

# BubbleTea Recap

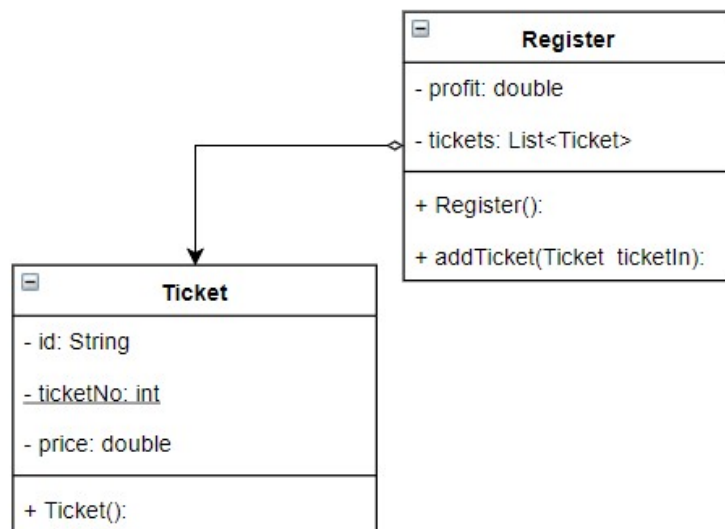
## 1) Kassa - Associatie

We gaan voor onze TheeBar een kassa bijhouden. Hierin houden we onze omzet van de dag bij en de tickets die binnenkomen houden we bij in een lijst/array van Tickets. Maak hiervoor een Register klasse aan en een klasse Ticket.

De klasse ticket bevat een statische counter en een prijs die we voor nu zelf kunnen bepalen. De id is voorlopig enkel gebaseerd op het ticketNo van

### MainV1:

Maak enkele tickets 3,5,10 euro aan en voeg ze toe aan uw register. Druk nadien uw profit af in de main.



### **Console**

Profit is 18.0

## 2) BubbleTea - Overerving

Maak het Paket BubbleTea aan Hierin zullen we onze BubbleTea en Ingredienten gaan definiëren.

Zowel de klasse Fruit, Spice, Cream en Bubble zijn afgeleid van BubbleTeaIngredients. Via deze klasse zullen we ook onze prijs gaan bijhouden. Zorg ervoor dat de subklassen via Constructor chaining ingesteld worden.

Maak De subklassen:

Fruit: Apple ,Strawberry ,Orange

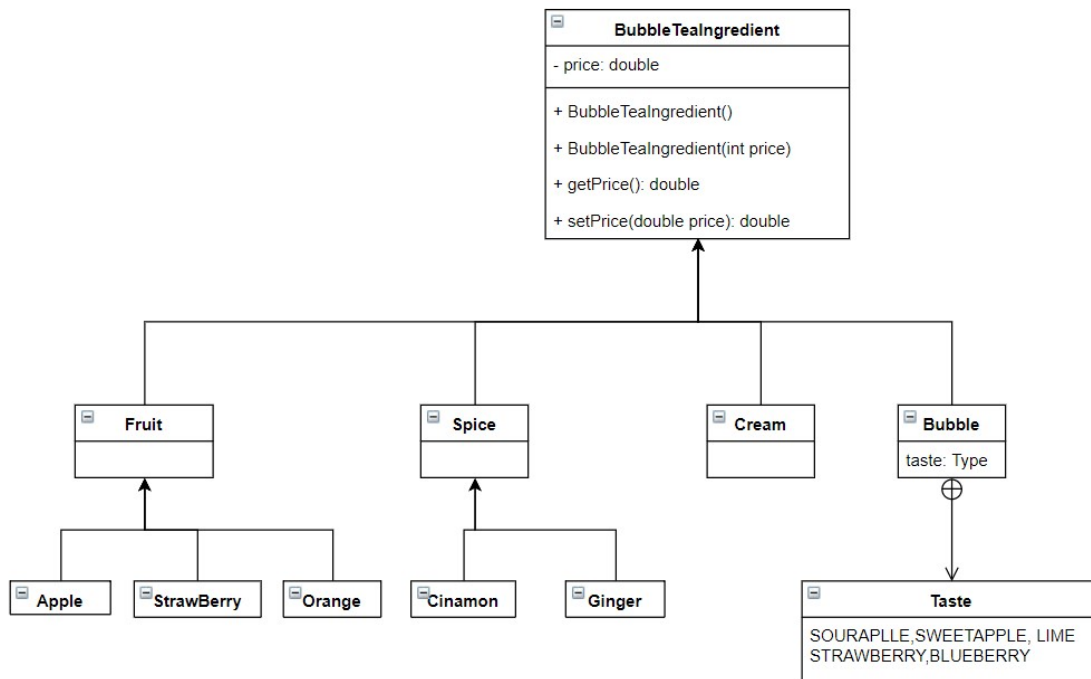
Spice: Cinamon,Ginger

Ook de Cream klasse zullen we aan de hand van de juist constructorchaining gaan definiëren

Daarnaast kunnen we aan onze smoothie ook nog een extra Bubbles toevoegen.

