

Laboratorio de PRO2. Ejercicio 'Gestión de una lavadora'.

Generado por Doxygen 1.9.4

Capítulo 1

Ejemplo de diseño modular: Gestión de una lavadora.

En este ejemplo se construye un programa modular que ofrece un menú de opciones para gestionar una lavadora. Se introducen las clases [Lavadora](#), [Cubeta](#) y [Prenda](#).

Sólo se documentan elementos públicos. En una próxima sesión se verá un ejemplo de proyecto completamente documentado, incluyendo los elementos privados.

Capítulo 2

Índice de clases

2.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

Cubeta	Representa una cubeta de ropa	??
Lavadora	Representa una lavadora	??
Prenda	Representa una prenda de ropa con atributos peso y color	??

Capítulo 3

Indice de archivos

3.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

Cubeta.hh	Especificación de la clase Cubeta	??
Lavadora.hh	Especificación de la clase Lavadora	??
Prenda.hh	Especificación de la clase Prenda	??
pro2_especif.cc	??

Capítulo 4

Documentación de las clases

4.1. Referencia de la Clase Cubeta

Representa una cubeta de ropa.

```
#include <Cubeta.hh>
```

Métodos públicos

- [Cubeta](#) ()
Creadora por defecto.
- [Cubeta](#) (const [Cubeta](#) &c)
Creadora copiadora.
- void [anadir_prenda](#) (const [Prenda](#) &p)
Añade una prenda a la cubeta.
- void [completar_lavadora](#) ([Lavadora](#) &lav)
Completa una lavadora con las prendas de la cubeta.
- void [escribir](#) () const
Operación de escritura.

4.1.1. Descripción detallada

Representa una cubeta de ropa.

Puede contener prendas blancas y de color. Puede usarse para intentar llenar una lavadora; en ese caso, las prendas se sacan de la cubeta en orden inverso al de entrada

4.1.2. Documentación del constructor y destructor

4.1.2.1. Cubeta() [1/2]

```
Cubeta::Cubeta ( )
```

Creadora por defecto.

Se ejecuta automáticamente al declarar una cubeta.

Precondición

cierto

Postcondición

El resultado es una cubeta sin prendas de ningún tipo

Coste

Constante

4.1.2.2. Cubeta() [2/2]

```
Cubeta::Cubeta (
    const Cubeta & c )
```

Creadora copiadora.

Permite declarar una cubeta nueva como copia de otra ya existente.

Precondición

cierto

Postcondición

El resultado es una cubeta igual que c

Coste

Lineal respecto al número de prendas de c

4.1.3. Documentación de las funciones miembro

4.1.3.1. anadir_prenda()

```
void Cubeta::anadir_prenda (
    const Prenda & p )
```

Añade una prenda a la cubeta.

Precondición

cierto

Postcondición

El parámetro implícito pasa a contener sus prendas originales más p

Coste

Constante

4.1.3.2. completar_lavadora()

```
void Cubeta::completar_lavadora (
    Lavadora & lav )
```

Completa una lavadora con las prendas de la cubeta.

Precondición

lav está inicializada

Postcondición

Se han eliminado del parámetro implícito y se han añadido a lav las prendas del parámetro implícito del color adecuado que más se acercan entre todas al peso máximo de lav sin pasarse, eligiéndose primero las que se introdujeron en último lugar

Coste

Lineal respecto al número de prendas del parámetro implícito

4.1.3.3. escribir()

```
void Cubeta::escribir ( ) const
```

Operación de escritura.

Precondición

cierto

Postcondición

Escribe el contenido del parámetro implícito por el canal estándar de salida

Coste

Lineal respecto al número de prendas del parámetro implícito

La documentación para esta clase fue generada a partir del siguiente fichero:

- [Cubeta.hh](#)

4.2. Referencia de la Clase Lavadora

Representa una lavadora.

```
#include <Lavadora.hh>
```

Métodos públicos

- [Lavadora](#) ()
Creadora por defecto.
- void [inicializar](#) (int pmax, bool col)
Inicializa la lavadora.
- void [anadir_prenda](#) (const [Prenda](#) &p)
Añade una prenda a la lavadora.
- void [lavado](#) ()
Realiza un lavado.
- bool [esta_inicializada](#) () const
Consultora del estado de la lavadora.
- bool [consultar_color](#) () const
Consultora del color de la lavadora.
- int [consultar_peso](#) () const
Consultora del peso actual de la lavadora.
- int [consultar_peso_maximo](#) () const
Consultora del peso máximo de la lavadora.
- void [escribir](#) () const
Operación de escritura.

