

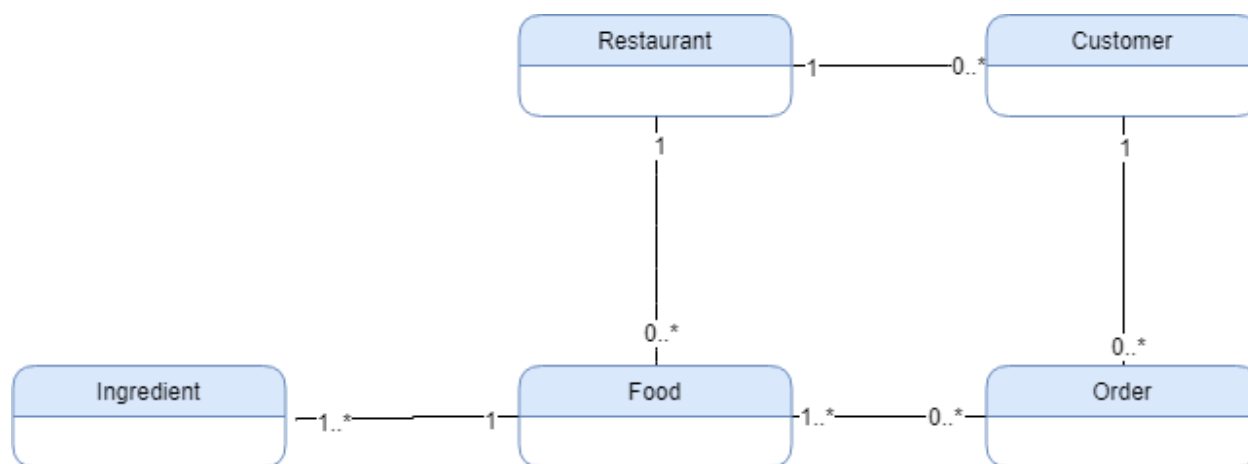
کافه رستوران جاوا

احمد و بابک دانشجوی کامپیوتر دانشگاه امیرکبیر هستند. این دو دانشجو علاوه بر اینکه در رشته خود بسیار توانمند هستند، کارهای جانبی دیگری نیز در طول زندگی خود آموخته اند (صرفاً به سری کد زن تک بعدی نیستند). احمد دست پخت خوبی دارد و بابک هم کیک و کافه‌های خوبی درست می‌کند و علاقه زیادی هم به کافه و رستوران رفتن دارند تا طمع‌های مختلف را امتحان کنند.

خلاصه، یک روز احمد به بابک می‌گوید که پدرش یک دهانه مغازه چند متر بالاتر از دانشگاه خریده است و می‌خواهد در آنجا کافه-رستوران راه بیندازد. احمد هم این فرصت را غنیمت شمارده است و مدیریت کافه-رستوران را برعهده گرفته است و می‌خواهد آنرا با کمک بابک مدیریت کند. بابک تصمیم می‌گیرد به جای کمک مالی، یک سامانه مدیریت برای رستوران بنویسد. او که در اول کار (همانند بیشتر برنامه نویسان تازه وارد) نوشتن این سامانه را ساده می‌پنداشت، سریع کد زدن را شروع کرد.

بابک بعد از مدتی دید که کارش با مشکلات زیادی مواجه شده است و اگر کارش به همین صورت پیش برود برنامه در مدت خواسته شده آماده نخواهد شد. به همین دلیل دست به دامن یکی از دوستان سال بالاتر خود شد. دوست بابک نیز از آنجایی که یک سر داشت و هزار سودا، نمی‌توانست در پیاده‌سازی این پروژه کمک کند و از طرفی نمی‌خواست روی بابک را زمین بزند. او تصمیم می‌گیرد تا ساختار برنامه تعریف کند و آنرا را به عنوان تمرین به بابک بدهد.

کار شما تکمیل سامانه شبیه‌سازی محیط رستوران و پیاده‌سازی منطقی آن است.



شکل ۱ نمودار UML رستوران

در اول کار، با مدیریت رستوران شروع می‌کنیم و رستوران را بصورت یک کلاس تعریف می‌کنیم.

کلاس رستوران:

دو constructor برای این کلاس تعریف می‌کنیم.

public Restaurant (**int** fund, String name)

که در آن fund مقدار بودجه اولیه رستوران را مشخص می‌کند و name بیانگر نام رستوران است.

public Restaurant(**int** fund)

همانند بالا، اما در این حالت رستوران نام پیش‌فرض "JAVA Coffee House" را می‌گیرد.

