

pooFebreroOrd23.pdf



Anónimo



Programación Orientada a Objetos



2º Grado en Ingeniería Informática



**Escuela Superior de Ingeniería
Universidad de Cádiz**

Máster

Online en Ciberseguridad

Nº1 en España según El Mundo



**Hasta el 46%
de beca**



Mejor Máster
según el
Ranking de
ELMUNDO

Para ser el mejor hay que aprender
de los mejores.

IMEF

Smart Education

Deloitte.

Infórmate

Consigue Empleo o Prácticas

Matrícúlate en IMF y accede sin coste a nuestro servicio de Desarrollo Profesional con más de 7.000 ofertas de empleo y prácticas al mes.



IMF
Smart Education

¿Quieres conocer todos los servicios?



Examen final Programación Orientada a Objetos

Convocatoria ordinaria de Febrero 2023

Enunciado aproximado (muy cercano al real)

WUOLAH

Ejercicio 1 - (3 puntos)**3**

- a) Completa la clase Vector con las definiciones de los métodos para que las siguientes instrucciones funcionen como se especifica. (0.5p)
- b) Implementa los métodos que has definido en el apartado a. (1p)
- c) ¿Es válido del comportamiento del constructor por movimiento por defecto? De ser afirmativo razónalo, si no, implementa el constructor por movimiento para que funcione correctamente. (0.5p)
- d) Implementa el método Vector::at para acceder elementos por su índice de tal forma que ejecute correctamente el siguiente fragmento de código y la salida por pantalla sea "Vector::at, índice fuera de rango". (1p)

Ejercicio 2 - (2 puntos)**4**

- a) Defina la clase de asociación que permita implementar dicha relación escribiendo exclusivamente las definiciones de los miembros imprescindibles para implementarla. (1p)
- b) ¿Es obligatorio usar una clase de asociación? Si es así explica razonadamente el porqué, si no, implementa como hacerlo escribiendo las declaraciones de los miembros esenciales. (1p)

Ejercicio 3 - (2.5 puntos)**5**

- a) Explica la relación que se puede establecer entre ListaOrdenada y la clase list. Implemente la clase ListaOrdenada. (1.5p)
- b) Añade el método size_t contar (const T& e) const que cuente el número de ocurrencias de un elemento dado. Para ello utilice count_if() de la clase STL que recibe dos iteradores y un predicado (clase objeto función que devuelve un booleano). Defina el predicado como una clase de objetos función o como una función anónima (función lambda) equivalente. (1p)

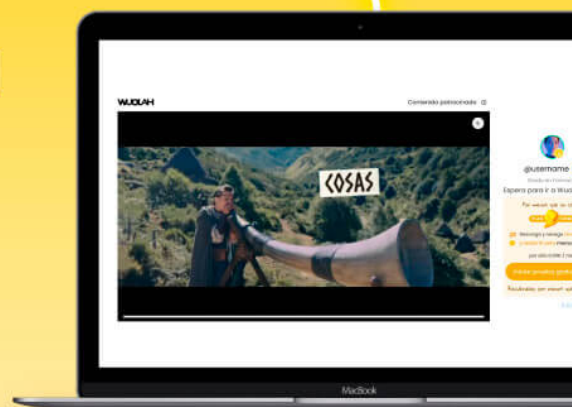
Ejercicio 4 - (2.5 puntos)**6**

- a) ¿Se puede mejorar la implementación de esta jerarquía de clases usando métodos polimórficos? En caso afirmativo reescribe el programa para obtener un resultado idéntico (1.5p)
- b) Implemente una función que clasifique un vector de punteros a instrumentos en otros 3 vectores de punteros a instrumentos, uno para cada clase, percusión, cuerda y viento ignorando aquellos que no pertenezcan a estas clases (1p)

WUOLAH

**¿Es pesado y te molesta
mientras estudias?
Nos deshacemos de
él por 1€.**

(De tu hermano no, del vídeo)
Hazte turbo.



