

Programación Orientada a Objetos. Práctica 5

Implementa los siguientes ejercicios sueltos que se han comentado en clases de teoría:

1. Descargar, compilar, ejecutar y analizar el código fuente de los ejemplos de vector y lista de la STL que hay en la web de la asignatura.
2. Ir al enlace “Beginners guide to the std::sort() function” que hay en la web de la asignatura y entender como se ordena un vector de la STL, los #includes que hay que utilizar, la forma en que hay que invocar a la función sort(), etc.
3. A partir de uno de los ejemplos de vector anteriores hacer un programa que pida los elementos de un vector al usuario por teclado, los ordene y los muestre ordenados en pantalla.
4. Investiga como hacer el mismo programa del punto anterior pero en orden descendente. Realiza un programa que solicite al usuario el orden en el que quiere ordenar el vector: ascendente o descendente.
5. Hacer un programa que ordene un vector de Jugadores por el DNI y los muestre en pantalla ordenados.
6. Implementa en C++ un ejemplo usando la plantilla de función de los apuntes de C++ que se ha visto en clase de teoría. La función debe recibir un vector y su tamaño, y escribir su contenido en pantalla sea el vector de int, float o char.
7. Implementa en C++ un ejemplo usando la plantilla de clase de los apuntes de C++ que se ha visto en clase de teoría y ampliando con algunas funciones miembro más a tu gusto y criterio. La plantilla **MiClase** que realiza una división de sus datos privados sean int, float, etc.
8. Copia `datos.h` y `datos.cc` al directorio de la práctica 5. Añade a la clase *Dados* un insertador y un extractor propio. El insertador debe mostrar el valor de los dos dados en pantalla indicando con un texto el valor del dado1 y con otro texto el valor del dado2. El extractor debe pedir el valor para el dado 1 y para el dado 2 enviándolos como parámetros a la función `setDado1()` y `setDado2()`, e indicando si el valor es correcto o no, en cuyo caso lo volverá a pedir hasta que sea correcto.