توابع کلاس رستوران :

public String getName()

نام رستوران را برمی‌گرداند

public String getFund()

میزان موجودی کنونی رستوران را باز می‌گرداند.

public void addFood(Food food)

یک غذا به لیست غذاهای رستوران اضافه می‌کند.

public ArrayList<Food> getFoods()

لیست غذاهای رستوران را بر اساس بیشترین تعداد سفارش برمی‌گرداند (هر غذا که بیشتر سفارش داده شده بود اول بیاید).

public void addCustomer(Customer customer)

این تابع یک مشتری جدید را به مشتریان کنونی رستوران اضافه می‌کند.

public HashSet<Customer> getCustomers()

لیست مشتریانی که از ابتدا تا کنون به رستوران آمده اند را برمی‌گرداند.

public HashSet<Customer> getCurrentCustomers()

مشتریانی که هم اکنون در رستوران هستند را برمی‌گرداند.

public void prepareOrder(Order order)

یک منو غذایی را گرفته و غذاهای داخل آنرا آماده می‌کند. در این مرحله باید به اندازه مبلغ غذاهای داخل منو از موجودی رستوران کاسته شود.

public void acceptPayment(Customer customer)

در این متد مشتری مبلغ منوی خود را پرداخت می‌کند و از رستوران خارج می‌شود.

۱۰ درصد قیمت order به آشپز داده می‌شود و بقیه آن به موجودی رستوران اضافه می‌شود.

public int getNumberOfFoodsDelivered()

تعداد غذاهای سفارش داده شده را بازمیگرداند.

کلاس مواد اولیه:

public Ingredient (**int** id, String name, **int** price)

که در آن id شناسه ماده و name اسم آن و price قیمت هر واحد از آن ماده است.

توابع این کلاس عبارتند از:

public String getName()

اسم ماده اولیه را برمی‌گرداند

public int getID()

شناسه ماده اولیه را برمی‌گرداند

public int getPrice()

قیمت ماده اولیه را برمی‌گرداند

کلاس غذا:

public Food(**int** id, String name)

id شناسه غذا را مشخص می‌کند و name نام غذا را.

public int getID()

شناسه غذا را بازمی‌گرداند

public String getName()

نام غذا را بازمی‌گرداند

public void addIngredient(Ingredient ingredient, **int** amount)

مواد اولیه را به لیست مواد اولیه غذا اضافه می‌کند.

Ingredient : ماده غذایی

Amount : تعداد واحد های مورد نیاز

public int getPrice()

قیمت غذا را بازمی‌گرداند. قیمت غذا از روی مجموع هزینه مواد اولیه برای تولید غذا حساب می‌شود.

کلاس سفارش:

این کلاس نیز دو constructor دارد

public Order (**int** id, Customer customer, ArrayList<Food> foods)

id بیانگر شناسه منو ، customer مشتری و foods لیست غذاهایی است که مشتری سفارش داده .

public Order (**int** id, Customer customer)

id بیانگر شناسه منو و customer مشتری است که غذا را سفارش داده .

public ArrayList<Food> getFoods()

لیست غذاهای سفارش را بازمی‌گرداند

public void setFoods(ArrayList<Food> foods)

لیست غذاهای سفارش را به آن اضافه میکند.

public int getPrice()

مبلغی را که باید مشتری بابت غذاهای داخل منو پرداخت کند را حساب می‌کند. هزینه غذا به علاوه ۴۰ درصد (هزینه و سود رستوران). در صورتی که جواب اعشاری شد آنرا به کوچکترین عدد صحیح نزدیک به آن رند کنید.

public Customer getCustomer()

مشتری مربوط به سفارش را بازمی‌گرداند.

```
public void addFood(Food f, int amount)
```

غذا f را به تعداد amount به لیست خرید اضافه می‌کند.

کلاس مشتری:

```
public Customer(int id, String name)
```

id شناسه مشتری را مشخص می‌کند و name نام مشتری را.

```
public int getID()
```

شناسه مشتری را بازمی‌گرداند.

```
public String getName()
```

نام مشتری را بازمی‌گرداند

```
public Order getCurrentOrder()
```

سفارش فعلی مشتری را بازمی‌گرداند.

```
public void addOrder(Order order)
```

سفارش جدید را به مشتری اختصاص می‌دهد.

همچنین برای هر یک از کلاس‌هایتان متد toString تعریف کنید.