4.2.1. Descripción detallada

Representa una lavadora.

Dispone de dos estados posibles (inicializada / no inicializada); si está inicializada tiene un peso máximo y un color y puede contener prendas de dicho color hasta alcanzar dicho peso máximo; si no está inicializada no contiene ninguna prenda y solo se puede inicializar

Todas las operaciones son de **coste constante** salvo las indicadas

4.2.2. Documentación del constructor y destructor

4.2.2.1. Lavadora()

```
Lavadora::Lavadora ( )
```

Creadora por defecto.

Se ejecuta automáticamente al declarar una lavadora.

Precondición

cierto

Postcondición

El resultado es una lavadora no inicializada

4.2.3. Documentación de las funciones miembro

4.2.3.1. anadir_prenda()

```
void Lavadora::anadir_prenda (
    const Prenda & p )
```

Añade una prenda a la lavadora.

Precondición

El parámetro implícito (L) está inicializado, color de p = color de L, peso de L

- peso de p <= peso máximo de L

Postcondición

El parámetro implícito contiene su carga original más p

4.2.3.2. consultar_color()

```
bool Lavadora::consultar_color ( ) const
```

Consultora del color de la lavadora.

Precondición

El parámetro implícito está inicializado

Postcondición

El resultado es el color del parámetro implícito

4.2.3.3. consultar_peso()

```
int Lavadora::consultar_peso ( ) const
```

Consultora del peso actual de la lavadora.

Precondición

El parámetro implícito está inicializado

Postcondición

El resultado es la suma de los pesos de las prendas del parámetro implícito

4.2.3.4. consultar_peso_maximo()

```
int Lavadora::consultar_peso_maximo ( ) const
```

Consultora del peso máximo de la lavadora.

Precondición

El parámetro implícito está inicializado

Postcondición

El resultado es el peso máximo del parámetro implícito

4.2.3.5. escribir()

```
void Lavadora::escribir ( ) const
```

Operación de escritura.

Precondición

El parámetro implícito está inicializado

Postcondición

Escribe las propiedades y el contenido del parámetro implícito por el canal estándar de salida

Coste

Lineal respecto al número de prendas del parámetro implícito

4.2.3.6. esta_inicializada()

```
bool Lavadora::esta_inicializada ( ) const
```

Consultora del estado de la lavadora.

Precondición

cierto

Postcondición

El resultado indica si el parámetro implícito está inicializado

4.2.3.7. inicializar()

```
void Lavadora::inicializar (
    int pmax,
    bool col )
```

Inicializa la lavadora.

Precondición

El parámetro implícito no está inicializado, $pmax > 0$

Postcondición

El parámetro implícito es una lavadora vacía inicializada con peso máximo "pmax" y color "col"

4.2.3.8. lavado()

```
void Lavadora::lavado ( )
```

Realiza un lavado.

Representa que se realiza el lavado, se retiran la prendas que contiene la lavadora y ésta queda en estado de volver a usarse

Precondición

El parámetro implícito está inicializado

Postcondición

El parámetro implícito no está inicializado

Coste

Lineal respecto al número de prendas del parámetro implícito

La documentación para esta clase fue generada a partir del siguiente fichero:

- [Lavadora.hh](#)

4.3. Referencia de la Clase Prenda

Representa una prenda de ropa con atributos peso y color.

```
#include <Prenda.hh>
```

Métodos públicos

- [Prenda \(\)](#)
Creadora por defecto. Se ejecuta automáticamente al declarar una prenda.
- [Prenda \(int pes, bool col\)](#)
Creadora con valores concretos.
- void [modificar](#) (int pes, bool col)
Modificadora de los atributos.
- int [consul_peso](#) () const
Consultora del peso.
- bool [consul_color](#) () const
Consultora del color.
- void [escribir](#) () const
Operación de escritura.

4.3.1. Descripción detallada

Representa una prenda de ropa con atributos peso y color.

Todas las operaciones son de **coste constante**

4.3.2. Documentación del constructor y destructor

4.3.2.1. Prenda() [1/2]

```
Prenda::Prenda ( )
```

Creadora por defecto. Se ejecuta automáticamente al declarar una prenda.