The extra Bubbles zijn er in verschillende smaken:

SOURAPPLE,SWEETAPPLE,REDBERRY,BLUEBERRY,LIME,STRAWBERRY



#### MainV2:

Maak 3 Ingredienten aan:

- 1.maak een Strawberry aan die 0.50 kost.
- 2.maak een Cinamon aan die 0.75 kost.
- 3.maak een LimeBubble aan die 0.50 kost

#### **Console**

```

BubbleTea.Apple prijs 2.0
BubbleTea.Cinamon prijs 0.75
LIME Bubble(1.0)prijs 1.0
    
```

### **3) BubbleTea - Abstracte klasse**

We gaan opdracht 2 refactoren door de Ingredient klasse abstract te maken en er een Abstracte double `getPrice()` in te gaan definiëren. Zo ben je zeker dat elk nieuw Ingredient dat we later zouden gaan toevoegen correct wordt aangemaakt.

Standaard prijzen(kunnen aangepast worden)

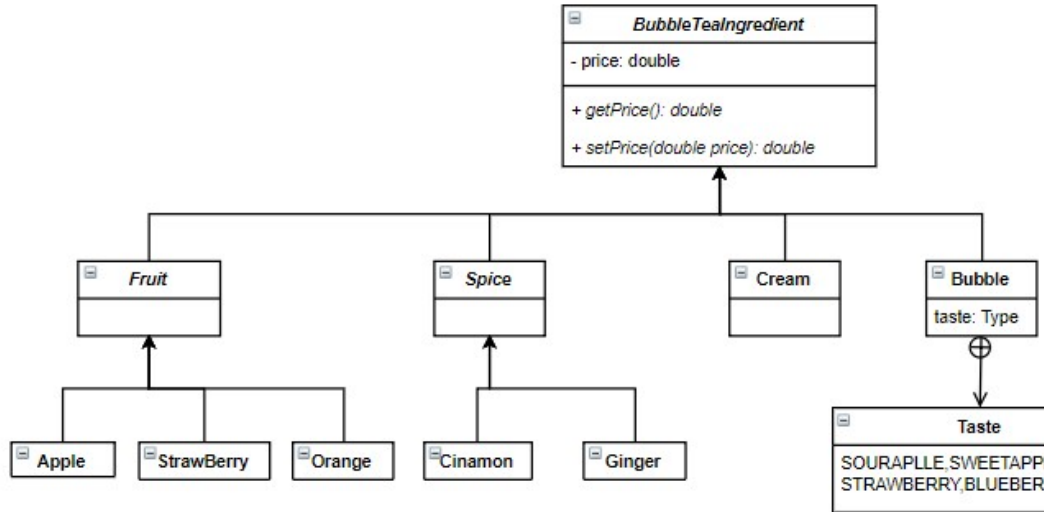
```

Apple: 0.75  Strawberry: 0.50  Orange:0.50
Ginger:1  Cardamon:0.50
Cream:1
Bubble:0.50
    
```

#### MainV3

Maak 4 Ingredienten aan:

1. maak een apple met een Contructor
2. maak Een zure apple Bubble
3. maak Strawberry die €1 kost
4. pas de Prijs aan van een reeds gemaakte Sinaas naar €0.75



### Console

```
BubbleTea.Apple prijs 1.0
BubbleTea.Cinamon prijs 0.75
LIME Bubble(0.5)prijs 0.5
```

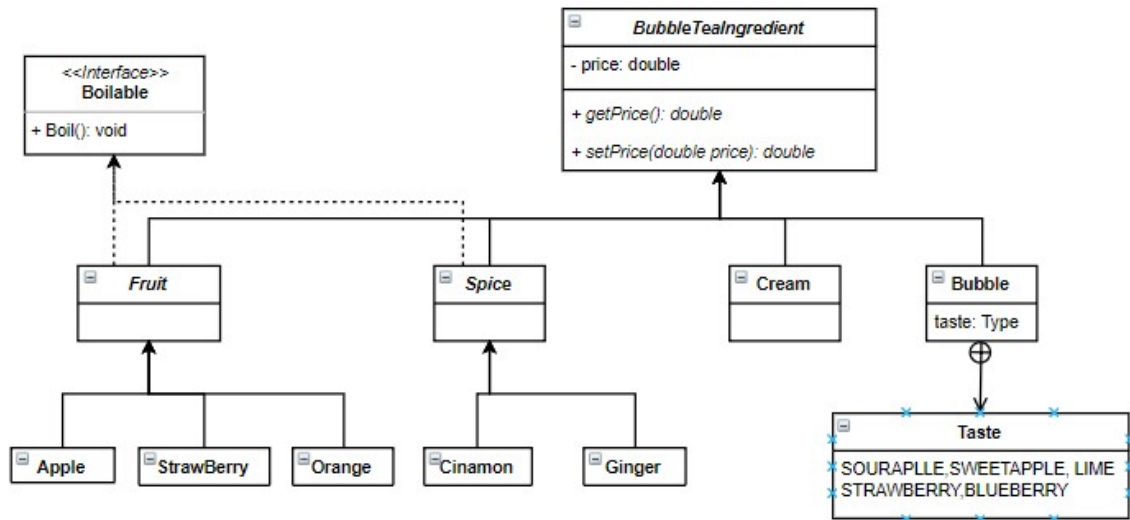
#### 4) BubbleTeaIngredient - Interface klasse

