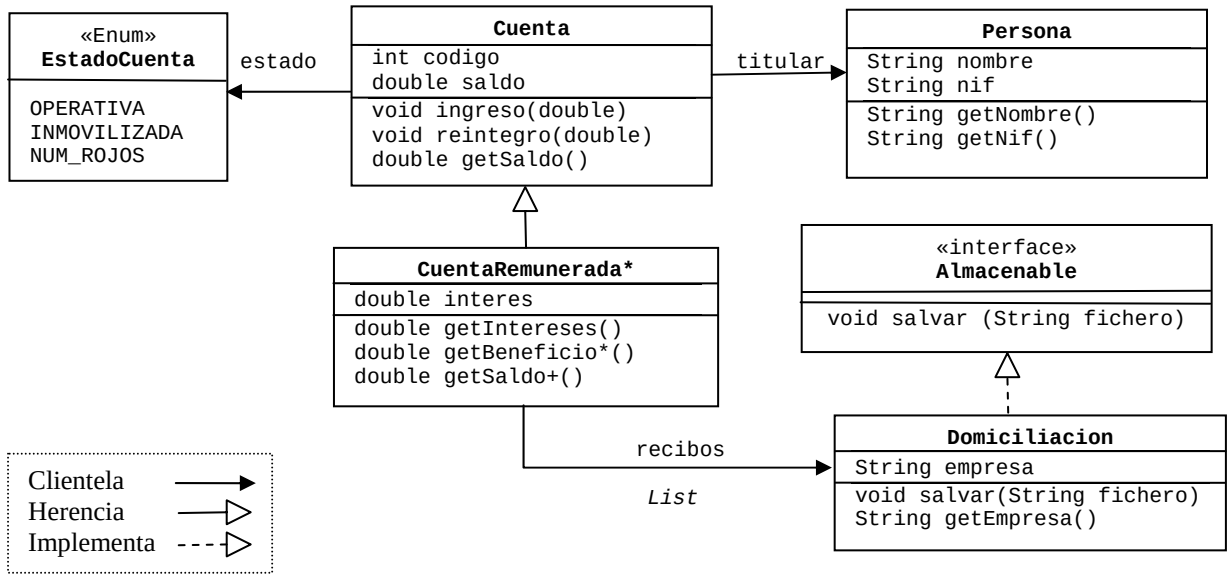
Examen de la convocatoria de Junio de 2009

Bloque II. Problemas — 5 puntos

Se debe tener un mínimo del 40% de la puntuación de este bloque para poder aprobar el examen.

En este apartado se proponen ejercicios de programación que deben ser resueltos utilizando el lenguaje de programación **Java**. Se valorará que los ejercicios cumplan la funcionalidad requerida y que el código cumpla los **criterios de calidad** expresados en el tema 1 de la asignatura: extensibilidad, reutilización, principios de diseño modular, etc.

Para los ejercicios que solicitan expresar gráficamente el código de la aplicación debe utilizarse la siguiente **notación**. Las clases y métodos abstractos se marcarán con *. Un método redefinido se marcará con +.



PROGRAMACIÓN ORIENTADA A OBJETOS **1/7/2009**

1. En una aplicación dedicada a la gestión del almacenamiento en una empresa aparecen dos conceptos: *Contenedor* y *Mercancía*. Los contenedores son recipientes de gran tamaño que son utilizados para el almacenamiento de mercancías. Un contenedor se caracteriza por su capacidad (volumen total del contenedor, valor entero), las mercancías que contiene, número de mercancías y si está abierto o cerrado para recibir mercancías. Por su parte, una mercancía se caracteriza por ocupar un volumen y puede estar contenida en un contenedor.

   a) (***0,5 puntos***) Identifica las propiedades de los objetos de la clase `Contenedor`. Para cada propiedad indica el atributo que almacena su información o si es una propiedad calculada el cálculo necesario para implementarla. Asimismo, hay que indicar qué operaciones de acceso/modificación, justificando cada una de ellas, deben definirse para cada propiedad. Por último, escribe el código de un único constructor para la clase.

   b) (***0,75 puntos***) Entre las clases `Contenedor` y `Mercancia` existe una relación de clientela bidireccional. Un contenedor puede contener varias mercancías y una mercancía puede estar contenida como máximo en un contenedor. La relación de clientela debe ser consistente entre ambos extremos. Es decir, si se establece el contenedor de una mercancía, entonces la mercancía formará parte del contenido de dicho contenedor y al revés. Teniendo en cuenta que se tiene que mantener la relación de clientela bidireccional entre las dos clases y aplicando la técnica del diseño por contrato implementa: b.1) el método `addMercancia` en la clase `Contenedor` sabiendo que un contenedor podrá alojar mercancías siempre que esté abierto y que el volumen de la nueva mercancía y el del resto de mercancías que contiene no sobrepase la capacidad del contenedor, y b.2) la clase `Mercancia` que cumpla las restricciones de este apartado y las descritas en el preámbulo del ejercicio.

