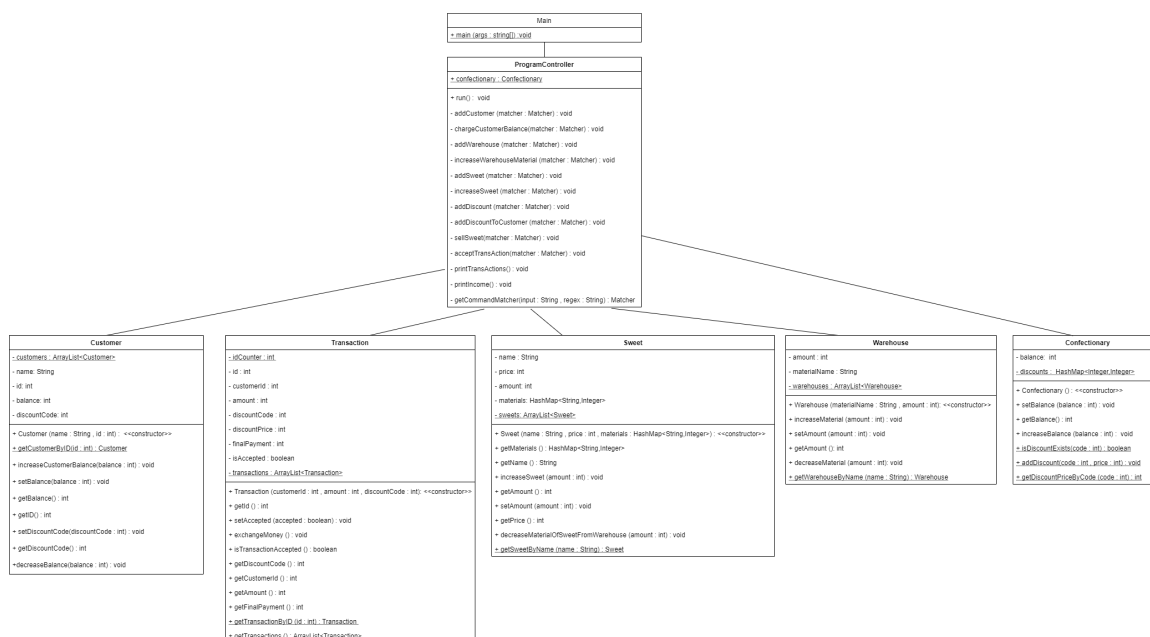




۱ قنادی حاج محمود و شرکا

حاج محمود که یکی از کسبه بازار است، در راسته شیرینی فروش‌ها یک قنادی دارد و سالیان سال است که در همین دکان کار می‌کند. او به اندازه‌ای کارش را دوست دارد که با وجود شیوع ویروس کرونا حاضر نشد درب دکان را ببندد بلکه تصمیم گرفت از این فرصت استفاده کند و برند خودش را جهانی کند. از طرفی عید نوروز فرصتی است که باید آن را غنیمت می‌شمرد، بنابراین از نوه‌اش خواست که یک سامانه فروش آنلاین برایش دست و پا کند تا به اهدافش برسد. به نوه حاج محمود کمک کنید تا این سامانه را پیاده‌سازی کند.

در این سوال، از شما می‌خواهیم که این برنامه را طبق UML Class Diagram داده شده، پیاده‌سازی کنید.





قنادی

قنادی، مقداری شیرینی دارد و نیز انبارهایی از مواد اولیه که برای پخت شیرینی از آن مواد استفاده می‌کند. هم‌چنین، یک کیف پول دارد که درآمد حاصل از فروش شیرینی‌ها به آن واریز می‌شود. علاوه بر آن‌ها، یک لیست کد تخفیف نیز دارد که می‌تواند به مشتری‌هایش کد تخفیف هدیه بدهد.

مشتری

هر مشتری یک اسم و یک شناسه یکتا دارد که از طریق آن می‌تواند موجودی حسابش را افزایش دهد.

انبار

انبار، مکانی است که در آن، مواد اولیه ذخیره می‌شوند. هر انبار تنها یک نوع مواد اولیه می‌تواند ذخیره کند. توجه کنید که انبارها با نام جنسی که ذخیره می‌کنند، شناخته می‌شوند و در نتیجه، برای هر ماده‌ی اولیه، بیش از یک انبار وجود ندارد.

شیرینی

هر شیرینی، یک اسم و یک قیمت دارد و برای پخت آن نیز به یک یا چند نوع مواد اولیه نیاز است.

تراکنش

هر تراکنش، یک شناسه، یک شناسه مشتری و مقداری پول دارد و می‌تواند تخفیف نیز داشته باشد. بعد از خرید موفق هر شیرینی توسط مشتری، یک تراکنش ساخته می‌شود که در صورت تأیید شدن این تراکنش، پول از مشتری کم شده و به حساب قنادی واریز می‌شود.



