

II. TRABAJO PRACTICO: Objetos en C++

A. Elementos necesarios para resolver los ejercicios

Clases. Atributos y funciones miembros. Instanciación. Objetos. Constructor. Destructor. Mensajes. Sobrecarga. Agregación. UML. Diagrama de clases. Diagrama de secuencia. Diagrama de Estados.

B. Forma de Entrega

Luego de resolver las consignas entregue un archivo comprimido, identificado con su apellido y nombre que contenga:

- ✓ Código fuente de la solución a cada consigna obligatoria.
- ✓ Informe de diseño y ejecución según lo indicado en clase.

C. Consignas (de desarrollo obligatorio)

1. Proponga un diagrama de clases solución e implemente una clase Móvil Elemental así como el programa principal capaz de mover a ese móvil en el espacio, a partir de una secuencia de órdenes dadas por el operador.

Cada orden consta, básicamente, de un sentido de movimiento y una distancia en unidades de representación.

Este objeto es capaz de memorizar la secuencia de todas las órdenes dadas y la distancia total recorrida, entre otros datos (su identificación, la carga de su batería, el momento temporal en que empezó a moverse).

El programa principal debe emitir, luego de cada movimiento, un reporte que muestre la cadena de órdenes satisfechas, la distancia recorrida (medida en saltos) y el tiempo que el móvil estuvo en movimiento (desde su creación hasta que el operador eligió TERMINAR). Para ello aproveche el código mostrado en las diapositivas del archivo `poo_c++1`. Recuerde aplicar el principio de modularidad.

2. Proponga un diagrama de clases solución e implemente un programa capaz de trazar un segmento de recta a partir de conocer las coordenadas de 2 puntos en el plano. Se deben mostrar los datos de los puntos extremos. El programa debe permitir al usuario elegir el carácter que será usado en el trazado. Se satisfacen las siguientes relaciones:

a. $\forall x/x \in \mathbb{R} \wedge \{x \mid 0 \leq x < 74\} \wedge x \text{ es abcisa}$

b. $\forall y/y \in \mathbb{R} \wedge \{y \mid 0 \leq y < 22\} \wedge y \text{ es ordenada}$

c. $\forall P(x,y) / P(x,y) \in \text{Punto}$

d. $\forall S(P(x,y), P(x,y), \text{símboloElegido}) \in \text{Segmento}$

e. En caso que el operador omita ingresar el valor de la coordenada asumir valor 0. En caso que omita el carácter, asumir el signo + (Para cada caso use constructores diferentes).

f. Separe modularmente el código en la declaración de clase, declaración de funciones y principal.

3. Diseñe el diagrama de clases solución proponiendo como mínimo 3 entidades e implemente el programa correspondiente capaz de construir figuras elementales en un mapa de códigos de color para la representación gráfica en dispositivos de una resolución dada. Esto significa que el mapa de códigos debe tener la cantidad de Filas por Columnas igual a la resolución del dispositivo. Cada código está dado por un valor de 0 a 9, siendo 0 el equivalente a sin color (negro) y quedando el 9 reservado para casos de error en el dato. El código de color está formado por la letra # y el número del color.

Una de las clases del modelo corresponde a la de EntidadGeometrica. Las entidades geométricas que pueden trazarse son: segmentos de recta, triángulos, rectángulos y círculos. Las 3 primeras definidas por las coordenadas de sus vértices y la última por su centro y radio. El sistema coordenado del dispositivo se considera centrado (0,0) en el vértice inferior izquierdo.

Existe la restricción, que cada una de ellas posee un código de color diferente.

Los datos se proveen codificados (ver ejemplo) en un archivo según el formato:

Resolución, carácter (no alfanumérico) de relleno

Nombre de Entidad, #color1, param1, param2, param3, param4 [, param5...[, paramN]

Nombre de Entidad, #color2, param1, param2, param3, param4 [, param5...[, paramN]

.....

El programa debe:

- Permitir al usuario ingresar el nombre del archivo que contiene los parámetros correspondientes. Tener en cuenta que en el archivo existe una cantidad cualquiera de entidades pero como máximo se pueden representar las 8 primeras válidas.
- Mostrar la resolución, el carácter de relleno elegido y el listado, en formato tabular, de los valores leídos/adaptados o el mensaje de error que corresponda. En caso de éxito en la lectura del archivo, indicar la cantidad total de líneas de entidad leídas, la cantidad de entidades erróneas y la cantidad presentes de cada tipo.
- Generar un archivo de texto (formato ASCII) con el mapa del diseño resultante para la resolución indicada, conteniendo los códigos de color.
- En caso de parámetros dimensionales fuera del rango de la resolución, que no respeten la figura informada o estén omitidos, la figura se considera errónea y no se representa.
- En el caso que se omita el color utilizar el código de color 9.
- En el relleno usar el código de color 0, salvo que se indique un carácter de relleno personalizado, en cuyo caso se debe colocar en el archivo generado ese carácter en lugar del 0.
- Considere especialmente, el uso de constructores diferentes para manejar las situaciones de los ítems d, e y f.