PLAN TURBO

Descarga **rápido**, descarga **Turbo**



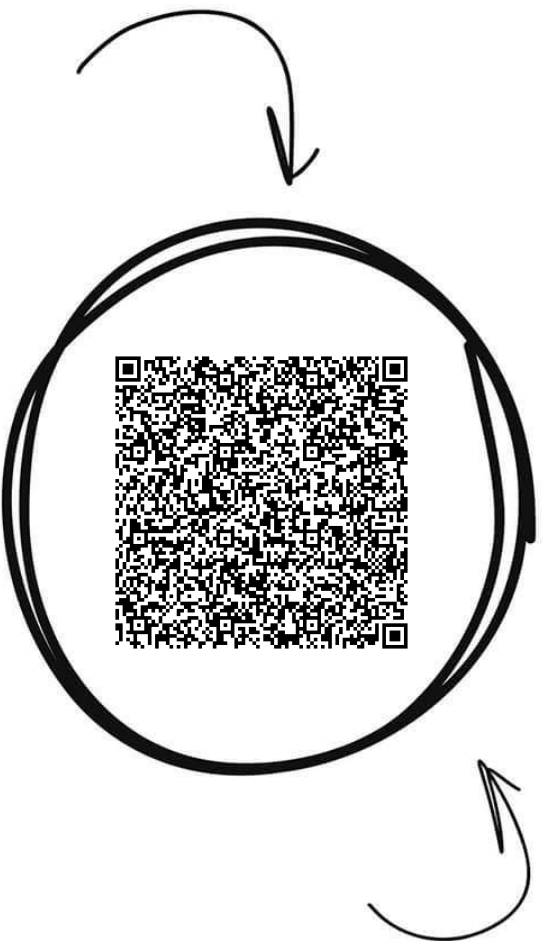
Aprovecha las ventajas del plan turbo y su rapidez, no esperes más, además te llevas 10 coins para descargas premium mensuales.

wuolah.com

Programación Orientada a Obj...



Comparte estos flyers en tu clase y **consigue más dinero y recompensas**



Banco de apuntes de la

WUOLAH

- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



Ejercicio 1 - (3 puntos)

```
1 class Vector
2 {
3     public:
4
5     (. . .)
6
7     private:
8         size_t n;
9         double* datos;
10 }
```

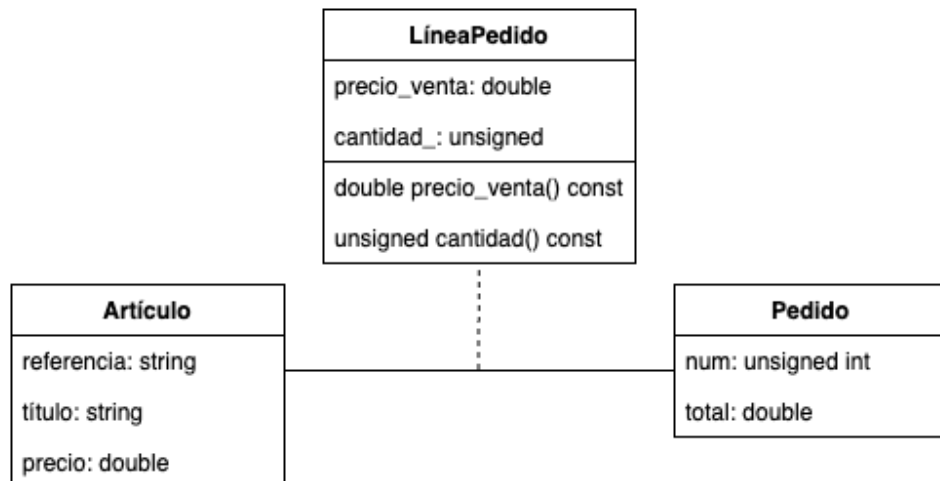
- a) Completa la clase Vector con las definiciones de los métodos para que las siguientes instrucciones funcionen como se especifica. **(0.5p)**
- Vector v1 (6) // v1 = {0, 0, 0, 0, 0, 0}
 - Vector v2 (4, 7.5) // v2 = {7.5, 7.5, 7.5, 7.5}
 - Vector v3 = {2.5, 1, 4.75, 3} // v3 = {2.5, 1, 4.75, 3}
 - Vector v4 // error de compilación
 - Vector v5 = 3 // error de compilación
- b) Implementa los métodos que has definido en el apartado a. **(1p)**
- c) ¿Es válido el comportamiento del constructor por movimiento por defecto? De ser afirmativo razónalo, si no, implementa el constructor por movimiento para que funcione correctamente. **(0.5p)**
- d) Implementa el método Vector::at para acceder elementos por su índice de tal forma que ejecute correctamente el siguiente fragmento de código y la salida por pantalla sea "Vector::at, índice fuera de rango". **(1p)**

```
1 int main()
2 {
3     const Vector v6 = {1, 2, 3, 4, 5, 6};
4     Vector v7 (6);
5
6     try{
7         v7.at(0) = v6.at(9);
8     } catch (std::out_of_range &e){
9         std::cerr << e.what() << std::endl;
10    }
11 }
```