Precondición

cierto

Postcondición

El resultado es una prenda de peso 0 y color blanco

4.3.2.2. Prenda() [2/2]

```
Prenda::Prenda (
    int pes,
    bool col )
```

Creadora con valores concretos.

Precondición

$pes > 0$

Postcondición

El resultado es una prenda con peso "pes" y color "col"

4.3.3. Documentación de las funciones miembro

4.3.3.1. `consul_color()`

```
bool Prenda::consul_color ( ) const
```

Consultora del color.

Precondición

cierto

Postcondición

El resultado es el color del parámetro implícito

4.3.3.2. `consul_peso()`

```
int Prenda::consul_peso ( ) const
```

Consultora del peso.

Precondición

cierto

Postcondición

El resultado es el peso del parámetro implícito

4.3.3.3. `escribir()`

```
void Prenda::escribir ( ) const
```

Operación de escritura.

Precondición

cierto

Postcondición

Se han escrito los atributos del parámetro implícito en el canal standard de salida.

4.3.3.4. `modificar()`

```
void Prenda::modificar (
    int pes,
    bool col )
```

Modificadora de los atributos.

Precondición

`pes > 0`

Postcondición

El parámetro implícito pasa a tener peso "pes" y color "col"

La documentación para esta clase fue generada a partir del siguiente fichero:

- [Prenda.hh](#)

Capítulo 5

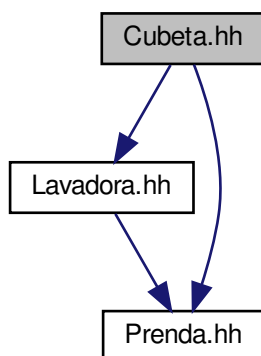
Documentación de archivos

5.1. Referencia del Archivo Cubeta.hh

Especificación de la clase [Cubeta](#).

```
#include "Lavadora.hh"  
#include "Prenda.hh"
```

Dependencia gráfica adjunta para Cubeta.hh:



Clases

- class [Cubeta](#)

Representa una cubeta de ropa.

5.1.1. Descripción detallada

Especificación de la clase [Cubeta](#).

5.2. Cubeta.hh

[Ir a la documentación de este archivo.](#)

```

1
5 #ifndef _CUBETA_HH_
6 #define _CUBETA_HH_
7
8 #include "Lavadora.hh"
9 #include "Prenda.hh" // redundante, pero necesario para el diagrama modular
10
11 #ifndef NO_DIAGRAM // ver explicacion en Prenda.hh
12 #include <stack>
13 #endif
14
21 class Cubeta {
22 public:
23     // Constructoras
24
32     Cubeta();
33
41     Cubeta(const Cubeta& c);
42
43     // Modificadoras
44
50     void anadir_prenda(const Prenda& p);
51
59     void completar_lavadora(Lavadora& lav);
60
61     // Escritura de una cubeta
62
69     void escribir() const;
70
71     // en este ejemplo no documentamos los elementos privados, lo haremos en el
72     // ejemplo de los experimentos inmunológicos (habrá que cambiar también el
73     // Doxyfile)
74 private:
75     stack<Prenda> ropacolor;
76     stack<Prenda> ropablanca;
77     static void completar_lavadora_pila_rec(stack<Prenda>& p, Lavadora& l);
78     static void completar_lavadora_pila_it(stack<Prenda>& p, Lavadora& l);
79     static void escribir_pila_prenda(const stack<Prenda>& p);
80 };
81 #endif

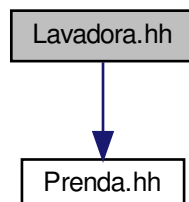
```

5.3. Referencia del Archivo Lavadora.hh

Especificación de la clase [Lavadora](#).

```
#include "Prenda.hh"
```

Dependencia gráfica adjunta para Lavadora.hh:



Clases

- class [Lavadora](#)

Representa una lavadora.

5.3.1. Descripción detallada

Especificación de la clase [Lavadora](#).

