

INF-253 Lenguajes de Programación

Tarea 3: Java

Profesor: José Luis Martí Lara

Ayudante Cátedras: Sebastián Godínez San Martín

Ayudante Tareas: Gabriel Carmona Tabja - Sebastián Campos Muñoz

2 de junio de 2020

1. Nueva Vida

Mario después de una larga rehabilitación, logró recuperarse y volver a sus aventuras lleno de alegría y felicidad, sin embargo, Giuseppe no tuvo tanta suerte. Una vez que Mario terminó su hospitalización, Giuseppe decidió aprender a programar en C uniéndose a un curso en línea patrocinado por Inezgod, el famosísimo programador genio que logró la paz mundial. Grande fue su sorpresa cuando su aprendizaje fue sabotado por dos malvados goombas que le hacían evaluaciones poco claras, lo cual provocaba que el tuviera que rehacer su código en repetidas ocasiones, frustrándolo y abandonando la vía de C. En su infinita tristeza, Giuseppe decidió descargarse MIMS, un videojuego en el cual puedes emular una vida feliz (la cual le fue arrebatada de sus manos). Después de largas jornadas de juego, Giuseppe recuperó la fé en la informática, y decidió aprender JAVA emulando de forma muy simple la manera en que funcionaba MIMS; sin embargo, ya no confía en nadie más que en ustedes, por lo que les pide ayuda para que le enseñen a utilizar Java y de esa forma ayudar a Giuseppe a recuperar las ganas de vivir.

2. Idea General

Giuseppe sólo quiere aprender Java, así que su versión de MIMS será muy simplificada. En esta versión, los personajes creados podrán comer, dormir, comprar alimento, trabajar, estudiar, casarse, tener hijos, alimentar a los hijos y hacer dormir a los hijos. Al inicio, sólo existirán 4 adultos (18 años) los cuales podrán hacer lo indicado anteriormente. Una vez casados podrán tener hijos, los cuales deberán alimentar a través del tiempo. Los hijos podrán jugar, y con el tiempo crecerán y podrán hacer todas las cosas que pueden hacer los adultos.

3. Estructura

3.1. Personaje

Un personaje es una clase abstracta está definido según lo indicado en la figura 1. Donde sus atributos significan lo siguiente:

- Nombre : Nombre del personaje
- Sexo : Sexo del personaje
- Dinero : Cantidad de dinero (comienza en 0)

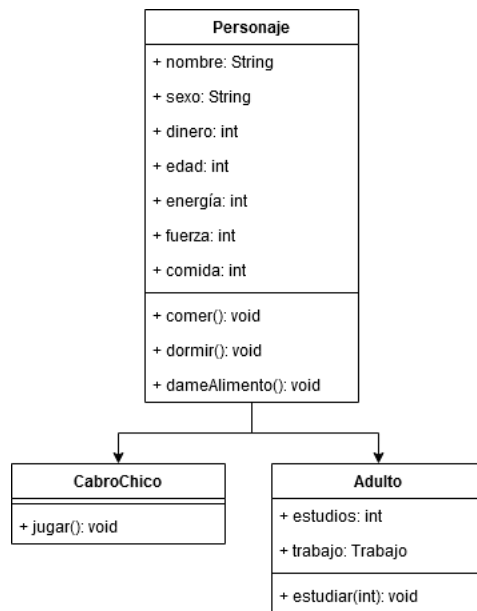


Figura 1: Personaje

- Edad : Edad del personaje
- Energía : Valor que indica el energía del personaje (comienza en 100)
- Fuerza : Indica el valor del fuerza del personaje (todos comienzan con 30)
- Comida: Cantidad de comida que tiene el personaje (comienza en 0)

Los personajes realizan acciones (las cuales se definirán más adelante), estas tienen beneficios y consecuencias.

Cada vez que se realizan cuatro acciones, todos los personajes envejecen 1 año. Estas cuatro acciones son todas menos: esculpir, pintar, escribir, enseñar, investigar, experimentar, comer, acostar hijos y dame alimento. Un personaje al llegar a los 18 años se vuelve adulto, y al llegar a los 80 años muere (por lo que se borra su existencia de todos lados). Por otro lado, un personaje puede comer, transformando la cantidad de comida en un aumento de fuerza, en una relación de 3 unidades de comida es a 1 unidad de fuerza para un niño y 6 unidades de comida es a 1 unidad de fuerza para un adulto. Si la fuerza llega a 0, el personaje muere. Cada acción menos esculpir, pintar, escribir, enseñar, investigar, experimentar, comer y dame alimento le quitará 2 fuerza a todos los personajes. La función dame alimento intercambia dinero por alimento en razón 4 es a 6.

