Q1

Your task here is to implement a **Java** code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of classes, data fields, and methods unless mentioned otherwise.

**Specifications:**

enum definition:

enum FoodType:

VEG, NONVEG

class definitions:

class Consumer:

final String name

final int age

final FoodType foodType

visibility : private

Define a parameterized constructor **with** **public** visibility

Implment getters **with** **public** visibility

toString() method has been implemented **for** you **as** a part **of** the code stub

**class** Implementation:

getNonVegetarianList(**List**<Consumer> consumer):

**return** **type**: **List**<Consumer>

visibility: **public**

sortConsumerByAge(**List**<Consumer> consumer):

**return** **type**: **List**<Consumer>

visibility: **public**

**Task:**

**enum :**has been defined for you in the code stub

**class Consumer:**

- define the data members according to above specifications

**-**define a **constructor** and **getters**according to the above specifications

-**toString()** method has been implemented for you as a part of the code stub

class **Implementation:**

Implement the below method for this class using in **Stream API:**

* **List<Consumer> getNonVegetarianList(List<Consumer> consumer):**

fetch the details where FoodType is NONVEG, put into a list and return the list

* **List<Consumer> sortConsumerByAge(List<Consumer> consumer):**

sort the list of consumers by age and return it(in ascending order)

*Refer Sample Input Output for more details*

**Sample Input**

Implementation imp = **new** Implementation();

 Consumer p = **new** Consumer("Sarah", 45, FoodType.VEG);

 Consumer p1 = **new** Consumer("John", 26, FoodType.NONVEG);

 Consumer p2 = **new** Consumer("Mirra", 7, FoodType.NONVEG);

**List**<Consumer> consumers = Arrays.asList(p, p1, p2);

imp.getNonVegetarianList(consumers)

imp.sortConsumersByAge(consumers)

**Sample Output**

[Consumer{name='John', age=26, foodType=NONVEG}, Consumer{name='Mirra', age=7, foodType=NONVEG}]

---------------------METHOD 1------------------------------

[Consumer{name='Mirra', age=7, foodType=NONVEG}, Consumer{name='John', age=26, foodType=NONVEG}, Consumer{name='Sarah', age=45, foodType=VEG}]

---------------------METHOD 2------------------------------

**NOTE**

* You can make suitable function calls and use **the RUN CODE** button to check your **main()** method output.

**ALLOWED TECHNOLOGIES**

* Java 8

**TAGS**

* Stream API
* Lambda

**Code :**

//DOSELECT Problem Statement 1

//Class Consumer

**package** CAPG;

**import** java.util.\*;

**enum** FoodType

{

***VEG***, ***NONVEG*** ;

}

//

**class** Consumer **implements** Comparable<Consumer> {

**private** **final** String name;

**private** **final** **int** age ;

**private** **final** FoodType foodType ;

**public** Consumer(String name,**int** age,FoodType foodType)

{

**this**.age=age;

**this**.name=name;

**this**.foodType=foodType;

}

**public** String getName() {

**return** **this**.name;

}

**public** **int** getAge() {

**return** **this**.age;

}

**public** FoodType getFoodType() {

**return** **this**.foodType;

}

@Override

**public** String toString() {

**return** String.*format*("Consumer [name=" + name + ", age=" + age + ", foodType=" + foodType + "]");

}

@Override

**public** **int** compareTo(Consumer o)

{

**if**(o.getAge()>**this**.getAge())

{

**return** 1;

}

**else** **if**(o.getAge()<**this**.getAge())

{

**return** -1;

}

**return** 0;

}

}

//DOSELECT Problem Statement 1

//Class Implementation

package CAPG;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

public class Implementation<consumer> {

public List<Consumer> getNonVegetarianList(List<Consumer> consumer)

{

List<Consumer> li=new ArrayList<Consumer>();

for(int i=0;i<consumer.size();i++)

{

if((consumer.get(i).getFoodType()).equals(FoodType.NONVEG))

{

li.add(consumer.get(i));

}

}

return li;

}

public List<Consumer> sortConsumerByAge(List<Consumer> consumer)

{

Collections.sort(consumer,Collections.reverseOrder());

return consumer;

}

public static void main(String[] args)

{

Implementation imp =new Implementation();

Consumer p=new Consumer("Sarah",45, FoodType.VEG);

Consumer p1 = new Consumer("John", 26, FoodType.NONVEG);

Consumer p2 = new Consumer("Mirra", 7, FoodType.NONVEG);

List<Consumer> consumers = Arrays.asList(p, p1, p2);

System.out.println(imp.getNonVegetarianList(consumers));

System.out.println(imp.sortConsumerByAge(consumers));

}

}

**Output :**