5.4. Lavadora.hh

[Ir a la documentación de este archivo.](#)

```
1
2
3
4
5 #ifndef _LAVADORA_HH_
6 #define _LAVADORA_HH_
7
8 #include "Prenda.hh"
9
10 #ifndef NO_DIAGRAM // ver explicacion en Prenda.hh
11 #include <list>
12 #endif
13
14
15
16
17
18
19
20
21
22
23
24 class Lavadora {
25 public:
26     // Constructoras
27
28     Lavadora();
29
30     // Modificadoras
31
32     void inicializar(int pmax, bool col);
33
34     void anadir_prenda(const Prenda& p);
35
36     void lavado();
37
38     // Consultoras
39
40     bool esta_inicializada() const;
41
42     bool consultar_color() const;
43
44     int consultar_peso() const;
45
46     int consultar_peso_maximo() const;
47
48     // Escritura de la lavadora
49
50     void escribir() const;
51
52 private:
53     list<Prenda> prendas;
54     bool col;
55     int pes;
56     bool ini;
57     int pesmax;
58 };
59 #endif
```

5.5. Referencia del Archivo Prenda.hh

Especificación de la clase [Prenda](#).

Clases

- class [Prenda](#)
Representa una prenda de ropa con atributos peso y color.

5.5.1. Descripción detallada

Especificación de la clase [Prenda](#).

5.6. Prenda.hh

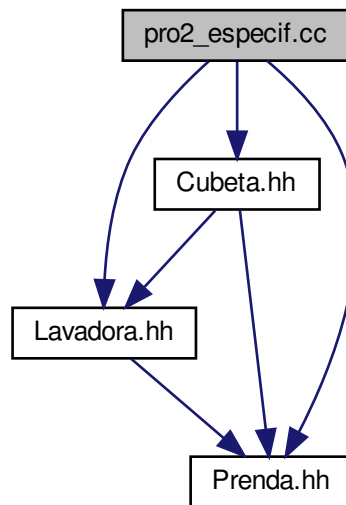
[Ir a la documentación de este archivo.](#)

```
1
2
3
4
5
6 #ifndef _PRENDA_HH_
7 #define _PRENDA_HH_
8
9 #ifndef NO_DIAGRAM          // esto hace que el Doxyfile de la sesión no
10 #include <iostream          // mientras que el compilador de c++
11                             // incluya estas clases en los diagramas modulares,
12 #endif                      // si que las procesa correctamente
13
14
15
16
17
18
19
20
21 class Prenda {
22 public:
23     // Constructoras
24
25
26
27
28     Prenda();
29
30     Prenda(int pes, bool col);
31
32
33     // Modificadoras
34
35
36     void modificar(int pes, bool col);
37
38
39     // Consultoras
40
41
42     int consul_peso() const;
43
44
45     bool consul_color() const;
46
47
48     // Escritura de prenda
49
50     void escribir() const;
51
52
53 private:
54     // incluimos los campos de la clase para que se vea que los elementos privados
55     // no aparecen en la documentación si está así configurado el Doxyfile
56     int peso;
57     bool color;
58 };
59
60 #endif
```

5.7. Referencia del Archivo pro2_especif.cc

```
#include "Cubeta.hh"
#include "Lavadora.hh"
#include "Prenda.hh"
```

Dependencia gráfica adjunta para pro2_especif.cc:



Funciones

- `int main ()`

Programa principal para el ejercicio Gestión de una lavadora.

5.7.1. Documentación de las funciones

5.7.1.1. main()

```
int main ( )
```

Programa principal para el ejercicio *Gestión de una lavadora*.

```
31 {  
32     Lavadora l;  
33     Cubeta c;  
34     int peso;  
35     string color;  
36  
37     string ins;  
38     while (cin > ins and ins != "-8") {
```

```
39 if (ins == "-1") {
```

```
40 cin > peso color;
```

```
41 bool b;
```

```
42 if (color == "true") b = true;
43 else b = false;
44 l.inicializar(peso, b);
45 } else if (ins == "-2") {
46     cin >> peso >> color;
47     if (l.esta_inicializada()) {
48         bool b;
49         if (color == "true") b = true;
50         else b = false;
51         Prenda p(peso, b);
52         l.anadir_prenda(p);
53     }
54 } else if (ins == "-3") {
55     cin >> peso >> color;
56     bool b;
57     if (color == "true") b = true;
58     else b = false;
59     Prenda p(peso, b);
60     c.anadir_prenda(p);
61 } else if (ins == "-4") {
62     if (l.esta_inicializada())
63         c.completar_lavadora(l);
64 } else if (ins == "-5")
65     l.lavado();
66 else if (ins == "-6") {
67     cout << " Cubeta: " << endl;
68     c.escribir();
69 } else {
70     cout << " Lavadora: " << endl;
71     l.escribir();
72 }
73 }
74 }
```