We gaan opdracht 3 uitbreiden door de Ingredienten die eerst moeten gekookt worde alvorens we ze kunnen gebruiken te gaan Definieren aan de hand van een interface klasse Boilable.

Enkel het Fruit en de Spices kunnen Gekookt worden.

#### MainV4

1. Maak terug 4 Ingredienten aan en stop ze in een array/list van Ingrediënten:
  1. maak een apple
  2. maak Een zure apple Bubble
  3. maak een Strawberry
2. Bereken van deze Array de prijs.
3. Print de ingredienten af die kunnen gekookt worden.



### Console

```

Apple(1.0)
Cinamon(0.75)
LIME Bubble(0.5)

```

```

forEach: 2.25
streamSum: 2.25

```

```

Boiling water:
    Adding Apple to the boil
    Adding Cinamon to the boil

```

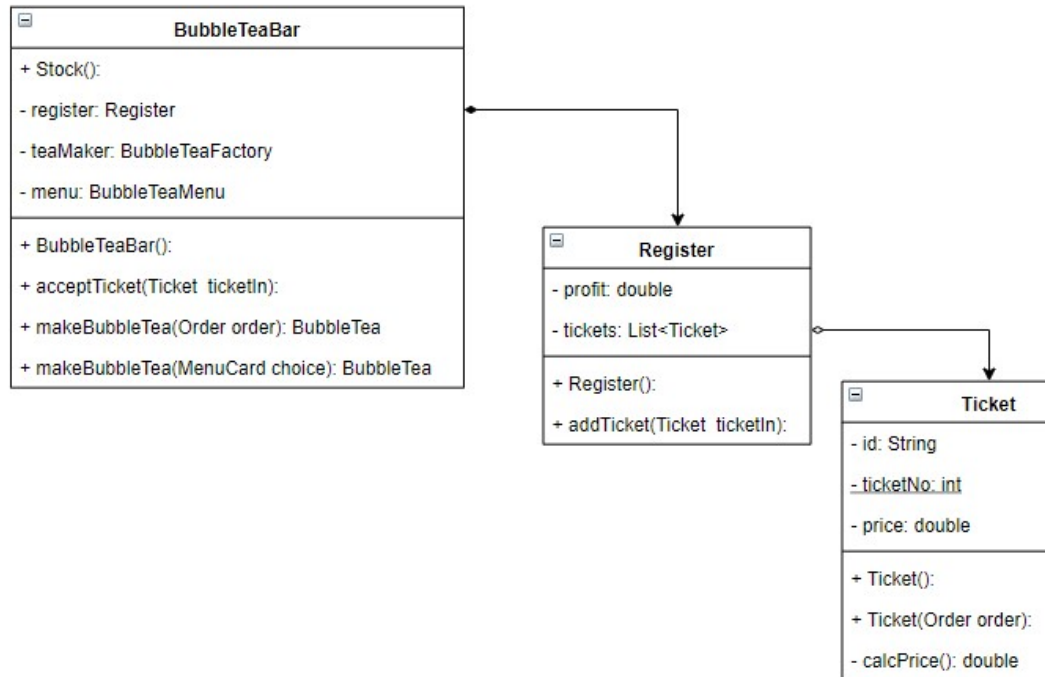
### 5) BubbleTeaBar Kassa - Main Vervanger

We gaan nu alles gaan linken en de behandeling van onze objecten buiten de main gaan laten verlopen. Enkel een instantie van de BubbleTeaBar en de tickets zullen daar nu aangemaakt worden.

Zorg ervoor dat de bar een kassa heeft. Deze zal de tickets verwerken, de winst bijhouden en later de order doorgeven aan het personeel/methode.

