


<b>Programación – Examen Final Enero</b> <b>20 de Enero de 2017</b>	 Universidad Carlos III de Madrid
Grado en Ingeniería Informática	

LEA **CUIDADOSAMENTE** ESTAS INSTRUCCIONES ANTES DE COMENZAR EL EXAMEN:

- No responda en esta hoja, utilice las que se proporcionan.
- Rellene todas las páginas con un **bolígrafo** (datos personales y respuestas). No utilice lápiz o bolígrafo rojo.
- No olvide incluir su nombre y apellido en cada página.
- Pregunte al profesor antes de usar cualquier biblioteca de Java. Como regla general, las bibliotecas no vistas durante el curso están **prohibidas**.
- Se permiten notas, diapositivas, libros, etc. No se permiten dispositivos electrónicos.
- La duración de este examen es de **4 horas**.

**Problema 1 (10 Puntos).**- Los trasteros de autoservicio son pequeños almacenes que se pueden alquilar para almacenar artículos. Un trastero tiene un nombre y contiene entre 10 y 50 unidades de almacenamiento (siempre en múltiplos de 10). Las unidades de almacenamiento pueden estar alquiladas o no y el alquiler puede estar activo o haber expirado (el arrendatario no paga el alquiler). Una unidad de almacenamiento contiene artículos (sólo si está alquilada), que tienen un nombre y un valor monetario. Para los propósitos de este problema, asumiremos que dentro de una unidad de almacenamiento sólo se pueden encontrar los siguientes tipos de artículos:

- Muebles: su valor oscila entre 10€ y 100€.
- Ropa: su valor oscila entre 1€ y 15€.
- Electrodomésticos: su valor oscila entre 5€ y 60€.
- Antigüedades: su valor oscila entre 50€ y 500€.

Una unidad de almacenamiento puede tener un máximo de 10 artículos, que serán combinaciones aleatorias de los anteriores, cada uno con un valor diferente (por ejemplo, podría contener muebles valorados en 30€, ropa valorada en 2€ y muebles valorados en 52€). Si el arrendatario de una unidad de almacenamiento no paga el alquiler (el alquiler ha expirado), el contenido puede ser vendido en una subasta como un lote único de artículos: el comprador con la puja más alta compra todo el contenido de la unidad de almacenamiento. Los compradores profesionales tienen un nombre y un presupuesto (en el rango de 10.000€ a 20.000€) y compran las unidades de almacenamiento expiradas para obtener beneficios.

El objetivo de este problema es simular un episodio de la serie de televisión "¿Quién da más?", donde varios compradores profesionales compiten por ser los que tienen mayores ganancias después de una subasta. El beneficio se mide como la diferencia entre el dinero gastado en comprar las unidades de almacenamiento y el valor de los artículos que contienen. Las reglas del juego son las siguientes:

- Habrá 10 trasteros, denominados trastero1, trastero2, etc. cada uno con un número aleatorio diferente de unidades de almacenamiento.

- Las unidades de almacenamiento pueden estar alquiladas o no (esto se decidirá al azar con un 75% de probabilidad de que una unidad de almacenamiento esté alquilada). Cada unidad de almacenamiento alquilada contendrá entre 1 y 10 artículos que pertenecerán a una de las categorías anteriores y tendrán un valor monetario aleatorio. Se debe calcular y guardar el valor total de los elementos contenidos por cada unidad de almacenamiento.
- Las unidades alquiladas habrán expirado con una probabilidad del 25%, estando entonces disponibles para la subasta. Durante la subasta se imprimirá cada unidad de almacenamiento expirada, con sus elementos y su valor estimado total (ver ejemplo a continuación).
- 15 compradores profesionales, denominados comprador1, comprador2, etc. con presupuestos iniciales aleatorios, competirán por las unidades de almacenamiento expiradas. Para cada subasta cada comprador hará una oferta de una cantidad aleatoria entera entre 50% y 100% del valor estimado de la unidad de almacenamiento. El que tenga la oferta más alta compra la unidad. En caso de empate, la primera puja gana. Cuando se compra una unidad, se establece como no alquilada y todos los elementos que contiene pasan a pertenecer al comprador. Considerar que un comprador puede comprar más de una unidad y que no pueden pujar más alto que el dinero que les queda.
- Al final del programa, se debe ordenar los compradores en orden descendente de ganancias y se deben imprimir como en el ejemplo (para simplificar, en el ejemplo se consideran solamente 1 unidad de almacenamiento y 3 compradores)

El programa se evaluará de la siguiente manera:

- a) (1 punto). Descripción de las clases que se utilizarán para representar los conceptos anteriores, incluyendo sus campos y métodos más importantes. Se debe realizar de manera similar a como se hizo en el informe de la práctica final. Máximo una cara.
- b) (0,5 puntos) Creación del conjunto necesario de métodos `set` y `get` dentro de cada clase. No crear más que los necesarios. Considerar qué atributos deben modificarse una vez que se crean los objetos y cuáles nunca se modificarán. Todos los atributos deben ser privados.
- c) (2 puntos) Creación de las unidades de almacenamiento dentro de los trasteros y de los artículos dentro de cada unidad. Todo debe ser realizado al azar dentro de sus respectivos constructores.
- d) (0,5 puntos) Creación de un método denominado `vaciarUnidad` que elimina todos los artículos de una unidad y la establece como no alquilada y no expirada.
- e) (2 puntos) Creación de métodos que permitan a un comprador pujar por una unidad de almacenamiento (la puja no puede ser superior a su presupuesto actual) y agregar nuevos artículos al stock del comprador y vaciar la unidad de almacenamiento correspondiente.
- f) (1 punto) Creación de métodos `toString` para imprimir la información requerida. Situar en las clases más apropiadas.
- g) (0,5 puntos) Creación de un método para ordenar a los compradores por sus ganancias en orden descendente.
- h) (2,5 puntos) Creación del programa principal en el que se definirán los objetos, se realizarán las subastas de las unidades expiradas y se presentarán los resultados finales. Debe ser lo más simple posible y puede contener métodos estáticos si es necesario.

## Observaciones:

- Se requiere el uso de Programación Orientada a Objetos. Los programas realizados en una sola clase obtendrán una puntuación de 5 puntos como máximo.
- Es recomendable crear cada clase en una hoja diferente, de esta manera será más fácil modificarlas y completarlas.

## Ejemplo de ejecución:

```
trastero0, unidad 8
Valor total: 54
Artículos: ropa, ropa, muebles, ropa.
Ofertas
comprador0: 44
comprador1: 34
comprador2: 32
Ganador comprador0
```

```
trastero0, unidad 9
Valor total: 196
Artículos: muebles, electrodomésticos, muebles, ropa, electrodomésticos,
muebles, ropa.
Ofertas
comprador0: 193
comprador1: 189
comprador2: 164
Ganador comprador0
```

```
trastero0, unidad 17
Valor total: 805
Artículos: muebles, muebles, antigüedades, muebles, electrodomésticos,
electrodomésticos, antigüedades, ropa, ropa.
Ofertas
comprador0: 548
comprador1: 449
comprador2: 544
Ganador comprador0
```

```
Los beneficios finales son:
comprador0, beneficio: 270
comprador2, beneficio: 0
comprador1, beneficio: 0
```

```
El ganador es comprador0.
comprador0 compró ropa, ropa, muebles, ropa, muebles, electrodomésticos,
muebles, ropa, electrodomésticos, muebles, ropa, muebles, muebles, antigüedades,
muebles, electrodomésticos, antigüedades, ropa.
```