

Trabajo Práctico nº 2

Clases y Objetos en Visual Studio C++

Trabajo Preliminar

Crear una solución llamada **TrabajoPractico-2** que contendrá los proyectos: **Ejercicio-0, Ejercicio-1, Ejercicio-2, Ejercicio-3 y Ejercicio-4**.

Ejercicio 0

Este ejercicio estará compuesto por el último ejemplo del apunte Unidad 2(clases y objetos en C++) página 17 y 18 deberá completarlo y ejecutarlo.

Los métodos **setX** y **setY** tienen que setear los atributos **_x** e **_y** con los valores que vienen por parámetro.

Los métodos **getX** y **getY** tienen que devolver el valor de los atributos **_x** e **_y**.

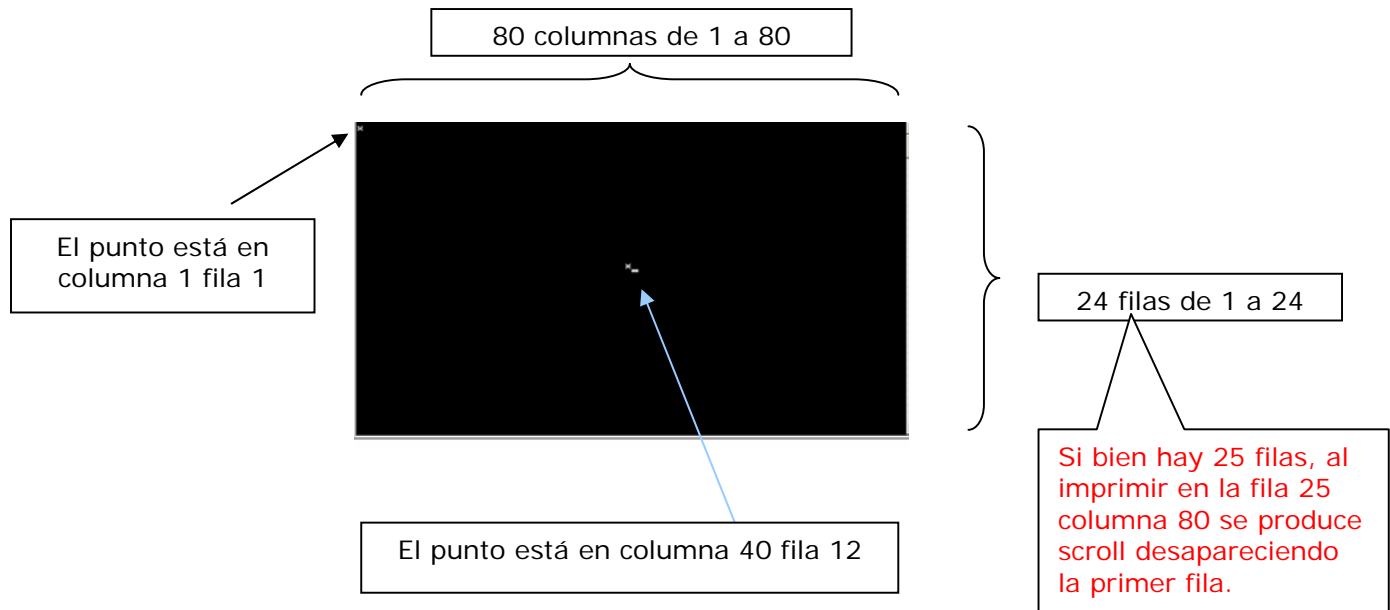
El método llamado **mostrar** imprime en consola: "[x, y]=_x,_y".

El método **dibujar** debe situarse en la pantalla utilizando gotoxy().

Deberá crear 2 objetos cuyas coordenadas son **1,1** y **40,12** mostrarlos por pantalla y destruir los objetos.

Punto
-int _x
-int _y
+void setX(int x)
+void setY(int y)
+int getX()
+int getY()
+void mostrar()
+void dibujar()
Representa a un Punto en el plano

Compile, linkee y pruebe el ejercicio tal que muestre lo siguiente en pantalla:



Ejercicio 1

Este ejercicio cuyo proyecto se llama **Ejercicio-1** deberá copiar la clase **Punto** del ejercicio 0 y completarla para que tenga dos constructores, un destructor y los métodos mover y borrar.

El constructor por defecto(no recibe parámetros) setea los atributos x e y en 0.

Sobrecargar el constructor (es decir, crear otro) que reciba 2 parámetros x e y y setear los atributos con los valores de estos parámetros.

Los métodos seteadores ahora deberán validar que los valores de x e y no superen los límites de la pantalla.

El método **mover** cambia las coordenadas del punto validando que superen los límites de la pantalla.

El método **borrar** borra el punto de la pantalla.