#### MainV5

1. Maak een Bar aan in de main.
2. Schrijf 5 tickets van 4,20 euro weg in uw kassa.
3. toon de omzet van uw Bar



### Console

-----  
Ticket zonder order price: 4.2  
-----

winst 12.600000000000001  
winst geformat 12.60

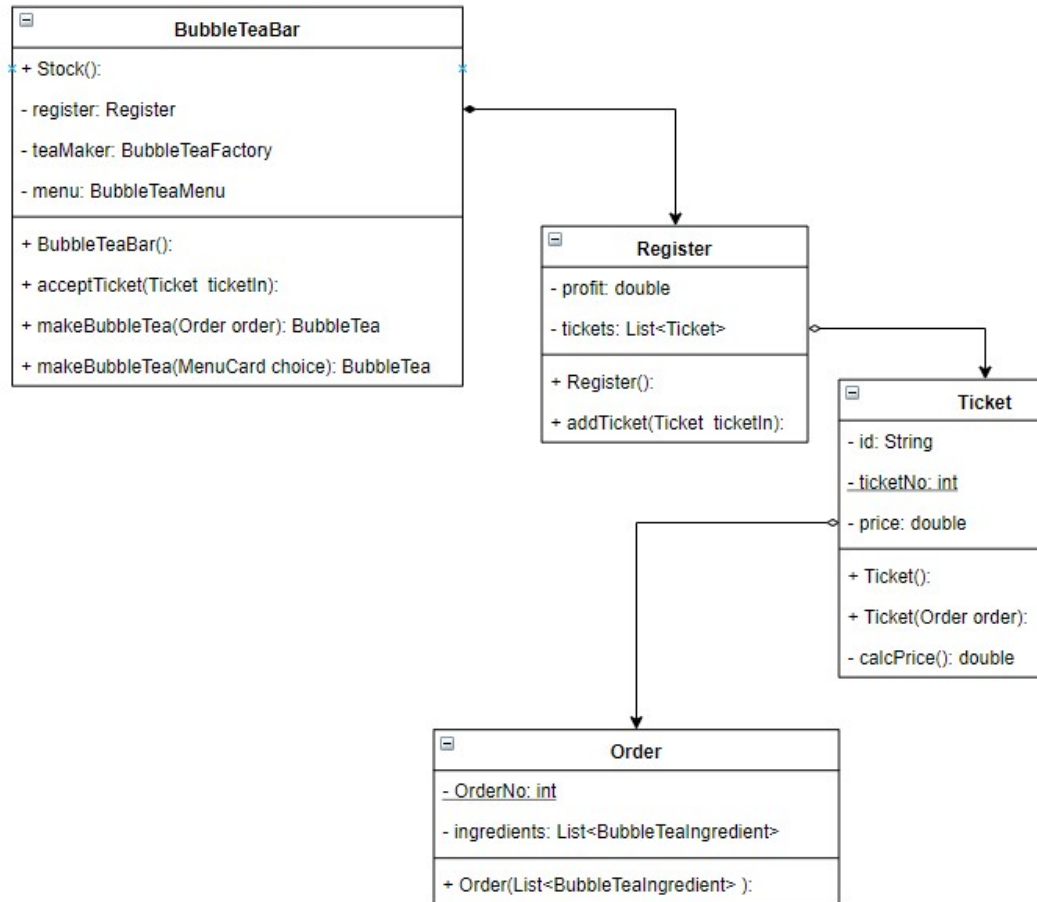
### 6) OrderHandler - passing values;

De klasse Ticket gaan we uitbreiden, zodat deze onze order kan verwerken. We gaan nu geen prijzen meer meegeven met onze tickets, maar een Order (List of Array) van Ingrediënten. Op basis van deze list zullen we dan later een BubbleTea gaan aanmaken. Maak optioneel een extra Order klasse die voor ons de lijst gaan bijhouden.

Zorg ervoor dat bij zowel het declareren als bij het zetten van de prijs van een order alles automatisch berekend wordt. en zorg er ook voor dat de klant niet meer zelf een prijs kan zetten.

### MainV6

1. Maak 3 nieuwe tickets aan met daarin een order van:
  1. AppleTea met StrawberryBubbles
  2. CinamonTea met SweetAppleBubbles en Cream
  - 3 hetzelfde als eerste order maar met cream
2. toon de prijs van elke order
3. toon de totale omzet na de drie orders



## Console

```

-----
Ticket 0
Order: 0: [Apple(1.0), STRAWBERRY Bubble(0.5)]
- starting making new order: WaterHeating up
  Adding Apple to the boil
- added STRAWBERRY Bubble(0.5) to the cup
  total price: 1.5
-----
  
```

## 7) TicketTime -LocalDateTime;

Om onze boekhouding te helpen gaan we onze Tickets gaan bijhouden met een tijdsStempel. Deze gaan we baseren op de locale tijd op het moment van het Creëren van een ticket.

Ook gaan we onze Ticket een id geven en die gaan we maken op basis ons tijds object. We gaan id laten starten met de dag van het jaar. Dan de nanoSecondne van onze bestelling met dan het huidige statische ticketNo.