Algunas acciones consumirán energía, si un personaje tiene 0 o menos de energía, no podrá realizar ninguna acción que consuma energía hasta que duerma. Si alguna acción descuenta más energía del que posee, el personaje termina la acción antes de no poder hacer otra. Cada personaje tendrá como máximo 100 de energía, dormir recupera 20 a 50 de energía (si se supera los 100 de energía al dormir, se deja en 100)

Existen dos clases que implementan personaje: cabro chico y adulto.

- Cabro chico: Tiene la característica especial de que puede jugar, al jugar hay una pequeña probabilidad de encontrar dinero o alimento en pequeñas cantidades, pero consume energía. Existe un 10 % de probabilidades de encontrar dinero y 5 % de encontrar alimento, el dinero

capaz de encontrar es entre 2 a 3 unidades y de alimento de 1 a 2. Pierde 2 a 10 de energía al jugar.

- Adulto: El adulto puede estudiar o trabajar. Si estudia puede elegir un trabajo, no puede trabajar sin estudiar antes. Estudiar quita 50 de energía. Si tiene un trabajo, desbloquea la acción trabajar que viene implícita en su trabajo. La variable estudio será : 1: Científico , 2: Humanista, 3: Artístico y 4: ?????

3.2. Trabajo

Los trabajos siguen la siguiente estructura:

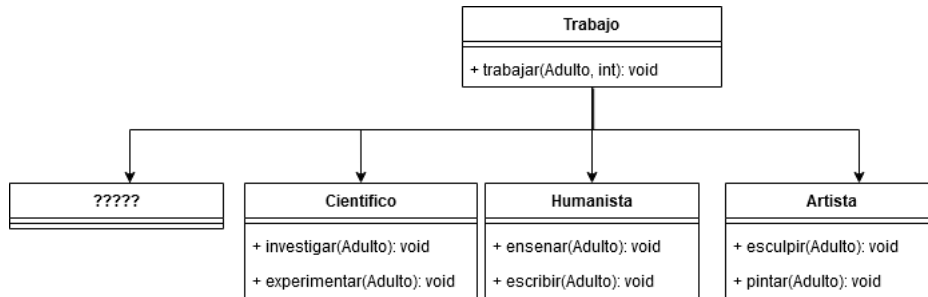


Figura 2: Trabajo

Donde Trabajo es una interface y el resto clases que implementan Trabajo.

Trabajo tiene el método trabajar, el cual recibe un objeto de tipo adulto y un entero que señala la cantidad de horas que va a trabajar. Cada trabajo tiene 2 acciones que consumen 1 hora cada uno, cada acción genera dinero y energía de forma aleatoria (utilizando la función random implementada en Java). Si aún le quedan horas de trabajo y tienes 0 o menos de energía, no podrá seguir trabajando hasta dormir.

Los trabajos disponibles son:

- Científico: Los científicos son respetados, por lo que ganan bastante dinero, pero son explotados por lo que se cansan bastante. Investigar otorga entre 10 a 20 unidades y pierde entre 10 a 15 de energía. En cambio, experimentar otorga entre 10 a 30 unidades, pero quita entre 10 a 20 de energía.
- Humanista: Los humanistas le gusta su trabajo y les pagan bien, pero al ser mirados en menos se cansan mucho. Enseñar otorga entre 9 a 13 unidades pero pierdes entre 6 a 9 de energía, mientras que escribir otorga de 4 a 14 unidades pero pierdes entre 4 a 13 de energía.
- Artístico: Los artistas tienen bastante vocación por lo que se cansan menos al trabajar, pero lamentablemente no son tan bien remunerados. Al pintar puedes ganar entre 4 a 8 unidades pero pierdes entre 3 a 7 de energía. En cambio, si esculpes puedes ganar entre 0 a 25 unidades, y perder entre 6 a 30 de energía.
- ?????: Trabajo inventado por ustedes :D, debe contener dos acciones siguiendo el mismo modelo que los anteriores y debe implementar la interfaz trabajo.

