



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 Programación Avanzada (I/2016)

Tarea 1

1. Objetivos

- Aplicar conceptos de programación orientada a objetos (OOP) para modelar un problema.
- Trabajar con archivos de texto para leer y procesar datos.
- Desarrollar algoritmos para la resolución de problemas complejos.

2. Introducción

Después de haber salido al mercado laboral, a pesar de haber tenido una gran formación como programador(a), te encuentras cesante. Nadie te quiere, y ni siquiera Python es capaz de ayudarte con un print(comida). La situación parece no tener solución... sin embargo, un día recibes una carta de un viejo amigo que fue aceptado en una empresa que te interesó en su momento.

Procedes a leer:

“Querid@ ----- (inserta tu nombre), al escribir esto estoy triste. El antiguo dueño de nuestra empresa ha sido derrocado y REEMPLAZADO POR EL BENÉVOLO BUCCHI, TODOS AMAMOS A BUCCHI Y A SU GLORIOSA ADMINISTRACIÓN. VEN A TRABAJAR CON NOSOTROS”.

Si bien sospechas de la carta, las necesidades mandan, por lo que vas directo a la empresa, ahora llamada Bucchi Incorporated. Están buscando vacantes para su nuevo sistema MercadoPreso, enfocado en un novedoso sistema de subastas online que claramente no busca copiar a ningún sistema antes conocido. Si bien encuentras que hay cosas raras de por medio, en tu estado actual, finalmente decides tomar el puesto.

3. Problema

En esta tarea deberás modelar un sistema de subastas mediante programación orientada a objetos. Esto implica el uso de clases y la relación correcta entre ellas. Además, deberás crear un programa en consola que permita a los usuarios hacer uso del sistema y de sus funcionalidades.

4. MercadoPreso

4.1. Estructura

MercadoPreso funciona mediante la interacción entre las siguientes entidades:

- **Producto:** Todo producto tiene un identificador, un nombre, un rating, un precio de hipoteca y un indicador de si es nuevo o usado. Los productos están clasificados en distintos grupos según su naturaleza y uso:
 - Autos: Posee rendimiento por litro de combustible.
 - Computación: Poseen sistema operativo, memoria y un bool que indica si es portátil o de escritorio.
 - Electrodomésticos: Poseen consumo de energía por hora.
 - Muebles: Poseen dimensiones (tupla).
 - Música y Películas: Poseen duración.
 - Plantas de Jardín: Poseen color y altura.
 - Libros: Poseen autor, editorial y dimensiones (tupla).
 - Vestuario: Poseen la talla de la prenda y el género al que está destinado.
- **Usuario:** Todos los usuarios pueden comprar y vender productos. Posee nombre de usuario (único), contraseña, correo electrónico (único), nombre, apellido, dinero y productos.
- **Subasta:** Posee identificador, vendedor, productos, precio actual, fecha inicio, fecha término, historial (lista) de postor-precio (tupla) y comprador actual.
- **Multa:** Contiene información sobre la multa de usuarios. Posee el nombre de usuario del individuo multado, la cantidad de dinero a pagar y la fecha de vencimiento de la multa.

4.2. Funcionalidades

MercadoPreso debe tener las siguientes funcionalidades:

- **Antes de loguearse en el sistema:**
 - **Sistema de ingreso y registro:** Cada usuario será capaz de hacer uso del sistema luego de autenticarse. Al momento de ingresar al sistema se deben ingresar la fecha y hora actuales, para una correcta interacción con el programa. Al iniciar sesión, automáticamente deberán cerrarse todas las subastas que hayan estado abiertas y cuya fecha de término estén cumplidas. También deberán verificarse los plazos de las multas por vencer. En caso de no estar registrado, se dará la opción de registrar un nuevo usuario, llenando todos los datos pertinentes para que sean agregados al sistema (notar que su sistema debe aceptar nuevos usuarios que contengan un correo electrónico válido, y además un nombre de usuario y correo que no hayan sido utilizados previamente, en caso contrario, no se permitirá crearlos). Todos los usuarios nuevos comienzan con 100.000 MercadiPesos (un regalo de parte del benevolente Bucchi).
- **Luego de ingresar al sistema, las siguientes funcionalidades deben ser implementadas (se debe poder acceder a ellas a través de un menú inicial):**
 - **Subastas disponibles:** Muestra todas las subastas disponibles en las que todavía se puede participar (la fecha actual aún no supera la fecha de término) de la forma: id de la subasta, nombre del producto (si son varios se imprimen los nombres seguidos de un guión), precio, tiempo restante. Se deberá dar la opción de buscar por nombre de producto o tipo de producto. Al seleccionar una subasta, se debe mostrar su información relevante (nombre de producto, precio, historial de apuestas, tiempo restante) y se dará la opción de apostar.

