

Examen de la convocatoria de Febrero de 2009

Bloque II. Problemas

5 puntos

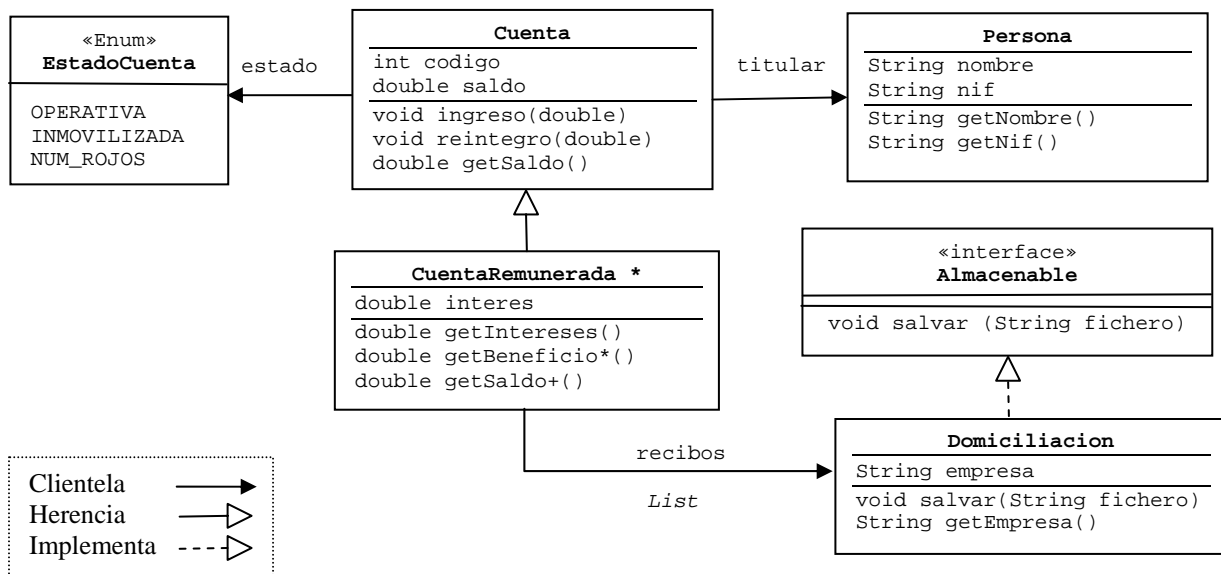
Nombre: \_\_\_\_\_

DNI: \_\_\_\_\_ Titulación: \_\_\_\_\_

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 +.



1- Cada *titulación* ofertada por la universidad se caracteriza por un nombre y una lista de los alumnos que se han preinscrito en dicha titulación después de la selectividad. Cada *alumno* se caracteriza por su nombre, dni y la nota obtenida en selectividad. Además, puede haber alumnos preinscritos que procedan de un distrito diferente al regional y, para este tipo de alumnos, se debe establecer la comunidad autónoma de la que procede y el año en el que aprobaron la selectividad.

En la universidad se generan decenas de listados diferentes. Para imprimir el listado de los alumnos preinscritos en una titulación se debe tener en cuenta a quien va dirigido dicho listado. Por ejemplo:

- Si el listado se va publicar en el **tablón de alumnos**: se debe imprimir el dni y la nota del alumno.

- Si el listado se va a archivar en la **secretaría** del centro: se debe imprimir el dni, el nombre del alumno, comunidad autónoma (sólo para los alumnos de otra región) y la nota.

a) (1 pto) Representa gráficamente los tipos que implementan la funcionalidad de la aplicación siguiendo la notación propuesta. Implementa el método `generarListado` en la clase `Titulacion` que devuelva la cadena de texto a imprimir correspondiente al listado de los alumnos preinscritos según el tipo de listado (tablón de alumnos, secretaría, etc.). Por último, implementa todo el código necesario para generar el listado para el tablón de alumnos.

b) (0'5 ptos) Implementa un método en la clase `Titulacion` que calcule la nota media de los alumnos preinscritos que procedan de una comunidad autónoma dada. Por ejemplo, podría interesar saber la nota media de selectividad de los alumnos que proceden de Andalucía.

2- (1 pto) El puerto de Cartagena debe coordinar las actividades de carga de los barcos con fruta y verdura traída por camiones desde el campo. Los barcos que esperan ser cargados se ubican en una *cola de espera* que los ordena teniendo en cuenta su capacidad (los barcos con mayor capacidad serán atendidos en primer lugar) y los camiones que esperan ser descargados se ubican también en una *cola de espera* que los ordena teniendo en cuenta los minutos de espera (los camiones que llevan más tiempo esperando serán atendidos en primer lugar).

Implementa la clase genérica `ColaEspera`, haciendo uso de la clase `LinkedList` de Java, que sirva para almacenar los barcos y camiones del problema. La clase `ColaEspera` debe disponer de los métodos para: (1) insertar un elemento en la cola, (2) extraer un elemento de la cola (la *precondición* para extraer un elemento es que la cola no esté vacía) y (3) consultar el número de elementos de la cola. También se debe especificar cualquier implementación que deba hacerse en las clases que representen a los barcos y camiones para dar soporte a la solución propuesta.

3- Un laboratorio farmacéutico se encarga de la elaboración de medicamentos. Los componentes de un medicamento pueden ser principios activos (Vitamina C, Codeína, Paracetamol, etc.) y otros medicamentos. Cada componente se caracteriza por tener un nombre y unas acciones terapéuticas. A su vez, las acciones terapéuticas se caracterizan por tener un nombre (Cefalea, Gripe, etc.) y una descripción detallada sobre el efecto que producen. Los componentes mantienen un conjunto ordenado por nombre de las acciones terapéuticas. En el caso de los medicamentos, las acciones terapéuticas son la unión (sin repetidos) de las acciones terapéuticas de todos sus componentes.

Para elaborar un medicamento se irán añadiendo uno a uno los componentes que lo forman indicando el porcentaje de dicho componente en el medicamento, siempre que el porcentaje que se vaya a añadir no exceda el 100% de la composición. De esta forma, podríamos elaborar los siguientes medicamentos:

- Medicamento A cuya composición es 20% de Codeína, 80% de Paracetamol
- Medicamento B cuya composición es 50% de Vitamina C, 50% de Medicamento A

En el proceso de fabricación de un medicamento hay que tener en cuenta que puede encontrarse en dos estados: *incompleto*, si no ha alcanzado el 100% de la composición, y *completo*, si ya se ha alcanzado el 100% de la composición. Cuando un medicamento está completo no pueden añadirse más componentes.

**Nota:** Para la implementación de las clases propuestas deben utilizarse las colecciones de Java.

- a) (0,5 ptos) Representa gráficamente los tipos que implementan la funcionalidad de la aplicación siguiendo la notación propuesta.
- b) (1 pto) Implementa la clase `Medicamento` siguiendo la técnica del Diseño por Contrato. Justifica todas las decisiones de implementación.
- c) (0'5 ptos) Implementa la clase `AccionTerapeutica`. Justifica todas las decisiones de implementación.
- d) (0'5 ptos) La especificación de cualquier componente del laboratorio farmacéutico incluye, en este orden, el nombre, la composición y las acciones terapéuticas. En el caso de los principios activos la composición sería la cadena "100%" y el nombre, por ejemplo, "100% paracetamol". Implementa el método `getEspecificacion`, aplicable sobre cualquier componente del laboratorio, que devuelva una cadena de texto con la especificación.