Healthy Heaven

Preparation

Download the skeleton provided in Judge. **Do not** change the **packages**.

Pay attention to name the package, all the classes, their fields and methods exactly the same way they are presented in the following document. It is also important to keep the project structure as described above.

Problem description

Your task is to create a repository(restaurant) which stores salads by creating the classes described below.

Vegetable

First, write a Java class **`Vegetable`** with the following fields:

- **name: String**
- **calories: int**

The class **constructor** should receive **name** and **calories**.

The class also should have the methods:

- Getter **`getName()`**
- Getter **`getCalories()`**
- Override the **`toString()`** method in the following format:
  `" - {name} have {calories} calories"`

Salad

Next, write a **Java** class **`Salad`** that has **products** (a collection field which stores the entity **Vegetable**). All entities inside the repository have the **same fields**. Also, the **`Salad`** class should have those fields:

- **name: String**

The class **constructor** should receive **name**, also it should initialize the **products** with a new instance of the collection**.**

The class also should have the methods:

- Getter **`getName`**
- Method **`getTotalCalories()`** – returns the sum of all vegetable calories in the salad
- Method **`getProductCount()`** - **returns** the **number** of products
- Method **`add(Vegetable product)`** – adds an entity to the products
- Override **`toString()`** – by the format bellow:
  "* Salad {name} is {calories} calories and have {product count} products:
  {Vegetable 1}
  {Vegetable 2}
  {Vegetable 3}
  {...}"

Restaurant

Next, write a **Java** class **`Restaurant`** that has **data** (a collection which stores the entity **Salad**). All entities inside the repository have the **same fields**. Also, the **`Restaurant`** class should have those fields:

- **name: String**

The class **constructor** should receive **name**, also it should initialize the **data** with a new instance of the collection.

Implement the following features:

- Field **`data`** – **collection** that holds added salads
- Method **`add(Salad salad)`** – adds an entity to the data
- Method **`buy(String name)`** – removes a salad by given name, if such exists, and returns boolean
- Mehod **`getHealthiestSalad()`** – returns the healthiest salad
- Method **`generateMenu()`** - **returns** a **string** in the following **format**:
  "**{name} have {salad count} salads:**
  {Salad 1}
  {Salad 2}
  {…}"

Constraints

- The **names** of the vegetables and salads will be **always unique**.
- The **calories** of the vegetables will always be with **positive values**.

Examples

This is an example how the **Restaurant** class is **intended to be used**.

Sample code usage

```
// Initialize the repository
Restaurant restaurant = new Restaurant("Casa Domingo");

// Initialize the entities
Vegetable tomato = new Vegetable("Tomato", 20);
Vegetable cucumber = new Vegetable("Cucumber", 15);

Salad salad = new Salad("Tomatoes with cucumbers");

salad.add(tomato);
salad.add(cucumber);

System.out.println(salad.getTotalCalories()); // 35
System.out.println(salad.getProductCount());  // 2

System.out.println(salad.toString());
// * Salad Tomatoes with cucumbers is 35 calories and have 2 products:
//   - Tomato have 20 calories
//   - Cucumber have 15 calories
```

```
restaurant.add(salad);

System.out.println(restaurant.buy("Invalid salad")); // false

// Initialize the second entities
Vegetable corn = new Vegetable("Corn", 90);
Salad casaDomingo = new Salad("Casa Domingo");

casaDomingo.add(tomato);
casaDomingo.add(cucumber);
casaDomingo.add(corn);

restaurant.add(casaDomingo);

System.out.println(restaurant.getHealthiestSalad()); // Tomatoes with
cucumbers

System.out.println(restaurant.generateMenu());
// Casa Domingo have 2 salads:
// * Salad Tomatoes with cucumbers is 35 calories and have 2 products:
//   - Tomato have 20 calories
//   - Cucumber have 15 calories
// * Salad Casa Domingo is 125 calories and have 3 products:
//   - Tomato have 20 calories
//   - Cucumber have 15 calories
//   - Corn have 90 calories
```

Submission

Submit **single .zip file**, containing **restaurant package, with the classes inside (Vegetable, Salad, Restaurant and the Main class**, there is no specific content required inside the Main class e. g. you can do any kind of local testing of you program there. However there should be **main(String[] args)** method inside.