- Apostar en una subasta: Para apostar en una subasta, el usuario debe haber seleccionado alguna. Se le preguntará la cantidad de dinero que desea apostar por el producto. Considere que un usuario puede ingresar una cantidad que supere el dinero que tiene al momento de apostar.
- Crear subasta: El sistema debe permitir crear una subasta. El usuario debe seleccionar los productos que desea subastar, y el sistema debe preguntar por el precio mínimo inicial y la fecha de término.
- Lista de productos: Se muestran todos los productos en el sistema que están en una subasta o en la tienda de productos de MercadoPreso. Por cada producto se debe imprimir su nombre, rating y todos los atributos pertenecientes únicamente al tipo de producto.
- Lista de subastas finalizadas: Deberá mostrar en consola la información esencial sobre todas las subastas que han finalizado.
- Tienda de productos de MercadoPreso: El usuario podrá comprar productos de la tienda de MercadoPreso. Se deberá dar la opción de buscar por nombre de producto o tipo de producto.
- Multas: El usuario podrá acceder a su estado de multas (cantidades y plazos) y podrá pagarlas mediante dinero o productos que posea.
- Salir del sistema: Cuando un usuario desee dejar de utilizar el sistema, el programa debe cerrarse y todos los cambios realizados (compras, ventas, nuevos usuarios, etc) deben actualizarse en los archivos de texto correspondientes.
- Consultas: El usuario podrá realizar las siguientes consultas:
 - Subastas por usuario involucrado: Ingresando el nombre de usuario, se deberán mostrar todas las subastas en la que este se haya visto involucrado (comprador, vendedor, o apostador).
 - Ranking de vendedores: Se deberá mostrar una lista con los 10 usuarios que más dinero han obtenido debido a ventas en subastas.
 - Trayectoria de un producto: Ingresando el id del producto, se deberá mostrar en consola cada dueño y transacción en donde se haya visto involucrado el producto especificado.

5. Sistema de apuestas

Al apostar, el usuario está indicando la cantidad máxima que está dispuesto a pagar en ese momento. Si es el primer postor, entonces el precio actual del producto será simplemente un MercadiPeso mayor al precio inicial. Así, si un usuario apuesta una cantidad menor a la máxima ingresada por el ganador vigente, el precio actual del producto será un MercadiPeso mayor a la cantidad ingresada por el segundo postor, salvo que el ganador actual seguirá siendo el primero. En cambio, si el segundo postor ingresara un máximo mayor, el precio actual sería un MercadiPeso mayor a la cantidad del primer postor, siendo el segundo ahora el ganador. Notar que los usuarios no pueden ver el máximo que ha apostado cada uno, solo pueden ver el precio actual de la subasta y el ganador actual. En caso de que ambos apuesten el mismo máximo, el precio actual será ese mismo, pero el primer postor será considerado el ganador actual. Si apuesta una cantidad menor o igual al precio actual, la apuesta no será ingresada.

6. Finalización de subastas y Sistema de Multas

Si al finalizar la subasta el ganador cuenta con la cantidad de dinero pactada, inmediatamente el vendedor recibe su dinero, este se resta del total del comprador y el producto pasa a ser del mismo. Si al momento de finalizar la subasta, el ganador (desde ahora llamado multado) no tiene la cantidad de dinero final, el ganador será el próximo que haya apostado la mayor cantidad de dinero por el producto.

Un usuario multado deberá pagar la totalidad del dinero que apostó en la subasta en un plazo de 2 días. La multa se puede pagar con dinero, con productos del multado (su valor corresponde al precio de hipoteca del producto) hasta alcanzar el valor de la multa, o con una mezcla de dinero y productos. En caso de no poder pagar la multa en su totalidad pasado el plazo, el usuario es baneado permanentemente del sistema y todos sus productos son embargados y entregados a la Tienda de productos de MercadoPreso. Además, debe restringirse la creación de nuevos usuarios que utilicen el correo electrónico del baneado (así evitamos que vuelva a ingresar al sistema... con el mismo correo al menos).

7. Tienda de productos de MercadoPreso

Todos los productos entregados por usuarios multados y usuarios baneados son agregados a la tienda de productos de MercadoPreso. Esta tienda no funciona igual que las subastas, ya que los productos se venderán inmediatamente al comprador al 150 % de su precio de hipoteca.

8. Hackeo

A pesar de todo lo que el benevolente Bucchi ha hecho por ti, decides crear un programa a parte del solicitado por la empresa (es decir, en otro archivo) que te permita obtener los productos que desees apostando la menor cantidad posible de veces (quedaste traumatado después de tanto tiempo sin comer bien). El sistema recibirá tu usuario y una subasta vigente, y a través de esta última, primero obtendrá todos los datos de las apuestas de dicha subasta, y además, los datos de todos los apostadores que participan de ella. De esta forma, tu sistema deberá estudiar todas las apuestas que han hecho los competidores en otros productos, y a través de esos valores y su monto de dinero actual, el sistema deberá encontrar el monto para el cual ninguno de los usuarios pueda superar tu apuesta sin tener que caer en una situación de multa (si llegasen a hacerlo, de todas formas ganarías por default). Esto te permitirá apostar de forma segura sin necesidad de preocuparte de ir cambiando tu máxima apuesta, o bien dejar de lado subastas que posiblemente no ganarás sin endeudarte. Recuerda que los resultados pueden ir cambiando a medida que se añaden competidores a la subasta.

9. Diagrama de Clases

Junto con el programa pedido, deberá enviar vía Siding un diagrama de clases con todo el modelamiento del problema. Esto incluye clases junto con sus métodos y atributos, y todas las relaciones existentes entre estas (asociación, composición y herencia).

10. Carga de Datos

Junto con el enunciado se encuentran muchos archivos de texto. Se detallará el contenido de cada uno a continuación:

■ usuarios.txt

Contiene la información de los usuarios en el sistema.

- nombre [str]: Nombre y apellido del usuario.
- nombre de usuario [str]: Nombre del usuario en el sistema.
- contraseña [str]: Contraseña del usuario en el sistema.
- correo electrónico [str]: Correo electrónico del usuario.
- dinero [int]: Indica la cantidad de dinero actual del usuario.
- productos [list]: Lista que contiene los id de los productos del usuario.

■ productos.txt

Contiene la información de los productos en el sistema.

- id [int]: Identificador del producto.
- nombre [str]: Nombre del producto.
- tipo [str]: Tipo de producto.
- rating [float]: Rating del producto (calidad).
- precio de hipoteca [int]: El precio de hipoteca del producto.
- nuevo [SI/NO]: Establece si es nuevo o no.

El resto de los atributos depende del tipo de producto.

■ subastasact.txt

Posee toda la información necesaria sobre todas las subastas activas. Se encuentra de la forma:

- id [int]: Identificador de la subasta.
- vendedor [str]: Nombre de usuario del vendedor de los productos.
- productos [list]: Lista con los id's de los productos que se están subastando.
- precio [int]: Precio actual de los productos en la subasta.
- fecha inicio [str]: Fecha de inicio de la subasta de la forma: dd/mm/aa/hh:minmin.
- fecha término [str]: Fecha de finalización de la subasta de la forma: dd/mm/aa/hh:minmin.
- historial de postor-precio [list]: Lista de tuplas postor precio de la forma: usuario-precio.

■ subastasfin.txt

Posee toda la información necesaria sobre todas las subastas finalizadas. Se encuentra de la forma:

- id [int]: Identificador de la subasta.
- vendedor [str]: Nombre de usuario del vendedor de los productos.

- productos [list]: Lista con los id's de los productos que se están subastando.
 - precio [int]: Precio final de los productos en la subasta.
 - fecha inicio [str]: Fecha de inicio de la subasta de la forma: dd/mm/aa/hh:minmin.
 - fecha término [str]: Fecha de finalización de la subasta de la forma: dd/mm/aa/hh:minmin.
 - historial de postor-precio [list]: Lista de tuplas postor precio de la forma: usuario-precio.
 - comprador [str]: Nombre de usuario del comprador de los productos en la subasta.
- **tiendamercado.txt**
 Posee toda la información necesaria sobre la tienda de productos de MercadoPreso. Se encuentra de la forma:
- id [int]: Identificador del producto.
- **multa2.txt**
 Posee toda la información necesaria los usuarios multados. Se encuentra de la forma:
- usuario [str]: Nombre de usuario del individuo baneado.
 - monto [int]: Cantidad de MercadiPesos a cancelar.
 - plazo [str]: Fecha de vencimiento de la multa de la forma: dd/mm/aa/hh:minmin.
- **banea2.txt**
 Posee toda la información necesaria los usuarios baneados. Se encuentra de la forma:
- usuario [str]: nombre de usuario del individuo baneado.

11. Restricciones y alcances

- Tu programa debe ser desarrollado en Python 3.4
- Esta tarea es estrictamente individual, y está regida por el Código de Honor de la Escuela: Clickear para Leer.
- Su código debe seguir la guía de estilos PEP8
- Si no se encuentra especificado en el enunciado, asuma que el uso de cualquier librería Python está prohibido. Pregunte por foro si se pueden usar librerías específicas.
- El ayudante puede castigar el puntaje¹ de tu tarea, si le parece adecuado. Se recomienda ordenar el código y ser lo más claro y eficiente posible en la creación algoritmos.
- La revisión de la tarea será realizada con distintos archivos **.txt**.
- Debe adjuntar un archivo **README.md** donde comente sus alcances y el funcionamiento de su sistema (*i.e.* manual de usuario) de forma *concisa* y *clara*.
- Cree un módulo para cada conjunto de clases. Divídalas por las relaciones y los tipos que poseen en común.
- Cualquier aspecto no especificado queda a su criterio, siempre que no pase por encima de ningún otro.

¹Hasta -5 décimas.

12. Entrega

- **Fecha/hora:** 28 de Marzo del 2016.
- **Lugar:** GIT - Carpeta: Tareas/Tarea_01

Tareas que no cumplan con las restricciones señaladas en este enunciado tendrán la calificación mínima (1.0).