¿Le odias tanto que no quieres volver a verle en tu vida? Le damos cita con San Pedro por 1€. (A tu ex no, al vídeo). Hazte Turbo.

PLAN TURBO
Descarga rápido, descarga Turbo

Ejercicio 2 - (2 puntos)



- a) Defina la clase de asociación que permita implementar dicha relación escribiendo exclusivamente las definiciones de los miembros imprescindibles para implementarla. **(1p)**
- b) ¿Es obligatorio usar una clase de asociación? Si es así explica razonadamente el porqué, si no, implementa como hacerlo escribiendo las declaraciones de los miembros esenciales. **(1p)**

Aprovecha las ventajas del plan turbo y su rapidez, no esperes más, además te llevas 10 coins para descargas premium mensuales.



Ejercicio 3 - (2.5 puntos)

ListaOrdenada usa la clase genérica *list* para implementarla.

```
1 template <typename T>
2 class ListaOrdenada
3 {
4     public:
5         typedef typename list<T>::const_iterator iterator;
6
7         void intentar (const T& e);
8         void eliminar (iterator p);
9
10        iterator begin() const;
11        iterator end() const;
12 }
```

- a) Explica la relación que se puede establecer entre ListaOrdenada y la clase *list*. Implemente la clase ListaOrdenada. **(1.5p)**
- b) Añade el método `size_t contar (const T& e) const` que cuente el número de ocurrencias de un elemento dado. Para ello utilice `count_if()` de la clase STL que recibe dos iteradores y un predicado (clase objeto función que devuelve un booleano). Defina el predicado como una clase de objetos función o como una función anónima (función lambda) equivalente. **(1p)**

Ejercicio 4 - (2.5 puntos)

```
1 #include <iostream>
2 #include <string>
3 #include <vector>
4
5 using namespace std;
6
7 class Instrumento
8 {
9     public:
10         enum tClase {instrumento, percusion, cuerda, viento};
11
12         Instrumento (string n): nom(n), clase_(instrumento){}
13
14         void tocar() const
15         {
16             cout << "Soy un " << nombre() << " y pertenezco a " << clase() << endl;
17         }
18
19         string nombre() const {return nom;}
20
21         string clase() const
22         {
23             switch (clase_)
24             {
25                 case percusion: return "percusión";
26                 case cuerda: return "cuerda";
27                 case viento: return "viento";
28                 default: return "instrumento";
29             }
30         }
31
32     protected:
33         string nom;
34         string clase_;
35 }
36
37 class Percusion: public Instrumento
38 {
39     public:
40         Percusion (string n): Instrumento(n) {clase_ = percusion}
41 }
42
43 class Cuerda: public Instrumento
44 {
45     public:
46         Cuerda (string n): Instrumento(n) {clase_ = cuerda}
47 }
48
49 class Viento: public Instrumento
50 {
51     public:
52         Viento (string n): Instrumento(n) {clase_ = viento}
53 }
54
55 Int main()
56 {
57     vector<Instrumento*> v(new Cuerda("Violin"), new Instrumento("Sintetizador"), new
Cuerda("Arpa"), new Percusion("Timbal"));
58
59     for (Instrumento*: v)
60     {
61         i -> tocar();
62         delete I;
63     }
64 }
```



WUOLAH

No si antes decirte
Lo mucho que te voy a recordar

Pero me voy a graduar.
Mañana mi diploma y título he de pagar

Llegó mi momento de despedirte
Tras años en los que has estado mi
lado.

Siempre me has ayudado
Cuando por exámenes me he
agobiado

Oh Wuolah wuolithah
Tu que eres tan bonita

- Febbrero Ord '23