



## EXAMEN PRÁCTICO DE PROGRAMACIÓN ORIENTADA A OBJETOS.

### 1. Codificar la clase Persona en los ficheros persona.h y persona.cc con las siguientes características.

- **Atributos:**
  - i. nombre\_ (string)
  - ii. direccion\_ (string)
  - iii. DNI\_ (string)
- **Constructor**
  - i. Versión 1: recibe como parámetros el nombre (obligatorio), la dirección (opcional) y el DNI (opcional). Los argumentos deberán tener un valor por defecto "XX".
  - ii. Versión 2: constructor de copia.
- **Observadores:**
  - i. getNombre.
  - ii. getDireccion.
  - iii. getDni.
- **Modificadores:**
  - i. setNombre.
  - ii. setDireccion.
  - iii. setDNI.

Todos ellos serán booleanos, comprobando que no se recibe una cadena vacía. En caso de recibirla, no se modificará la variable y retornarán false. En caso contrario, se modificará y retornará true.

- **Método getDatos():**

Dicho método retornará por referencia la concatenación del nombre, dirección y dni de la siguiente forma:

*"XXX con DNI YYY vive en CCC"*

donde XXX corresponde al nombre, YYY al DNI y CCC a la dirección.

### 2. Codificar la clase Jugador en los ficheros jugador.h y jugador.cc con las siguientes características.

- **La clase Jugador hereda de forma pública de la clase Persona y además dispone del siguiente atributo:**
  - puesto\_ (string). Puede tomar los valores de "base", "escolta", "alero", "ala-pivot" y "pivot".
- **Constructor:**
  - i. Versión 1: recibe como parámetros el nombre (obligatorio), la dirección (opcional), el DNI (opcional) y el puesto (obligatorio). Todos los argumentos deberán tener un valor por defecto.
  - ii. Versión 2: constructor de copia.



- **Observadores:**
  - i. getPuesto.
- **Modificadores:**
  - i. setPuesto.

Deberá comprobar que el valor es uno de los permitidos. En dicho caso, lo modificará y retornará true. En caso contrario, no lo modifica y retorna false.

- **Sobrecarga de los operadores:**
  - i. El operador =.
  - ii. El operador << escribirá en pantalla los valores de todos los atributos. Al ejecutar el test2 deberá salir por pantalla:

*Nombre 1;Dir 1;DNI 1;alero*

### 3. La clase Dado gestiona el lanzamiento de 1 Dado. Codifica la clase Dado en C++ con los siguientes métodos:

- **Constructor:** recibe 1 parámetro opcional con el valor inicial para el dado. Si no se recibe parámetro el valor inicial se asignará 1. Control de errores: Si el valor inicial no es correcto, el valor asignado será 1.
- **Observadores:**
  - i. get(). Devuelve el valor del dado (función inline).
- **Modificadores:**
  - i. set(). Recibe un parámetro con el valor entero a asignar al dato. Control de errores: Si se produce algún error en el valor asignado (debe estar entre 1 y 6), set() devuelve false y no modificado el dado, en caso contrario, set() modifica el dado y devuelve true.
- **Sobrecarga de los operadores:**
  - i. Sobrecarga del operador + para que se sume el valor de dos dados y devuelve un entero con el resultado.

### 4. Programa la clase Equipo la cual debe disponer de un vector de objetos Jugador, en los ficheros equipo.h y equipo.cc con los siguientes métodos:

- **Constructo vacío.**
- **Constructor de copia.**
- **Método size(),** que retorna el tamaño del vector de jugadores de la clase.
- **Método addJugador():** el cual insertar un jugador en el vector de jugadores siempre y cuando no se encuentre insertado el DNI de dicho jugador ya.



- **Operador []:** el cual recibe una posición del vector y retorna el jugador que se encuentre en dicha posición.
- **Método ordenar():** que ordena alfabéticamente el vector de jugadores según su nombre.

**5. Programa la clase Dato que posee un identificador de tipo int y un dato de tipo plantilla. Programe:**

- **Constructor:** con parámetro obligatorio id correspondiente al identificador y valor correspondiente al contenido de dato.
- **Observador por referencia:**
  - i. getId
  - ii. getDato
- **Modificador:**
  - i. setIdentificador
  - ii. setDato
- **Operador:**
  - i. **Comparacion ==** : comparar dos objetos de tipo Dato por el dato plantilla.
  - ii. **Comparación !=** : comparar dos objetos de tipo Dato por el dato plantilla.