

Espacio de Nombres en C++

1. En tiempos remotos (antes de la estandarización y sin espacios de nombres):
`#include <iostream.h>`
2. En tiempos más cercanos (después de la estandarización y aparición de espacios de nombres):
`#include <iostream>`
`using namespace std;`
3. En tiempos actuales, además de lo anterior se recomienda usar:
`#include <string>`
`using std::string;`

Clases INLINE vs. OFFLINE

1. Escribir una clase INLINE significa colocar la declaración y la definición de la clase en un mismo archivo .h, por ejemplo:

.h

```
class a
{
    private:
        int _valor;
    public:
        int getValor() {
            return _valor;
        }
        void setValor(int _valor)
        {
            _valor=valor;
        }
};
```

El compilador reemplaza la definición de funciones inline en tiempo de compilación.

2. Escribir una clase OFFLINE, significa colocar la declaración o interfaz de la clase en el .h, y la definición o implementación en el .cpp, por ejemplo:

.h

```
class a
{
    private:
        int _valor;
    public:
        int getValor();
        void setValor(int);
};
```

.cpp

```
#include "prueba.h"

int prueba::getValor()
{
    return _valor;
}

void prueba::setValor(int a)
{
    _a=a;
}
```

3. ¿Cuál se debe usar?

OFFLINE, ya que cuando trabajamos inline el compilador inserta o incrusta una copia del cuerpo de cada método en cada lugar en el cual es llamado. Además, favorece el mantenimiento.

Guardas y librerías

1. Escribir SIEMPRE, primero las librerías estándar y luego los .h referentes a las clases nuevas creadas.
2. Por razones de eficiencia y orden se deben agregar guardas en cada archivo de cabecera.
3. Las guardas y librerías que coloquemos siempre deben llevar un orden especial

Por ejemplo:

```
#ifndef TRABAJADOR_H }
#define TRABAJADOR_H

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

#include "A.h"

class B
{
    ...
};

#endif }
```

Guarda que evita que la clase sea declarada más de una vez. Toda clase solo debe ser declarada una única vez en el programa. Estos se escriben únicamente en el .h de la clase.

Librerías estándar requeridas, las cuales deben ir antes del espacio de trabajo

Espacio de trabajo

Clases hechas por nosotros mismos y requeridas por el código. Importante que vayan luego del espacio de trabajo y de las librerías estándar

Fin de la guarda inicial