### MainV7

1. Maak 3 nieuwe tickets aan.

Ticket
- id: String
- ticketTime: LocalDateTime
- ticketNo: int
- price: double
- order: Order
+ Ticket():
+ Ticket(Order order):
- calcPrice(): double

```

-----
Ticket 105_1555326495_1
Order: 0: [Apple(1.0), STRAWBERRRY Bubble(0.5)]
- starting making new order: WaterHeating up
  Adding Apple to the boil
- added STRAWBERRRY Bubble(0.5) to the cup
  total price: 1.5
-----

```

## 8) BubbleTea - Arrays/List

**Maak een Copy van uw project want we gaan veel veranderen in ons project.**

We zullen alles gaan omzetten van Arrays naar Lists veel makkelijker eens geïnitieerd. Soms wat onhandig in constructors.

Doe dit dus zowel voor

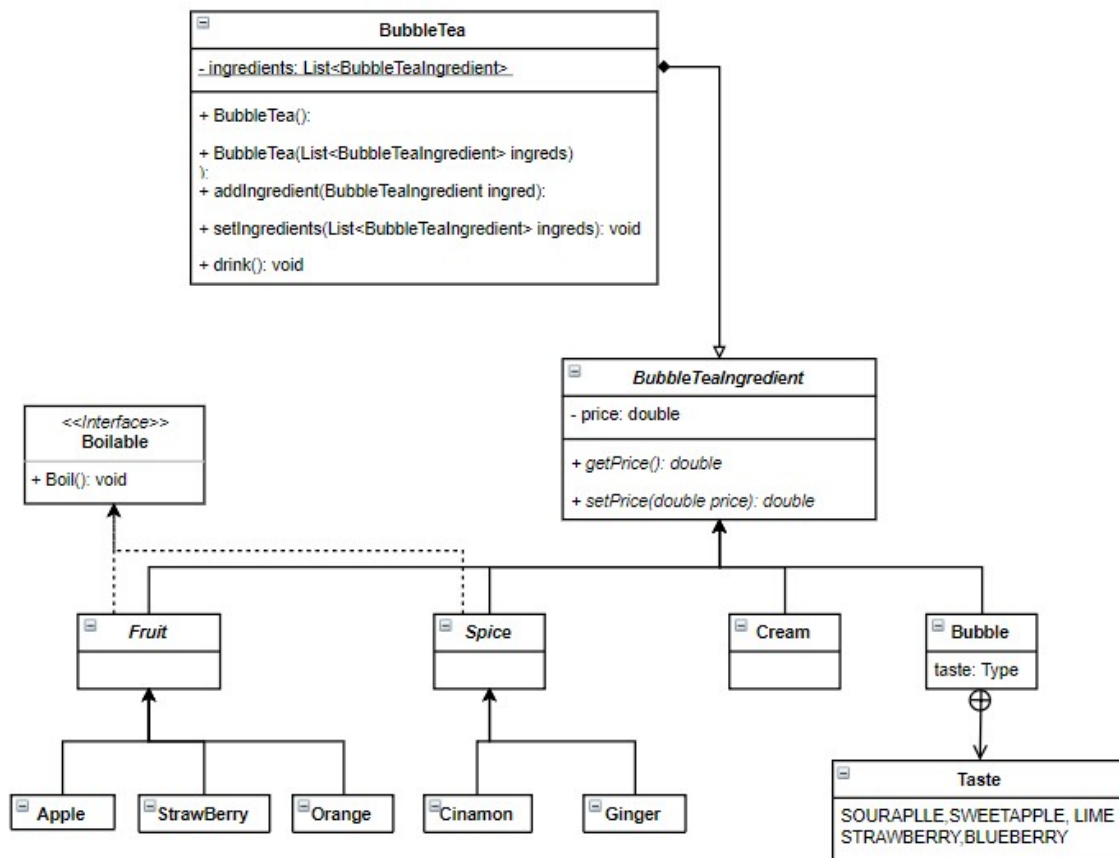
We gaan nu BubbleTeas gaan maken op basis van een List van Ingrediënten. We gaan er ook voor zorgen dat onze thee kan gedornken worden. eg hiervoor een Drink methode toe aan onze BubbleTea en roep hierin een aangepaste toString methode zoat we dat we al onze tea mooi kunnen afprinten.

BubbleTea
- ingredients: List<BubbleTeaIngredient>
+ BubbleTea():
+ BubbleTea(List<BubbleTeaIngredient> ingreds)
+ addIngredient(BubbleTeaIngredient ingred):
+ setIngredients(List<BubbleTeaIngredient> ingreds): void
+ drink(): void

## MainV8

1. Zorg ervoor dat dit kan op verschillende manieren.

1. aan de hand Constructor die een array/list accepteerd.
2. met een addIngredientMethode(BubbleTeaIngredient ingredient) die ze 1 voor 1 toevoegd
3. en met een addIngredientList die ook een array/list accepteerd.
4. Maak een Tea object in main en Drink je favoriete Tea uit het aanbod



## 9) BubbleTeaFactoryMethode- DesignPattern

We gaan nu Teas gaan maken in onze Bar met een BubbleTeaFactory methode. Zorg ervoor dat in die methode Bubbletea wordt bereidt(gekookt) en alle Ingredienten worden toegevoegd(sysout/sout) en er een nieuw bubbleTea object wordt terug gegeven. Deze methode zal werken op basis van een array/List die hij meekrijgt.

