Sesión 7

Diseño modular en C++ (II)

En esta sesión traducimos a C++ el problema de diseño modular *"Gestión de una lavadora"*. Los ficheros necesarios de las clases `Prenda`, `Cubeta` y `Lavadora` están en `$INCLUDES_CPP` y `$OBJETOS_CPP`.

7.1 Ejercicio: programa principal

Implementad el programa principal empleando todas las clases mencionadas. Introducid dos opciones más en el menú: `escribir_cubeta = -6` y `escribir_lavadora = -7`; el final queda desplazado al -8.

Probadlo con las entradas que queráis, incluyendo el fichero `datos.txt`. Los resultados correctos para éste se encuentran en `salida.txt`. Al implementar la operación `escribir_cubeta` deberéis imitar el formato de dicho fichero.

7.2 Ejercicio: juegos de pruebas especiales

Escribid y probad una serie de ficheros de datos que exploren diversas situaciones límite de la operación `completar_lavadora`, por ejemplo:

- que no se pueda sacar ninguna prenda de la cubeta, aunque en ella haya prendas del color correspondiente, porque la primera prenda posible hace que se pase del peso máximo (la lavadora no se modifica)
- que se saquen de la cubeta todas las prendas del color correspondiente
- que la lavadora quede llena (se alcance el peso maximo exacto)
- que la primera prenda que no se pueda sacar de la cubeta hagaa que se alcance el peso máximo más 1.

7.3 Ejercicio: Módulo `Cubeta`

Implementad vuestra propia versión de la clase `Cubeta` de modo que el programa principal siga funcionando si en vez de linkarlo con `Cubeta.o` del `$OBJETOS_CPP` lo linkáis con vuestra versión. Usad el diseño que aparece en los apuntes y, en particular, codificad las dos versiones de la operación `completar_lavadora`. Prestad especial atención a las auxiliares de éstas.

Realizad las pruebas con los mismos juegos empleados para probar el principal y después con otros nuevos, si detectáis situaciones que no hayan sido probadas.

7.4 Diseño modular

Recordemos el programa principal y las especificaciones de los diferentes módulos. En los ficheros `*.doc` están la cabeceras de las clases correspondientes. Observad que, como siempre, se ha de prestar atención a las posibles modificaciones que las cabeceras hayan sufrido al adaptarlas a C++ y a las operaciones nuevas (`escribir_cubeta` y `escribir_lavadora`).

```
Programa principal

  Sobre Cubeta, Lavadora, Prenda;

  var  c: Cubeta; p: Prenda; l: Lavadora;
       op: ent; pes: nat; col: bool;

  c:=instalar_cubeta();
  l:=instalar_lavadora();
  op:=leer();
  mientras op!=-6 hacer
    si op=-1 --> pes:=leer(); col:=leer();
                 inicializar_lavadora(l, col, pes);
    [] op=-2 --> pes:=leer(); col:=leer();
                 p:=crear_prenda(pes, col);
                 añadir_prenda_lavadora(l, p);
    [] op=-3 --> pes:=leer(); col:=leer();
                 p:=crear_prenda(pes, col);
                 añadir_prenda_cubeta(c, p);
    [] op=-4 --> completar_lavadora(c, l);
    [] op=-5 --> lavado(l);
    fsi;
    op:=leer();
  fmientras
```

Notad que estamos suponiendo que los datos de la entrada son correctos. Si no, habría que aplicar las protecciones correspondientes a las operaciones antes de utilizarlas, para garantizar que se cumplen sus precondiciones.

```
Módulo Cubeta {Especificación}

  Sobre Prenda, Lavadora

  Tipo Cubeta

  función instalar_cubeta () dev c: Cubeta
    /* cierto */
    /* devuelve una cubeta vacía */

  acción añadir_prenda_cubeta (e/s c: Cubeta; e p: Prenda)
    /* cierto */
    /* se ha añadido p  encima del todo de c*/

  accion completar_lavadora (e/s c: Cubeta; e/s l: Lavadora)
    /* l esta inicializada */
    /* se han retirado de c y se han añadido a l las prendas de c
       del color adecuado que más se acercan entre todas al peso
       maximo de l sin pasarse, eligiéndose primero las que se
       introdujeron en último lugar */


Módulo Lavadora {Especificación}

  Sobre Prenda

  Tipo Lavadora

  funcion instalar_lavadora () dev l: Lavadora
    /* cierto */
    /* l es una lavadora sin inicializar}

  accion inicializar_lavadora (e/s l: Lavadora, e col: bool, e p: nat)
    /* l no esta inicializada */
    /* l es una lavadora vacía, inicializada para cargar ropa de
       color col, sin pasarse de un peso p */

  accion añadir_prenda_lavadora (e/s l: Lavadora; e p: Prenda)
    /* l esta inicializada, peso de l + peso de p <= peso maximo de l;
       color de l = color de p */
    /* l contiene su carga inicial más p */

  accion lavado (e/s l: Lavadora)
    /* l esta inicializada */
    /* l no esta inicializada */
```

```
  funcion esta_inicializada (l: Lavadora) dev b: bool
    /* cierto */
    /* b indica si l esta inicializada */

  funcion consultar_color_lavadora (l: Lavadora) dev b: bool
    /* cierto */
    /* "col" es el color de l */

  funcion consultar_peso_lavadora (l: Lavadora) dev pes: nat
    /* cierto */
    /* "pes" es el peso de l */

  funcion consultar_peso_maximo (l: Lavadora) dev pes: nat
    /* cierto */
    /* "pes" es el peso maximo de l */


Módulo Prenda {Especificación}

  tipo Prenda

  función crear_prenda (pes: nat; col: booleano) dev p: Prenda
    /* cierto */
    /* devuelve una prenda de peso "pes" y color "col" */

  función consul_peso_prenda (p: Prenda) dev pes: nat
    /* cierto */
    /* "pes" es el peso de la prenda p */

  función consul_color_prenda (p: Prenda) dev col: bool
    /* cierto */
    /* "col" es el color de la prenda p */
```

7.5 Un ejercicio alternativo

Modificad la operación `completar_lavadora` para que no pare de pasar ropa de la cubeta a la lavadora cuando encuentre una prenda que no quepa en ésta, sino que siga probando las prendas situadas por debajo de dicha prenda. pasando a la lavadora cada prenda encontrada que quepa en ella. Las prendas que queden en la cubeta han de mantener el orden en el que estaban.

Ejemplo: si en la lavadora faltan por llenar 7 unidades de peso de ropa blanca y en la cubeta tenemos las siguientes prendas blancas

```
1
2
```

```
2
4
1
3
2
```

al terminar han de quedar en la cubeta sólo las prendas de pesos

```
4
3
2
```

Diseñad una nueva serie de juegos de pruebas para esta versión alternativa, similar a la anterior.