SOBRECARGA DE CONSTRUCTORES

Lic. Gladys Chuquimia

SOBRECARGA DE CONSTRUCTORES

- Permiten darle mayor flexibilidad a la inicializacion de variables al crear el objeto.
- Sintaxis:

```
class nombclase{
    //Miembros Dato
public:
  nombclase( );
  nombclase( td par1);
  nombclase( td par1, td par2);
  nombclase( td par1, td par2, td par3);
  nombclase( td par1, td par2, td par3, td parN);
};
```

CONTINUACION...

- Pueden ser declarados en linea (al interior de la clase o con la palabra clave inline), como fuera de linea.

```
nombclase::nombclase(td par1)
{
    // Instrucciones
}
```

EJEMPLO:

- En la clase cuadro que se considero, sobrecargue el constructor tres veces.

Cuadro
-X1 : int - y1 : int - x2 : int - y2 : int - fondo : int - frente : int
+ dibuja( ): void + oculta( ): void + coordenada(a, b, c, d): void

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
#include <process.h>
#include <dos.h>
#include <stdio.h>

class cuadro{
int x1, y1, x2, y2;
int fondo, frente;
public:
   cuadro()
   { x1 = getmaxx()/2 - 50;
     y1 = getmaxy()/2 - 50;
     x2 = getmaxx()/2 + 50;
     y2 = getmaxy()/2 + 50;
     fondo = 1;  frente = 15;
   }

   cuadro(int a, int b, int c, int d)
   { x1 = a;
     y1 = b;
     x2 = c;
     y2 = d;
     fondo = 1;  frente = 15;
   }
   cuadro(int a, int b);

   void dibuja();
   void dibuja(int color);
   void oculta();

   void coordenadas(int a, int b, int c, int d)
   { x1 = a; y1 = b; x2 = c; y2 = d;
   }

   void operator++(); //Sobrecarga de operador unario
   void operator - (int cant); //Sobrecarga de operador binario
   cuadro operator < (cuadro obj);
   friend void modografico();
};
```

```
cuadro::cuadro(int a, int b)
{ x1 = a;
  y1 = b;
  x2 = 10 * x1;
  y2 = 10 * y1;
  fondo = 1;  frente = 15;
}

void cuadro::dibuja( )
{ setfillstyle(1,frente);
  bar(x1, y1, x2, y2);
}

void cuadro::dibuja(int color)
{ frente = color;
  dibuja();
}

void cuadro::oculta( )
{ setfillstyle(1,fondo);
  bar(x1, y1, x2, y2);
}

void cuadro::operator ++( )
{
  outtextxy(0, getmaxy()-20, "Bienvenidos!");
}
```

```
void cuadro::operator - (int cant)
{ oculta();
  x1 = x1 + cant;
  y1 = y1 + cant;
  x2 = x2 - cant;
  y2 = y2 - cant;
  dibuja();
}

cuadro cuadro::operator < (cuadro obj)
{ int disp = x2 - x1;
  int diss = obj.x2 - obj.x1;
  if(disp < diss)
     return *this;
  else
     return obj;
}

void modografico( )
{ cuadro obj;
  int gd=DETECT, gm, sierror;
  initgraph(&gd,&gm,"c:\\borlandc\\bgi"); sierror = graphresult();
  if (sierror != 0)
  { cout<<"Error en modo grafico: "<<grapherrormsg(sierror)<<endl;
    cout<<"Presione una tecla para salir del programa..."; getch();
    exit(1);
  }
  setbkcolor(obj.fondo);
}

//Función para mostrar un mensaje
void mensaje(char *msg)
{ setfillstyle(1,1);
  bar(0,getmaxy()-30, getmaxx(), getmaxy()-10);
  outtextxy(0, getmaxy()-20, msg);
}
```

```
void main( )
{ modografico();
  cuadro a, d;
  cuadro b(10, 5);
  cuadro c(50, 70, 100, 90);

  ++a;
  getch();

  mensaje("Dibujando el primer objeto");
  a.dibuja();
  getch();

  mensaje("Dibujando el segundo objeto");
  b.dibuja();
  getch();


  mensaje("Dibujando el tercer objeto de color magenta");
  c.dibuja(5); //Color magenta
  getch();

  mensaje("Disminuyendo en 5 pixeles el primer objeto");
  a - 5;
  delay(2000);
  a - 5;
  delay(2000);

  mensaje("Dibujando de rojo el cuadro mas pequeno en eje x");

  d = a < b < c;
  d.dibuja(4);
  getch();


  closegraph();
}
```

¿CUÁNTOS CONSTRUCTORES SE EJECUTAN?

- cuadro a;
- cuadro b(10, 5);
- cuadro c(50, 70, 100, 90);

RESPUESTA

- Uno por cada Objeto (3 veces).
- Debe existir previamente su definición.

```
class cuadro{
int x1, y1, x2, y2;
int fondo, frente;
public:
   cuadro()
   { x1 = getmaxx()/2 - 50;
     y1 = getmaxy()/2 - 50;
     x2 = getmaxx()/2 + 50;
     y2 = getmaxy()/2 + 50;
     fondo = 1;  frente = 15;
   }

   cuadro(int a, int b, int c, int d)
   { x1 = a;
     y1 = b;
     x2 = c;
     y2 = d;
     fondo = 1;  frente = 15;
   }

   cuadro(int a, int b);

   void dibuja();
```

```
cuadro::cuadro(int a, int b)
{ x1 = a;
  y1 = b;
  x2 = 10 * x1;
  y2 = 10 * y1;
  fondo = 1;  frente = 15;
}
```