3.3. Familia

Las familias siguen la siguiente estructura:

Familia
+ adulto1: Adulto
+ adulto2: Adulto
+ hijos: ArrayList<Personaje>
+ alimentarHijos(String, int): void
+ hacerHijo(String): CabroChico
+ acostarHijos(): void

Figura 3: Familia

Estas son una clase que tiene dos atributos de tipo Adulto, la cual corresponde a los dos adultos que contrajeron matrimonio y una lista que contiene objetos de tipo Personaje, los cuales corresponden a los hijos del matrimonio.

Dentro de familia se tiene los siguientes métodos a implementar:

- alimentarHijos, que recibe el nombre del adulto que gastará comida para sus hijos y la cantidad de comida que se le dará si o si a cada hijo. A todos los hijos se le da la misma cantidad de comida. Si el adulto se queda sin comida suficiente para dar, los hijos que falten no recibirán alimento. El orden de alimentación es del último hijo hasta el primero.
- hacerHijo, recibe un string con el nombre del hijo que les va a llevar la cigüeña. El sexo del hijo debe ser elegido al azar entre Hombre y Mujer con una probabilidad de 50 % cada uno. Edad, dinero y comida se inicializan en 0. Fuerza se inicializa en 30. Energía se inicializa en 100. Finalmente se agrega a la lista de hijos de la familia.
- acostarHijos, los adultos de la familia ponen a dormir a cada uno de sus hijos. En esta situación poner a dormir todos los hijos se considera una acción (ósea que todos se duermen a la vez).

3.4. Main

La función main() debe estar dentro de una clase MIMS, la cual debe tener la siguiente estructura:

Mims
+ personajes: ArrayList<Personaje>
+ familia: ArrayList<Familia>
+ agregarAdulto(Adulto): void
+ casarse(Adulto, Adulto): void
+ main(String[]): void

Figura 4: Mims

Donde cada variable es:

- personajes: Lista con todos los personajes vivos del juego.
- familia: Lista con todas las familias del juego.

Y los métodos son:

- agregarAdulto: Se crea un adulto artificialmente pidiendo todos sus valores.
- Casarse: Forma una familia a partir de los nombres de dos adultos. Si uno de los adultos pertenece a alguna familia como hijo deben sacarlo de ahí. Una persona ya casada no se puede casar de nuevo.

4. Funcionamiento

Al ejecutar el programa, a través de un menú por consola, debería poder elegir un personaje disponible, una familia o un método del main para ejecutar. Además debería poder ver el estado de cada uno de los personajes (su nombre, fuerza ,etc) y los integrantes de cada una de las familias. Se deben describir todas las acciones realizadas (por ejemplo si compro comida se debe printear "Ha comprado X comida por x monedas", no tiene que ser literal, la forma de describir las acciones se lo dejamos a su criterio). Finalmente, debe haber una opción para salir del juego.

5. A entregar

- Main.java
- Cientifico.java (Se les entregará programado para llegar y usar)
- Familia.java
- Trabajo.java, Humanista.java, Artista.java y ????.java (nombre a elección del alumno)
- Personaje.java, CabroChico.java, Adulto.java
- makefile
- readme.txt

6. Sobre Entrega

- El código debe venir ordenado.
- Se darán puntos por creatividad :D.
- Cada método debe llevar un comentario que indique nombre de la función, parámetros que recibe, una breve descripción de lo hace y lo que retorna.
- Debe estar presente el archivo **makefile o ant** para que se efectúe la revisión, este debe compilar **TODOS** los archivos.
- El trabajo es individual.
- La entrega debe realizarse en tar.gz y debe llevar el nombre:
Tarea3LP_RolIntegrante.tar.gz
- El archivo README.txt debe contener nombre y rol del alumno e instrucciones detalladas para la compilación y utilización de su programa. **SE DESCONTARÁ SI FALTA ALGO DE LO SEÑALADO.**
- El no cumplir con las reglas de entrega conllevará un descuento máximo de 30 puntos en su tarea.

- La entrega será vía aula y el plazo máximo de entrega es hasta el **jueves 18 de Junio a las 23:55 hora aula**.
- Por cada día de atraso se descontarán 20 puntos.
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades.

7. Calificación

- Personaje (25 pts):
 - CabroChico (10 pts)
 - Adulto (15 pts)
- Trabajo (30 pts):
 - Humanista (10 pts)
 - Artista (10 pts)
 - ??? (10 pts)
- Familia (20 pts)
- Main (25 pts)
- Warning (-2 pts c/u, Max 20pts)
- No compila: Recorrección