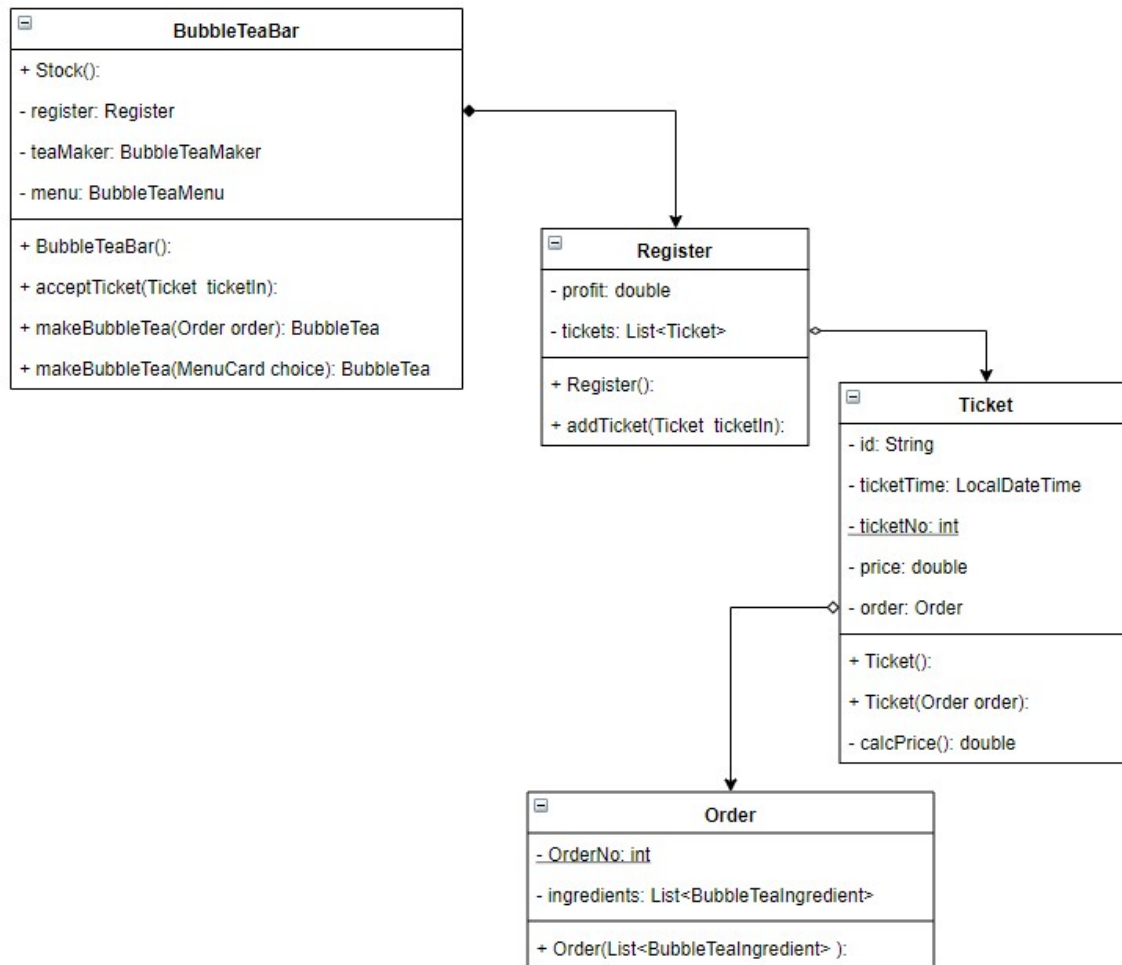
tip: Je zal dus methodes moeten toevoegen en sommige methoden nieuwe return waardes geven. Kan hiervoor misschien best een tweede BubbleTeaTruck voor aanmaken.

## MainV9

- 1 Maak 3 BubbleTea objecten in de main aande hand van uw BubbleTeaBar object:



1. Een Orange Tea met Lime Bubbles
2. Een GingerTea met Strawberry Bubbles en Cream
3. Een Strawberry Tea met StrawBerry bubbles



## 10) BubbleTeaShortage- Exceptions

We gaan ons programma nog beveiligen. We gaan ervoor zorgen dat we geen klanten laten betalen zonder dat we hun tea kunnen maken.

