

Trabajo Práctico nº 7

Atributos y Métodos estáticos en Visual Studio C++

Trabajo Preliminar

Crear una solución llamada **TrabajoPractico-7** que contendrá los proyectos: **Ejercicio-0, Ejercicio-1, Ejercicio-2.**

Ejercicio 0

Crear la siguiente clase:

```
class Bala{
    int _x;
    int _y;
    static int _cantBalas;
public:
    Bala(int x,int y);
    void dibujar();
    bool borrar();
    int getCantBalas();
    int getX(){ return _x;}
    int getY(){ return _y;}
};
void Bala::dibujar(){
    gotoxy(_x,_y);
    cout<<(char)'!';
}
void Bala::borrar(){
    gotoxy(_x,_y);
    cout<<" ";
}
```

Crear un programa tal que cree dos objetos **Bala** y muestre la cantidad de objetos creados tal que la salida sea:

La salida será:

```
la cantidad de balas creadas es:1
la cantidad de balas creadas es:2
```

Ejercicio 1

Crear la siguiente clase tal que posea un atributo y un método estático:

```
class Asteroide{
    int _x;
    int _y;
    static int _cantAsteroides; // Declaración atributo estático
public:
    Asteroide(int x, int y);
    ~Asteroide();
    void dibujar();
    void borrar();
    int getX(){ return _x;}
    int getY(){ return _y;}
    static int getCantAsteroides();// Ahora el método es estático.
};
```

```
void Asteroide::dibujar(){
    gotoxy(_x,_y);
    cout<<(char)'*';
}
void Asteroide::borrar(){
    gotoxy(_x,_y);
    cout<<" ";
}
}
```

Crear un programa tal que 3 objetos de clase **Asteroide** tal que la salida sea:

La cantidad de asteroides creados es:3

Borramos una instancia y ahora existen 2 instancias a la clase Asteroide

Borramos una instancia y ahora existen 1 instancias a la clase Asteroide

Existen 0 instancias a la clase Asteroide

Ejercicio 2

Crear la clase juego y completar los métodos siguientes:

```
Juego() // inicializa el array y crea 3 asteroides de posiciones (10,4),(4,8) y (15,10)
~Juego();// libera la memoria del array
draw();
// recorre el array, imprime los asteroides y muestra la cantidad en la fila 1 columna
60 // al finalizar pone _gameOver en true
update();// se deja vacío ya que por ahora no hay lógica de juego
```

```
#define MIN_FIL 1
#define MIN_COL 1
#define MAX_FIL 24
#define MAX_COL 80
#define TOPE 5

class Juego{
    int _tecla;
    bool _gameOver;
    Asteroide* vecAst[TOPE];
public:
    Juego();
    ~Juego();
    void init();
    bool gameOver();
    void play();
    void input();
    void update();
    void draw();
    void result();
};

Juego::Juego(){
    // completar
}

Juego::~~Juego(){
    // completar
}

void Juego::draw(){
    // completar
}

void update(){
    // se deja vacío ya que por ahora no hay lógica de juego
}
```

```
void Juego::init(){
    _tecla=' ';
    _gameOver=false;
    srand(time(0));
}
void Juego::play(){
    init();
    hideCursor();
    marco(1,2,80,24);
    while(!gameOver()){
        input();
        update();
        draw();
        miliSleep(100);
    }
    result();
}
void Juego::result(){
    if(_gameOver){
        gotoxy(30,25);
        cout<<"game over";
    }
}
void Juego::input(){
    if(_tecla=getKey(false)){
        switch(_tecla){
            case KEY_ESC:{
                _gameOver=true;
                break;
            }
        }
    }
}
bool Juego::gameOver(){
    return _gameOver;
}
```

Crear el archivo **Ejer-2.cpp** tal que cree un juego que dibuje los asteroides y su cantidad hasta apretar la tecla ESCAPE tal que la salida sea:

