

Programación 2
Tecnatura Universitaria en Desarrollo de Aplicaciones Informáticas
Trabajo Práctico Especial– Segunda Parte – 2015

Entrega

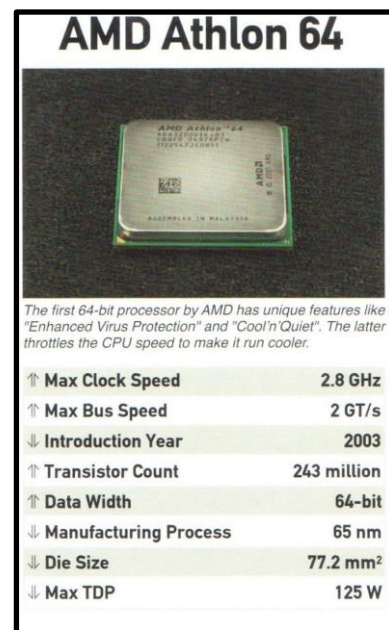
Viernes 11 de Septiembre - Por mail a prog2tudai@gmail.com

Juego De Cartas - Continuación

Defina las clases (nombre, superclase, atributos y métodos) para implementar una solución orientada a objetos para el siguiente problema e implemente en Java.

Al diseño definido en el práctico especial anterior se le incorpora la capacidad de que cada atributo defina si la contienda es por mayor o por menor. Por ejemplo en el caso de la carta “AMD Athlon 64” al comparar por el atributo *Max Clock Speed* gana quién tiene el valor más alto, mientras que al comparar el atributo *Introduction Year* gana quién tiene el valor más bajo.

Un atributo que está asignado a una carta no cambia nunca su forma de compararse.



Este cambio debe ser tenido en cuenta en el momento de la construcción del mazo, de manera tal de que a los controles anteriores (mismos atributos en cada carta) se controle que estos se definan por la misma forma.

La mecánica del juego queda de la siguiente forma: Se reparten las cartas en dos partes iguales, el primer jugador toma la primera del montón y selecciona un atributo de los disponibles. Se comparan las primeras cartas de cada jugador y el que posea la carta ganadora (acorde a la forma del atributo seleccionado) se lleva las dos cartas (van detrás de su montón) y también tiene el turno en la siguiente ronda. El juego termina cuando uno de los dos jugadores se queda sin cartas.

Programación 2
Tecnatura Universitaria en Desarrollo de Aplicaciones Informáticas
Trabajo Práctico Especial– Segunda Parte – 2015

En caso de empate, se toman las siguientes cartas y no se cambia el atributo por el cual se compite, esto se repite hasta que haya un ganador (si uno de los jugadores se queda sin cartas gana el otro jugador y si los dos se quedan sin cartas se declara un empate).

Alcance del primer práctico especial

Diseño e implementación de las clases involucradas con sus atributos y responsabilidades.

Se debe proveer funcionalidad que permita la creación de mazos y que un mazo de cartas pueda verificar que todas sus cartas son correctas (mismos atributos, misma forma cada atributo y cantidad).

Implementar el juego por línea de comando (se crean los objetos en el main) se juega eligiendo de forma aleatoria el atributo por el cual se van a enfrentar los jugadores.

Se debe generar una traza por pantalla de los resultados de cada ronda.

Es fundamental que la elección de la carta sea una funcionalidad del Jugador, ya que esto en un futuro es lo que permitiría incorporar una estrategia al juego o simplemente incorporar la selección de forma manual.

El archivo comprimido **trabajo.zip** se encuentra un mazo de cartas a modo de ejemplo el cual puede ser utilizado para la realización de trabajo. En ese caso puntual se asume que altura y peso es por menor valor, mientras que los demás atributos por mayor valor.

La entrega del práctico debe incluir:

- Un archivo comprimido con el código fuente de todas las clases implementadas.
- Un informe describiendo el diseño elegido, haciendo énfasis en las clases y métodos que considere más relevantes. Asimismo, el informe debe contener un diagrama de clases. Es importante incluir una explicación de las decisiones que permitieron resolver la funcionalidad solicitada en el ejercicio.