انواع عملیات ممکن

دقت کنید در صورت برخورد با هر دستور نامعتبری عبارت زیر چاپ می‌شود:

```
invalid command
```

افزودن قنادی

در اولین دستور، دستور ساخت و راه‌اندازی قنادی داده می‌شود. دقت کنید که تمام دستورهای قبل از این دستور، نامعتبر هستند و این دستور تنها یک بار به برنامه داده می‌شود.

```
create confectionary
```

افزودن یک مشتری با نام NAME و شماره شناسایی ID

```
add customer id [ID] name [NAME]
```

دقت کنید که شماره شناسایی، یک عدد طبیعی است. نام مشتری می‌تواند یک یا چند بخشی باشد؛ به عنوان مثال، اسم‌ها می‌توانند dutchman یا tir andaz khafan باشند. تضمین می‌شود که اسم مشتری تنها شامل حروف است.

مثال:

```
add customer id 21 name ali asghar
```

اگر مشتری‌ای با این شماره شناسایی، قبلاً وجود داشت، باید خطای زیر چاپ شود:

```
customer with this id already exists
```



شارژ کردن حساب مشتری (ریختن پول به مقدار AMOUNT به حساب مشتری با شناسه ID)

```
increase balance customer [ID] amount [AMOUNT]
```

تضمین می شود مقدار واریزی عددی طبیعی و مثبت است.

مثال:

```
increase balance customer 21 amount 25000
```

اگر مشتری ای با این شماره شناسایی وجود نداشت، باید خطای زیر چاپ شود:

```
customer not found
```

افزودن انبار با مواد اولیه MATERIAL NAME و مقدار AMOUNT واحد

```
add warehouse material [MATERIAL NAME] amount [AMOUNT]
```

دقت کنید که نام مواد اولیه می تواند یک یا چند بخشی باشد؛ به عنوان مثال، اسم ها می توانند konjed یا roghan oila باشند. تضمین می شود اسم مواد اولیه، تنها شامل حروف است.

مثال:

```
add warehouse material Sugar amount 50
```

اگر انباری با این مواد اولیه قبلا وجود داشت، باید خطای زیر چاپ شود:

```
warehouse having this material already exists
```



افزایش مقدار مواد اولیه یک انبار با مواد اولیه MATERIAL NAME و مقدار AMOUNT

```
increase warehouse material [MATERIAL NAME] amount [AMOUNT]
```

مثال:

```
increase warehouse material Oil amount 10
```

اگر انباری با این مواد اولیه وجود نداشته باشد، باید خطای زیر چاپ شود:

```
warehouse not found
```

افزودن شیرینی (که تاکنون وجود نداشته است) با اسم NAME (که یکتاست)، قیمت PRICE، یک یا چند مواد اولیه MATERIAL NAME و مقدار AMOUNT مورد نیاز آن مواد اولیه برای ساخت شیرینی به فرمت زیر دقت کنید (موارد مختلف با کاما جدا شده‌اند):

```
add sweet name [NAME] price [PRICE] materials: [MATERIAL] [AMOUNT],  
[MATERIAL] [AMOUNT], [MATERIAL] [AMOUNT], ...
```

دقت کنید که نام شیرینی می‌تواند یک یا چند بخشی باشد؛ به عنوان مثال، اسم‌ها می‌توانند khome یا cake yazdi باشند. تضمین می‌شود که اسم شیرینی‌ها تنها شامل حروف است، مقادیر مواد اولیه مورد نیاز اعدادی طبیعی هستند و برای هر شیرینی تنها یک بار این دستور داده می‌شود.

مثال:

```
add sweet name Cake price 2500 materials: khome 10, sugar 100
```

توجه: هر شیرینی‌ای که با این دستور اضافه می‌شود، تعداد موجود آن در قنادی صفر است که برای اضافه کردن این مقدار، از دستور بعدی استفاده می‌شود.



خطاها هنگام ساخت شیرینی برای اولین بار

- اگر انبار مربوط به آن ماده (ها) وجود نداشت، خطای زیر نمایش داده می شود:

not found warehouse(s): [MATERIAL] [MATERIAL] ...

دقت کنید که هر کدام از از موادی که انبارش وجود نداشت، باید در این خطا به همان ترتیبی که در دستور اضافه کردن شیرینی وارد شده است، نمایش داده شود؛ به عنوان مثال، اگر انبارهای مواد اولیه برای ساخت کیک (خامه، شکر) وجود نداشت، خطای زیر چاپ می شود:

not found warehouse(s): khamé sugar

اضافه کردن شیرینی با نام NAME به تعداد AMOUNT به موجودی فروشگاه (دقت کنید که هنگام افزایش شیرینی باید به همان مقدار AMOUNT از مواد اولیه مورد نیازش استفاده شود)

increase sweet [NAME] amount [AMOUNT]

با اجرای این دستور، تعداد شیرینی های NAME به مقدار AMOUNT که عددی طبیعی است، افزایش می یابد.