   c) (***0,5 puntos***) La clase `Caja` que es un tipo de `Mercancia` que se caracteriza por tener las propiedades de ancho, alto y fondo. El volumen se calcula como el *ancho x alto x fondo*. Un `Cubo` es un tipo de `Caja` que tiene el mismo valor para alto, ancho y fondo. Escribe el código de las clases `Caja` y `Cubo`.

   d) (***0,25 puntos***) Supuesta la implementación de la clase `Almacen`, implementa, aplicando la técnica del Diseño por Contrato, el método `cargar` encargado de intentar añadir una lista de mercancías a un contenedor. En el caso de que no lo consiga devuelve `false`. La signatura del método sería:

```
public static boolean cargar (Contenedor contenedor,
                              List<Mercancia> mercancias){…}
```

   e) (***0,5 puntos***) ¿Qué cambios habría que hacer a la clase `Contenedor` para que sea genérica? ¿Qué ventajas tendría esta implementación? Justifica las respuestas.

2. Una caseta meteorológica está equipada con distintos aparatos de observación: termómetro (para medir la temperatura), barómetro (para medir la presión atmosférica), etc. A lo largo del día se hace la observación, esto es, se leen los valores de cada uno de los aparatos para su registro. La observación de cada uno de los aparatos está formada por un identificador del aparato, el valor numérico de la medida y la unidad de dicha medida. Por ejemplo, "T: 34,6 ºC" en el caso del termómetro o "H: 7,4 Mb" en el caso del barómetro. La caseta debe ofrecer un método para devolver una cadena con la lectura de las observaciones de los aparatos.

   a) (***0,5 puntos***) Representa gráficamente las clases implicadas en el problema incluyendo como aparatos el termómetro y el barómetro. Implementa el método que devuelva la observación de los aparatos aplicando el patrón de diseño del *Método Plantilla*.

   b) (***1 punto***) El método `getObservaciones` de la clase que representa la caseta meteorológica (`CasetaMeteo`) devuelve una cadena de texto que incluye en primer lugar la fecha en la que se realiza la llamada seguida de las observaciones de los aparatos. Por ejemplo, "01/07/2009 T: 34,6 ºC – H: 7,4 Mb". El formato de las fechas varía en función de la caseta. Por ejemplo, entre otros, una caseta podría tomar el *formato abreviado* "01/07/2009" mientras que otra caseta podría tomar el *formato inglés* "07/01/2009". Al instalarse una caseta meteorológica se establece el formato de las fechas para las observaciones. Implementa la clase `CasetaMeteo` y todos los tipos que sean necesarios para conseguir la funcionalidad propuesta. Escribe el código que construya una caseta con el formato de fecha abreviado.

   Nota: el constructor de la clase `java.util.Date` construye un objeto con la fecha actual. La clase dispone de los métodos `int getYear()`, `int getMonth()`, `int getDay()`, que devuelven el año, mes y día del objeto fecha, respectivamente.

   c) (***0,5 puntos***) Dado que los aparatos se encuentran instalados de forma externa, puede que al solicitar la observación del aparato exista algún error en la transmisión de los datos. Este error será notificado mediante la excepción `FalloObservacionException`. En el caso de que se produzca un fallo en la observación se intentará la lectura una segunda vez. Si esta segunda vez tampoco es posible obtener la medida del aparato, se anotará en la lectura la cadena `"--"` para indicar que no se tomó la medida de ese aparato. ¿De qué tipo sería la excepción `FalloObservacionException`? Justifica la respuesta. Modifica el método `getObservaciones` de la clase `CasetaMeteo` para gestionar los fallos y reintentos descritos.

   d) (***0,5 puntos***) Se puede considerar que existe probabilidad de lluvia cuando se produce un descenso significativo de la presión atmosférica. Los barómetros, que son los encargados de medir la presión, podrían determinar si la presión actual ha descendido más de un determinado valor umbral desde la última observación. Esta situación puede consultarse mediante el método `existeProbabilidadLluvia`. Modifica el método `getObservaciones` de la clase `CasetaMeteo` de manera que se aproveche el momento en el que se obtienen las observaciones de los aparatos para que en la caseta se registre si existe probabilidad de lluvia. Supondremos que hay probabilidad de lluvia si alguno de los barómetros indica esta situación.