Introducción a Ciencias de la Computación
2020-2
Práctica 2: Clases y Objetos

Pedro Ulises Cervantes González
confundeme@ciencias.unam.mx

Emmanuel Cruz Hernández
emmanuel_cruzh@ciencias.unam.mx

Adriana Sánchez del Moral
adrisanchez@ciencias.unam.mx

Víctor Zamora Gutiérrez
agua@ciencias.unam.mx

Fecha límite de entrega: 5 de Marzo de 2020.
Hora límite de entrega: 23:59.

1. Objetivo

Conocer los conceptos de clases y objetos, así como saber traducirlos en *Java*. Además de tener una introducción a la programación de métodos e identificar claramente la diferencia entre los tipos de métodos.

También se espera que sepan utilizar los métodos de una clase para poder realizar operaciones o acciones propias de un objeto. Asimismo, saber cómo darle características apropiadas a un objeto para poder resolver un problema.

2. Actividad

Crea una clase llamada *Mascota* que modele una mascota.

2.1. Actividad 1 *(1 punto)*

- Debe haber al menos un atributo público.
- Debe haber al menos un atributo privado.
- Debe haber al menos una variable *final* como atributo.
- Puedes agregar todos los atributos que consideres convenientes para la representación de una mascota.

2.2. Actividad 2 *(2 puntos)*

- Debes implementar al menos dos métodos constructores para crear una mascota.
- Para cada uno de los atributos, debe haber un método de acceso y un método mutante.

2.3. Actividad 3 *(3 puntos)*

Además de los atributos que creaste anteriormente, crea uno que almacene lo que la mascota adora más de su dueño y otro que almacene sus recuerdos.

Crea los métodos que consideres necesarios que te permitan modelar el siguiente escenario:

- Si una mascota A le cuenta a una mascota B, lo que más adora de su dueño, entonces, la mascota A recordará lo que la mascota B adora más de su dueño.
- Si una mascota A le cuenta a una mascota B, lo que más adora de su dueño, pero ya tiene información almacenada en sus recuerdos, entonces, la mascota A recordará lo que la mascota B adora más de su dueño y, además, recordará las cosas que ya sabía antes.

Al principio, las mascotas no tienen recuerdos. Primero se deben contar entre ellos lo que más les gusta de su dueño. A partir de ese momento, las mascotas comienzan a contarse lo que más adoran de sus dueños y comienzan a generar recuerdos.

Crea un método *main* con al menos 6 mascotas y simula el escenario descrito, con la regla de que una mascota no puede recordar información de más de tres mascotas.

Al final, imprime el nombre de cada mascota, lo que más adora de su dueño y los recuerdos que almacena sobre otras mascotas. Puedes sobre-escribir el método *toString()*.

2.4. Actividad 4 *(1 punto)*

Crea una variable estática que indique cuantas instancias de la clase *Mascota* has creado.

Al final del programa, imprime el valor de la variable estática que creaste para crear instancias de *Mascota* y comprueba que corresponda al número de ejemplares.

Responde a la pregunta *¿todas las mascotas tienen el mismo valor en dicha variable o es distinto para cada mascota?* en tu archivo Readme.txt.

2.5. Actividad 5 *(3 puntos)*

Crea una clase *Veterinario* que modele a un veterinario que puede hacer que dos mascotas se hagan mejores amigos.

En esta clase, debes crear un método que permita hacer que dos mascotas se hagan mejores amigos mutuamente. Esto es que si A es mejor amigo de B, entonces B es el mejor amigo de A.

Un veterinario tiene la capacidad de revertir sólo la última relación de amistad que creó entre dos mascotas. Eso significa que sí antes de que la mascota A y B fueran mejores amigos, A y C eran mejores amigos, B y D eran mejores amigos, entonces A y B volverán a ser mejores amigos de B y D, respectivamente.

Agrega los métodos que consideres necesarios a la clase *Mascota* para simular este escenario. Crea una método *main* en la clase *Veterinario* donde instancies al menos 1 veterinario y al menos 6 mascotas. Tu deber es hacer que todas las mascotas tengan un mejor amigo, revirtiendo al menos 3 relaciones de amistad con ayuda de un Veterinario.

Imprime en el método *main* de la clase *Veterinario* el nombre de cada mascota y el nombre de su mejor amigo.

3. Reglas importantes

- Cumple con los lineamientos de entrega.
- Todos los archivos deberán contener nombre y número de cuenta.
- Tu código debe estar bien comentado. Esto abarca clases, atributos, métodos y comentarios extra.
- Para cada clase solicitada, crea un nuevo archivo.
- Queda **prohibido importar bibliotecas o usar clases ajenas a las de la práctica**.
- Utiliza correctamente las convenciones para nombrar variables, constantes, clases y métodos.
- Queda prohibido usar los enunciados *if*, *if... else*, *do... while*, *while*, *for*, o algún derivado.
- En caso de no cumplirse alguna de las reglas especificadas, se restarán 0.5 puntos de tu calificación obtenida.