اگر هنگام افزایش موجودی شیرینی، آن شیرینی از قبل وجود نداشته باشد، خطای زیر نمایش داده می شود:

sweet not found

اگر در انبارهای مربوطه، مقدار مورد نیاز موجود نبود، خطای زیر نمایش داده می شود:

insufficient material(s): [MATERIAL] [MATERIAL] ...



افزودن کد تخفیف به شماره CODE و مقدار قیمت PRICE به فروشگاه

```
add discount code [CODE] price [PRICE]
```

مثال:

```
add discount code 3223 price 10000
```

توجه کنید که شماره و مقدار کد تخفیف، اعدادی طبیعی هستند.

اگر کد تخفیف از قبل وجود داشت خطای زیر چاپ می‌شود:

```
discount with this code already exists
```

افزودن کد تخفیف به شماره CODE برای مشتری به شماره شناسایی ID

```
add discount code code [CODE] to customer id [ID]
```

تضمین می‌شود که هر مشتری در لحظه فقط می‌تواند یک کد تخفیف داشته باشد.

اگر کد تخفیفی با این شماره در لیست کدهای تخفیف فروشگاه وجود نداشت، خطای زیر نمایش داده شود:

```
discount code not found
```

اگر مشتری‌ای با این شماره شناسایی وجود نداشت، خطای زیر نمایش داده شود:

```
customer not found
```

دقت کنید که خطاها به همین ترتیب چک شوند. در صورت برخورد با یک خطا، نیازی به چک کردن بقیه‌ی خطاها نیست.

فروش شیرینی با نام NAME به مقدار AMOUNT به مشتری با شناسه ID

```
sell sweet [NAME] amount [AMOUNT] to customer [ID]
```



مثال:

sell sweet cake amount 3 to customer 2

اگر شیرینی ای با این نام وجود نداشت، خطای زیر نمایش داده شود:

sweet not found

اگر مقدار این شیرینی در فروشگاه کافی نبود، خطای زیر نمایش داده شود:

insufficient sweet

اگر مشتری ای با این شماره شناسایی وجود نداشت، خطای زیر نمایش داده شود:

customer not found

اگر مشتری پول کافی برای خرید این مقدار شیرینی نداشت، خطای زیر نمایش داده شود:

customer has not enough money

دقت کنید که خطاها به همین ترتیب چک شوند. در صورت برخورد با یک خطا، نیازی به چک کردن بقیه خطاها نیست.

در صورتی که هیچ خطایی وجود نداشته باشد، یک تراکنش ایجاد و پیام زیر نمایش داده می شود:

transaction [ID] successfully created

دقت کنید که اولین تراکنش ساخته شده، با شماره ۱ ساخته می شود و تراکنش های بعدی نسبت به آن شماره گذاری می شوند؛ مثلاً، شماره چهارمین تراکنش ساخته شده، ۴ است. همچنین، اگر مشتری کد تخفیف داشته باشد، این کد در تراکنش لحاظ شده و کد تخفیف مشتری باطل می شود. اگر مشتری کد تخفیف نداشته باشد، به جای آن مقدار ۱- ثبت می شود.



قبول کردن تراکنش با شماره ID

```
accept transaction [ID]
```

اگر تراکنشی با این شماره وجود نداشته باشد یا قبلاً قبول شده باشد، خطای زیر نمایش داده شود:

```
no waiting transaction with this id was found
```

بعد از اجرای موفق تراکنش، پول از حساب مشتری کم شده و به حساب قنادی ریخته می‌شود. اگر مشتری کد تخفیف داشته باشد، در این عملیات، مقدار آن تخفیف از کل پول کم شده و سپس مقدار باقی مانده بعد از تخفیف از حساب مشتری کم شده و به حساب قنادی ریخته می‌شود.

چاپ کردن لیست تراکنش‌ها

```
print transactions list
```

با اجرای این دستور، تمام تراکنش‌هایی که قبول شده اند به ترتیب از اولین تراکنش ساخته شده، با ترتیب زیر در چاپ مشخصات‌شان، چاپ می‌شوند:

شماره تراکنش (ID)، شماره شناسایی مشتری (customer id)، مقدار کل پول حاصل از فروش آن تعداد شیرینی خریده شده ($\text{payment amount} = \text{sweet price} * \text{amount}$)، کد تخفیف (discount code) و مقدار نهایی بعد از اعمال تخفیف (final price).

```
transaction [ID]: [customer id] [payment amount] [discount code] [final price]
```

چاپ کردن پول فروشگاه

```
print income
```

با اجرای این دستور، مقدار پولی که در قنادی از فروش شیرینی‌ها به دست آمده است، چاپ می‌شود.



خروج از برنامه

در انتها، با وارد کردن عبارت زیر، اجرای برنامه پایان می‌یابد:

```
end
```