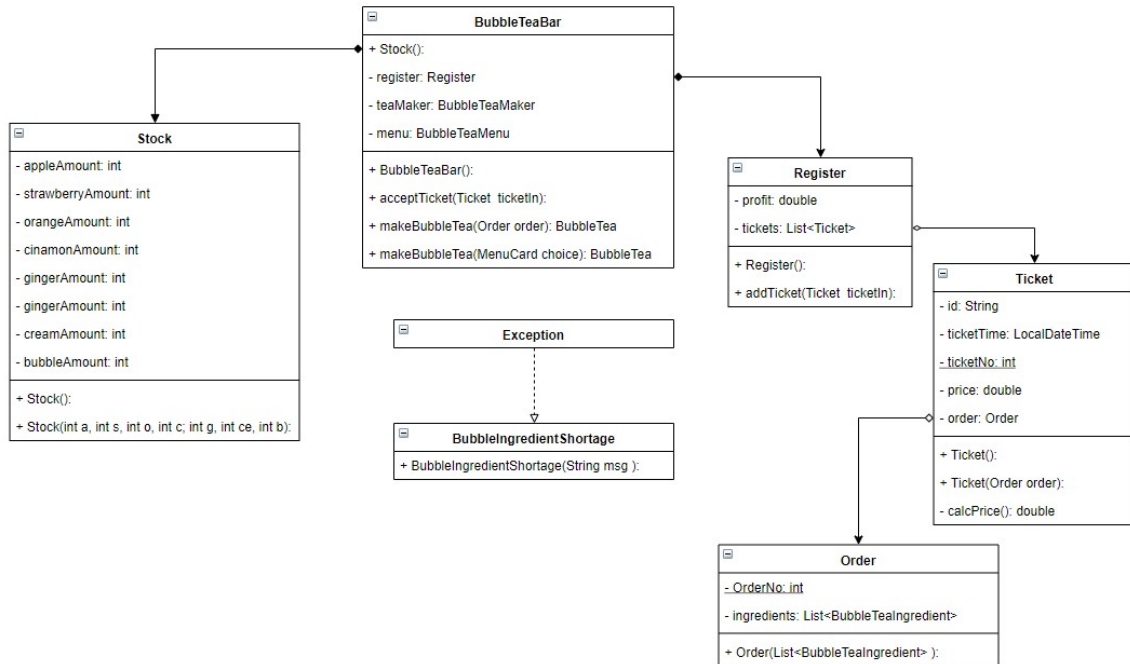
We voegen dus een stock Toe aan onze BubbleTeaBar en Zullen tijdens het maken van de Tea moeten gaan controleren als er nog genoeg stock is van elk product. Zorg ervoor dat de bar begint met een stock van 2 van elk product behalve de Bubbels en Cream daarvan hebben we er 5.

Schrijf hiervoor uw eigen Exception BubbleIngredientShortage.

### MainV10

- 1 Maak 5 BubbleTea objecten in de main aande hand van uw BubbleTeaBar object:
  1. Een Orange Tea met Lime Bubbles
  2. Een GingerTea met Strawberry Bubbles en Cream
  3. Een Strawberry Tea met SweetApple bubbles

4. Een Apple Tea met BlueBerry bubbles en Cream
5. Een Cinamon Tea met SourApple Bubbles
2. Maak nog een 6de Tea met bubbels en zorg ervoor dat de Exception opgevangen wordt
3. maak nu nog een 7de Tea zonder Bubbles.



## 11) BubbleTeaMaker -DesignPattern

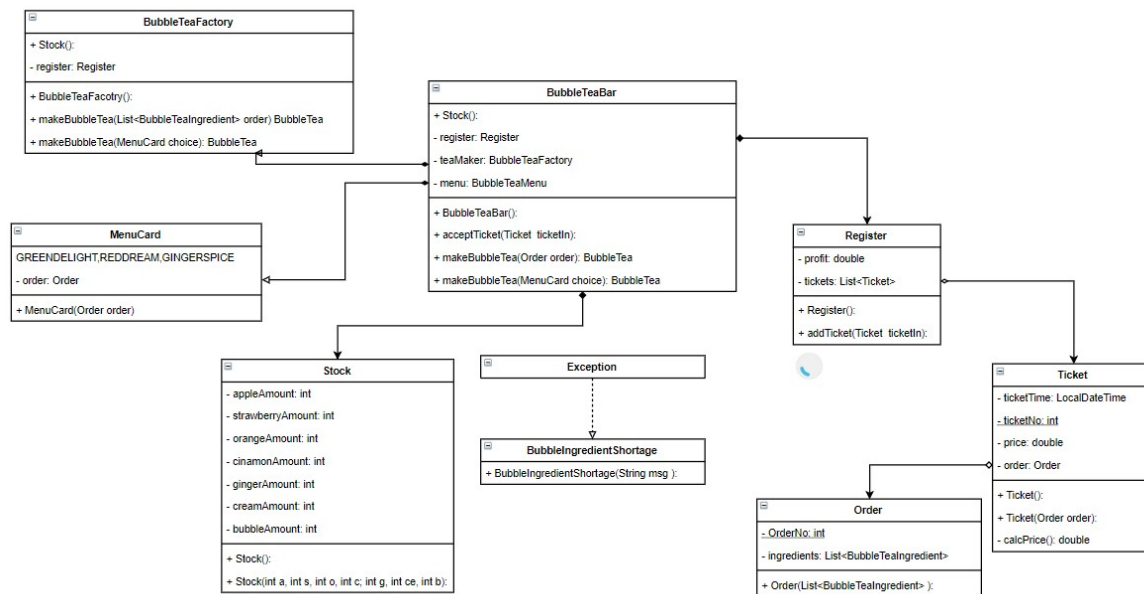
Haal de Logica uit het maken van uw Tea uit uw BubbleTeaBar en Maak een TeaFacotry aan die op basis van een Array/List of een enum uit de MenuCard Klasse ons een BuubbleTea terug geeft

