declarações devem ser feitas antes de qualquer especificador de acesso (*public*, *private*, etc.).

RelogioFriend.h

```
#include<iostream>
using namespace std;

class Relogio
{
      friend void alteraHMS(Relogio &r);
      public:
      Relogio(int, int, int);
      void setHora(int, int, int);
      void printHora();

      private:
      int h, m, s;
};
```

RelogioFriend.cpp

```
#include<iostream>
using namespace std;
#include "RelogioFriend.h"

Relogio::Relogio(int hr=0, int min=0, int seg=0)
{
    setHora(hr, min, seg);
}
void Relogio::setHora(int hr, int min, int seg)
{
      h = hr;
      m = min;
      s = seg;
}
void Relogio::printHora()
{
      cout<<h<<':'<<m<<':'<<s<<endl;
}
//passagem por referência
void alteraHMS(Relogio &r)
{
      r.h = 10;
      r.m = 10;
      r.s = 10;
}
```

Funções Amigas

- Note que o objeto deve ser enviado como parâmetro para a função
  - Uma vez que o objeto não chama a função, é necessário indicar qual é o objeto cujos atributos serão alterados;
  - Para que a alteração tenha efeito, precisamos enviar o objeto por referência
    - Por isso o uso do operador **&**.
- Qualquer outra tentativa de acesso a membros privados por funções que não são amigas de uma classe resultará em erro de compilação.

driverRelogioFriend.cpp

```
#include<iostream>
using namespace std;
#include"RelogioFriend.h"

int main()
{
	Relogio r(12,0,0);
	r.printHora();
	alteraHMS(r); //é necessário enviar o objeto como parâmetro
	r.printHora();

	return 0;
}
```