Calorie Counter Catalog

Create the Product class

* Attributes:
  + Product name
  + Number of grams of fat
  + Number of grams of carbohydrates
  + Number of grams of protein
  + Total number of calories
* Methods:
  + computeCalories(valid for all objects) - The method accepts as parameters the number of grams of fats, carbohydrates and proteins. The method returns the total number of calories.

Create the ProductCatalog class

* Attributes:
  + A list of products (maximum 30 products are accepted in the list)
  + Maximum number of products = 30 (constant)
  + Number of products added to the list
* Methods:
  + printProducts - The method does not accept any parameters. The method does not return anything, it only prints the list of products in the console.
  + addProduct - The method receives a product as parameter. If the product is already in the list or if the list is already full, it will return false (operation failed). If it is not already in the list, the product will be added to the list and returned true (operation successful).
  + getProductByName - The method receives as parameter the name of a product. The method returns the product in the list that has the name equal to the name received as a parameter. If the product was not found, null will be returned.
  + deleteProduct - The method receives as parameter the name of a product. The product will be searched in the list of products by name. If the product is not found it will return false (operation failed). Otherwise, the product will be deleted from the product list and will return true (operation successful).

Tips: The operation of searching for a product by name is used in 2 places, so a private method can be created to search for the product in the list, which returns the index in the array to find the product or -1 if the product is not found.

Create the CalorieCounter class. It will not have fields, but will contain the main method. In the main method a menu of options for the user will be displayed which will be shown again after each choice performed until the user chooses the option to exit the application.

* Methods:
  + printMenu. It will display the user menu in the console
* MENU:
* "1. Add a product and calculate it’s calories"
* "2. Calculate calories for a product without adding it to the catalog"
* "3. Displays all products in the catalog and their calories"
* "4. Delete a product from the catalog"
* "5. Find product by name"
* "6. EXIT"
* "----------------------------------------------- ----------------- ”
* "Choose option: ";

After displaying the menu, read the number corresponding to the option, entered by the user in the console and the operation specific to the chosen option will be performed. Based on the input it will be decided which operation should be performed using switch instruction. This block of code will fit into a performSelectedAction method (which will be called in main method after reading the input). On each branch of the case from the switch instruction, the ProductCatalog object will be called, the method corresponding to the chosen option. To perform operations 1,3,4,5 you need an instance of the ProductCatalog class. This instance is created in the main method before any line of code and is passed as a parameter of the performSelectedAction method (along with the input from the console) in order to be used to perform the necessary operations.