Maak Hiervoor de klasse MenuCard aan en defineer enkele standaard Teas:

1. CreamyGreen: Apple Tea met zure Appel bubbels en room
2. RedDream: Aardbei thee met aarbei Bubbels
3. GingerSpice: Ginger thee met Limoen Bubbels.

### MainV11

- 1 Maak 2 BubbleTea objecten in de main aande hand van uw BubbleTeaBar object:
  - 1 Maak een order aan van een Tea met Ginger, Apple en Strawberry bubbels  
geef deze order mee aan je BubbleTeaBarMaker
  - 2 Geef een menukeuze mee aan uw bubbleTeaBar

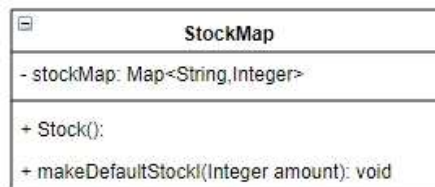


## 12) Collections Maps - StockMap

We merken nu ook nog dat stock klasse nog heel wat veel parameters bevat. Het zijn ook allemaal bijna dezelfde parameters. Deze kunnen we dus beter gaan onderbrengen in een Enum maar daar al wel genoeg van en is weer een nieuwe klasse bij. Daarom is het beter onze Stock te gaan bijhouden via een Map object waarbij onze Ingredient namen zullen dienen als key en de hoeveelheid als value.

### MainV12

- 1 Maak een stock object aan aan de hand van de Stock(Integer amount) constructor

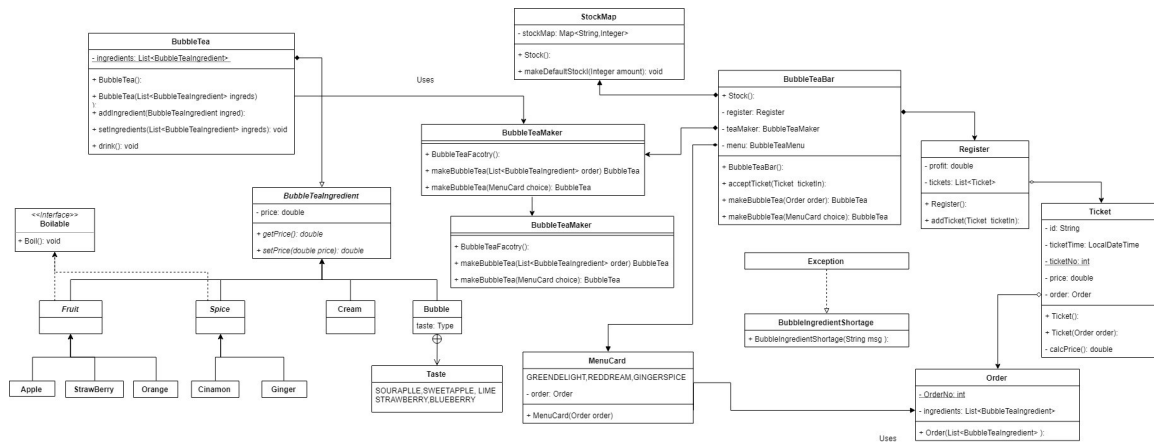


## 13) Final

Daaaaaamn wat een uml weer

### MainV13

1. Plaats 3 Bestellingen
  1. Via een Array/list object van Ingredienten die je meegeeft aan een Ticket
  2. Via een MenuKeuze
  3. Via een MenuKeuze en het toevoegen van extra Bolletjes Lime
2. druk het aantal tickets en de winst af voor de dag.



#### 14) PassionFruit Tea - anonieme klassen

Een extraatje is om tijdens het bestellen van een Tea een "Special of the day" te gaan toevoegen. We kunnen namelijk aan de hand van een lokale of een anoniemen klasse een PassionFruit tea gaan aanmaken en gaan meegeven als Order.

##### MainV14

- 1 Maak 2 BubbleTea objecten in de main aan de hand van uw BubbleTeaBar object:
  - 1 Maak een Lokaal Object en geef dit mee in de Array/List
  2. Maak een anonieme klasse aan in de Array/ die je meegeeft