

Examen I de programación Orientada a Objetos

Instrucciones cree en Java un proyecto llamado Examen, donde desarrollara lo solicitado en el examen. Una vez terminado guarde la carpeta de su proyecto en un archivo zip y subirlo a la plataforma virtual.

Se le ha contratado para generar una solución en una tienda de Baleadas, donde el `main` actuará de cliente que ordena baleadas a la tienda.

1) Cree la interfaz `Food` (5 ptos.):

- `getName()`: Retorna el nombre del alimento.
- `getCalories()`: Retorna la cantidad de calorías.
- `getPrice()`: Retorna el precio final.

2) Cree la clase abstracta `AbstractFood`, esta clase implementa la interfaz `Food` y tiene los campos `name`, `price`, `calories`. Esta clase abstracta implementa los métodos de `Food` (5 ptos.).

3) Cree la clase abstracta `Ingredient` que hereda de `AbstractFood` (10 ptos.).

4) Implemente las clases concretas `Platano`, `Huevo`, `Aguacate`, `Carne`. Todas las clases anteriores extienden de `Ingredient`, los valores de sus campos son (15 ptos.):

- `Platano`:
 - `name`: platano
 - `calories`: 194
 - `price`: 2
- `Huevo`:
 - `name`: huevo
 - `calories`: 299
 - `price`: 1
- `Aguacate`:
 - `name`: aguacate
 - `calories`: 495
 - `price`: 5
- `Carne`:
 - `name`: carne
 - `calories`: 15
 - `price`: 5

5) Cree la clase Abstracta `Baleada`, esta clase extiende de `AbstractFood`. (15 ptos.):

- `ingredients`: Lista de ingredientes de la Baleada.
- El constructor de `Baleada` instancia una lista vacía de ingredientes (hint: puede usar `ArrayList`)
- `addIngredient(Ingredient ingredient)`: agrega un ingrediente a la Baleada.

- `getName()`: El nombre de una Baleada, es su propio nombre más los nombres de sus ingredientes.
- `getCalories()`: Las calorías de una Baleada, es su propia caloría más las calorías de sus ingredientes.
- `getPrice()`: El precio de una Baleada, es su propio precio más los precios de sus ingredientes.

6) Implemente las clases concretas `ThickDough` (Masa gruesa) y `ThinDough` (Masa delgada). Estas las clases extienden de `Baleada`, los valores de sus campos son (15 pts.):

- `ThickDough`:
 - `name`: thick dough
 - `calories`: 100
 - `price`: 5
- `ThinDough`:
 - `name`: thin dough
 - `calories`: 50
 - `price`: 3

7) Implemente la clase concreta `BaleadaStore`, esta clase (15 pts.):

- `orderSimpleBaleada()`: Retorna una Baleada de masa delgada, con plátano y huevo.
- `orderOnlyMeatBaleada()`: Retorna una Baleada de masa gruesa, con queso y pepperoni.
- `orderOnlyCompleteBaleada()`: Retorna una baleada de masa delgada, con Aguacate ,Huevo y Carne.
- `addIngredient(Baleada baleada, Ingredient ingre)`: Dada una baleada, retorna una baleada con el ingrediente extra.

8) En el `main` instancie la tienda de baleada y genere distintas baleadas (10 pts.).

9) Discusión (10 pts.):

- ¿Qué habría que hacer para agregar bebidas gaseosas y tortillas con quesillo?
- ¿Qué cambios habría que hacer si se quiere agregar más tipos de tiendas?