Nombre y Apellido: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Cantidad de hojas entregadas: \_\_ Nota:\_\_\_\_\_\_\_\_

|  |
| --- |
| **Lea atentamente TODO el enunciado antes de empezar a desarrollar, recodar generar todas las firmas de las funciones y las especificaciones correspondientes para que el proyecto compile antes de programar la solución** |

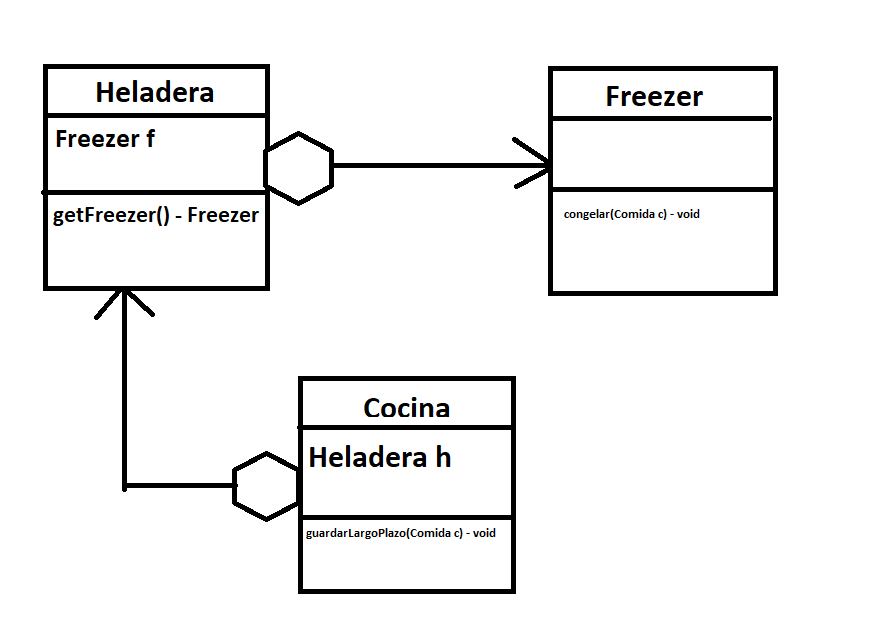
1) Los GuerrerosZ intentan salvar el mundo de amenazas de los EnemigosDeLaTierra. Para ello se tienen distintos tipos de GuerrerosZ con distintos ataques cada uno que causan un cierto daño. Desde ya sabemos que todos ellos tienen mucha fuerza en sus brazos por lo que un punietazo de cualquier GuerreroZ restara 50 punto de vida, Goku, Gohan y Piccolo son GuerrerosZ y tienen respectivamente el ataque Kamehameha, Masenko y makankosappo que, también respectivamente quitan 150,110,90 de vida.  
Se pide:  
a) Implementar en Java la clase GuerreroZ. También implementar las subclases Goku Gohan y Piccolo donde en ellas habra un único método que les permite hacer su propio ataque característico.  
Cada GuerreroZ debe poder lanzar dos ataques (El punietazo y el ataque especial correspondiente)

ACLARACION: Esta escrito puñetazo como punietazo en la consigna, ya que asi debe llamarse el metodo.   
AYUDA: Los metodos de ataque no tienen ningún valor de retorno, solo quitan vida.

Simplificación: Tanto el punietazo como el ataque especial reciben un objeto de clase “EnemigoDeLaTierra” al cual se puede usar una función llamada restarVida(int Vida) supuestamente ya implementada. Por ejemplo: e.restarVida(60); (Siendo ·e· instancia de la clase “EnemigoDeLaTierra”)

b) Como Goku es un guerrero con mucha fuerza, su punietazo debe hacer 5 puntos más de daño.

c) Si un enemigo se encuentra debilitado, es decir, su vida es igual a 0 (cero), al intentar lanzar cualquier ataque se debe lanzar EnemigoDebilitadoException

2) Dado el siguiente digarama de clases UML:

a) Implementar el método public boolean guardarLargoPlazo(Comida c) que recibe una Comida y la congela.  
Si el proceso de congelar fue exitoso devuelve true, sino devuelve false.

Para lograrlo, desde la clase Cocina debe hacerse uso de los atributos y métodos de las clases relacionadas, sabiendo que UNICAMENTE la clase Freezer debe ejecutar metodos de la clase Comida. En otras palabras, es en esta clase (Freezer) es donde debe hacerse uso del método congelar de la clase Comida.

El método debe estar sobrecargado tanto para recibir una sola Comida c, como un conjunto de comidas en un ArrayList.

Ayuda: Para sobrecargar un método, lo ubico que hace falta es escribir de nuevo el método pero cambiando los parámetros que recibe en su firma, en este caso pasa de recibir una Comida, a un ArrayList.

b) Manejo de excepciones:

Dentro de la clase Freezer: Si se intenta congelar una comida que ya esta congelada debe lanzar ComidaYaCongeladaException

Dentro de la clase cocina: Si al intentar congelar una comida, recibe como respuesta del freezer un ComidaYaCongeladaException, debe manejar la excepción y devolver false(el proceso de congelar fue fallido).