Punto
-int _x -int _y
+Punto() +Punto(int x,int y) +void setX(int x) +void setY(int y) +int getX() +int getY() +void mostrar() +void dibujar() +void mover(int x, int y) +void borrar()
Representa a un Punto en el plano

Deberá crear 2 objetos cuyas coordenadas son **1,1** y **40,12** mostrarlos por pantalla, mover el 2 objeto a la posición 15, 20 , mostrarlo y destruir los dos objetos. Compile, linkee y pruebe el ejercicio tal que muestre lo siguiente en pantalla:



Ejercicio 2

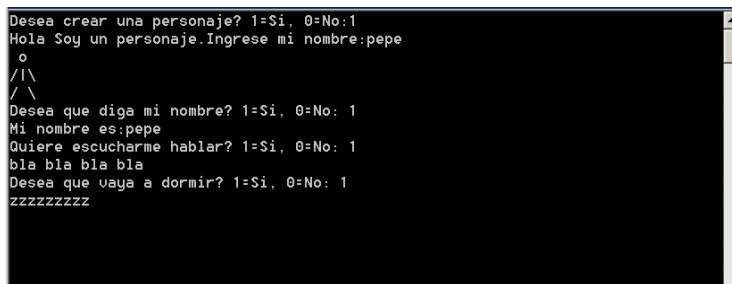
Este ejercicio cuyo proyecto llamado **Ejercicio-2** deberá escribir la clase **Personaje** cuyo diagrama de clase es el siguiente:

El método **dormir** imprime "zzzzzzzzzz"

El método **hablar** imprime "bla bla bla bla"

Probar la clase con 1 objeto Personaje compile, linkee y pruebe el ejercicio tal que muestre lo siguiente en pantalla.

Personaje
- string _nombre
+Personaje() +void dormir() +void hablar() +void setNombre(string nombre) +string getNombre() +void dibujar()
Representa a un personaje



El código para dibujar el personaje es :

```
cout<<" o\n";
cout<<" /|\n";
cout<<" / \n";
```

Ejercicio 3

Este ejercicio cuyo proyecto llamado **Ejercicio-3** deberá completar la clase **Personaje** cuyo diagrama de clase es el siguiente:

Con un atributo **_color** del tipo char, constructor parametrizado, setColor y getColor y modificar el método dibujar para que muestre el dibujo en colores utilizando la función setForecolor() y los #define que se encuentran en libreria.h (como ser COLOR_BLUE)

Recuerde que después de cambiar el color del texto este se mantiene hasta que se vuelva a cambiar.

El destructor imprime: "adios mundo cruel!"

Probar la clase con 1 objeto Personaje compile, linkee y pruebe el ejercicio tal que muestre lo siguiente en pantalla.

```

Desee crear un personaje? 1=Si, 0=No: 1
Hola soy un personaje. Ingrese mi nombre: pepe
/ \
/ \
Desee que diga mi nombre? 1=Si, 0=No: 1
Mi nombre es: pepe
Quiere escucharme hablar? 1=Si, 0=No: 1
bla bla bla bla
Desee que vaya a dormir? 1=Si, 0=No: 1
zzzzzzzzzz
Adios mundo cruel

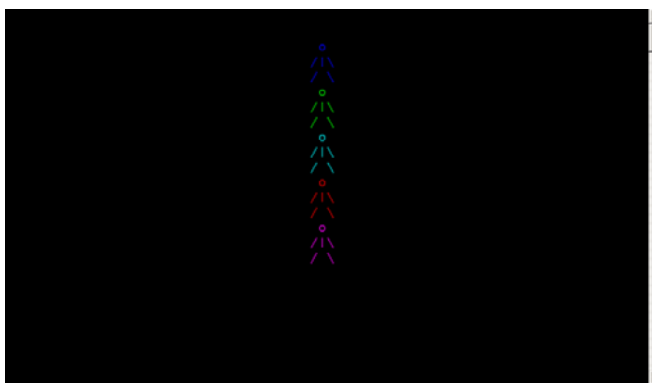
```

Personaje
-string _nombre -char _color
+Personaje() +Personaje(string nombre, char color) ~Personaje(); +void dormir() +void hablar() +void setNombre(string nombre) +void setColor(char color) +string getNombre() +char getColor() +void dibujar()
Representa a un personaje

Ejercicio 4

Este ejercicio cuyo proyecto llamado **Ejercicio-4** deberá completar la clase **Personaje** con el método **dibujar** sobrecargado tal que el dibujo salga en la posición x, y recibida por parámetro. Deberá esconder el curso con el método hideCursor().

Probar la clase con un array de 5 personajes compile, linkee y pruebe el ejercicio tal que muestre lo siguiente en pantalla.



Personaje
-string _nombre -char _color
+Personaje() +Personaje(string nombre, char color) ~Personaje(); +void dormir() +void hablar() +void setNombre(string nombre) +void setColor(char color) +string getNombre() +char getColor() +void dibujar() +void dibujar(int x, int y)
Representa a un personaje