Restricciones de Diseño: Las funciones deben ser miembros públicos mientras que los atributos deben ser privados. Separe el código en módulos, conforme a la organización básica de archivos de cabecera y de definición. Separe, además la capa de modelo respecto de la capa de vista de usuario.

Ejemplo: se muestra un ejemplo de reporte para el ítem b y un archivo de datos cualquiera dado.

```
600, 200, ?
s, #1, 1, 1, 150, 1
s, #2, 1, 1, 1, 500
t, #3, 5, 5, 50, 50, 95, 5
c, #4, 500, 100, 100
c, #5, 300, 100, 90
r, 10, 10, 10, 110, 260, 110, 260, 10
r, #6, 10, 10, 10, 110, 260, 110, , 10
r, #7, -10, 10, 10, 110, 260, 110, 2600, 10
y, #8, 5, 5, 50, 50, 95, 5
```

Archivo: tp3_figuras_1.txt

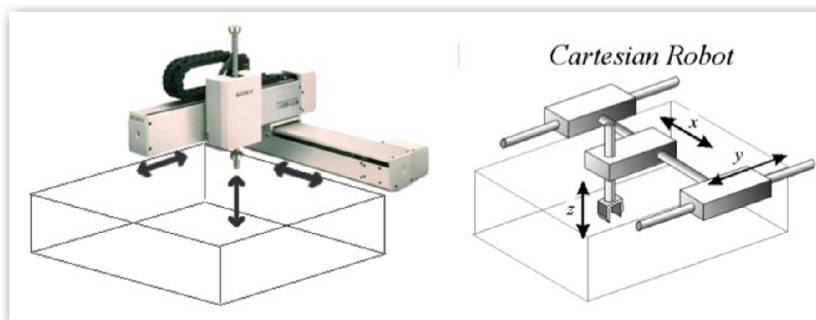
```
Resolucion: 600x200 puntos
Relleno personalizado ?
```

Figura	Color	vertice1/centro	vertice2/radio	vertice3	vertice4
Segmento	1	1, 1	150, 1		
Segmento	2	1, 1	1, 500		
Triangulo	3	5, 5	50, 50	95, 5	
Circulo	4	500, 100	100		
Circulo	5	300, 100	90		
Rectangulo	9	10, 10	10, 110	260, 110	260, 10
Error (parametros invalidos)					
Error (parametros invalidos)					
Error (figura inexistente)					

```
Líneas de entidad en el archivo: 9
Errores: 3
Segmentos: 2
Triangulos: 1
Rectangulos: 1
Circulos: 2
```

4. Escriba un programa capaz de simular de manera simplificada el trabajo de un robot cartesiano. Para ello defina:
- a) Una clase RobotCartesiano, capaz de registrar los puntos de trabajo deseados y de dar las órdenes de movimiento a cada articulación para resolver una trayectoria global. Es capaz de indicar al efector final el tiempo y el tipo de operación a realizar, en este caso: Tomar, Rotar y Soltar. Esta clase es la encargada de validar que en los datos no existan valores negativos, tampoco tiempos nulos u operaciones inválidas. De ser así informarlo y no permitir el arranque del dispositivo.
 - b) Una clase Articulacion, capaz de realizar su propio movimiento a partir de conocer la coordenada del punto a la cual debe desplazarse. Cada articulación conoce sus límites de trabajo, se maneja en su propio eje (sistema de coordenadas) y es el responsable de detectar/notificar los requerimientos fuera de su espacio de trabajo.
 - c) Una clase EfectorFinal, que puede encontrarse en 2 estados: inactivo o activo (según si se encuentra en camino a su coordenada de trabajo o ha alcanzado la misma). El efector final desconoce las coordenadas de su posición. Se asume por simplificación que, el efector puede moverse libremente en su espacio de trabajo. Es capaz de permanecer una cierta cantidad de ciclos realizando una tarea específica dada e informar cuando ésta ha terminado. Automáticamente se coloca en estado inactivo al finalizar la misma.

Restricciones de Diseño: Las funciones deben ser miembros públicos mientras que los atributos deben ser privados. Separe el código en módulos, conforme a la organización básica de archivos de cabecera y de definición. Separe, además la capa de modelo respecto de la capa de vista de usuario. Los tipos de operaciones a realizar deben estar definidos en una clase apropiada.



En la función principal (main) del programa debe resolverse lo siguiente:

- a) Ingresar los datos de N puntos en el espacio (el orden de los mismos determina la secuencia de trabajo), del tiempo de actividad del efector final (considerados equivalentes a la cantidad de ciclos de un contador) en cada punto y de la/s operación/es que debe realizar. Opcionalmente, los mismos pueden ser leídos desde un archivo.
- b) Proveer al operador las órdenes Inicio y Parada (equivalente a salir de la aplicación).
- c) Desde el momento en que arranca y hasta que se detiene al terminar la tarea, mostrar en pantalla, línea a línea, cada una de las posiciones alcanzadas por el efector final, así como su estado y la actividad que realiza.

Represente el diagrama de clases, el diagrama de actividad y el diagrama de estados de la solución propuesta para esta consigna.