

به نام خدا

## تمرین سوم درس برنامه‌نویسی پیشرفته

نیم‌سال دوم ۱۴۰۱-۱۴۰۰

۱. تمامی فایل‌های کد را به همراه فایل متنی که در قالب pdf است (مورد سوم را بخوانید) به صورت یک فایل آرشیو zip (zip != rar) که به قالب زیر نام‌گذاری شده است، بارگذاری نمایید.

AP-HW3-FirstName\_LastName-StudentNumber.zip

۲. در سوال‌هایی که ورودی و خروجی مطلوب آن‌ها مشخص شده است، رعایت نحوه ورودی گرفتن و نمایش خروجی اهمیت بسیاری دارد. دقیقاً همان‌طور که از شما خواسته شده است ورودی‌ها را خوانده و خروجی‌ها را تولید کنید. (سوالات دستی تصحیح می‌شوند اما لازم است که از فرمت ذکر شده تبعیت نمایید)

۳. پاسخ سوالات تشریحی را به صورت تایپ‌شده و در قالب یک فایل pdf (برای کل تمرین) تحویل دهید.

۴. در صورت مشاهده هرگونه تقلبی، طبق موارد گفته شده در قوانین درس برخورد خواهد شد.

۵. در صورت وجود هرگونه ابهام می‌توانید از طریق آدرس ایمیل «ap.winter2022@gmail.com» با تدریس‌یاران در ارتباط باشید.

۶. امکان آپلود تا دو روز پس از ددلاین و با ضریب‌های ۷۵ و ۵۰ درصد امکان‌پذیر است.

مهلت تحویل: تا جمعه ۲۷ اسفند ۱۴۰۰ ساعت ۲۳:۵۹ شب

صفحه

فهرست سوالات

سوال اول..... ۳

سوال دوم..... ۴

سوال سوم..... ۶

سوال چهارم..... ۱۲

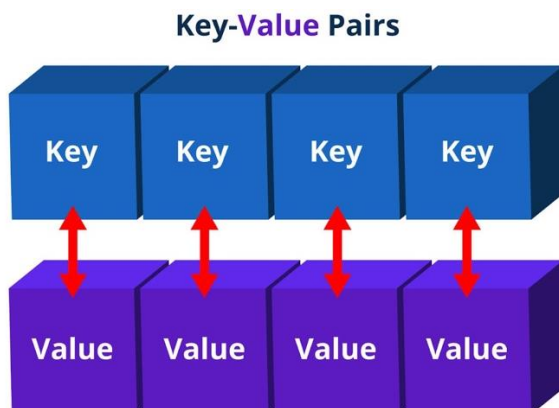
## سوال اول

الف) درستی یا نادرستی عبارات زیر را مشخص کرده و توضیح دهید.

۱. یک فیلد final را باید مقداردهی کنیم و گرنه با خطای کامپایل مواجه می‌شویم.
  ۲. یک map میتواند کلید تکراری داشته باشد.
  - ب) جاهای خالی را با عبارت مناسب پر کنید.
  ۱. Hashset و Hashmap نمی‌توانند داده با Typeهای ..... را ذخیره کنند.
  ۲. .... نوع خاصی از کالکشن است که می‌تواند عناصر غیر تکراری را ذخیره کند.
  ۳. اگر یک شیء را بدون نام تعریف کنیم و مستقیم به متدی که از آن استفاده می‌کند پاس دهیم، یک شیء ..... تعریف کرده‌ایم.
  ۴. کلاس جاوا ..... قابلیت‌های ساختمان داده‌های array-like را ارائه می‌کند و می‌تواند به صورت داینامیک resize شود.
  ۵. اگر یک بلوک ..... در کلاسی داشته باشیم، اولین باری که از آن کلاس یک شیء می‌سازیم یا اولین باری که به یک عضو static از آن کلاس دسترسی پیدا می‌کنیم، آن بلوک اجرا می‌شود.
- (امتیازی)

ج) به سوالات زیر پاسخ دهید.

۱. تفاوت‌های ArrayList و LinkedList در جاوا چیست؟
۲. مفهوم overloading را توضیح داده و حالت‌های مختلف overload کردن را نام ببرید.
۳. چرا نمی‌توان اعضای یک لیست را با foreach حذف کرد؟ روش مناسب برای انجام این کار چیست؟
۴. در جاوا برای ذخیره داده‌ها به شکل زوج‌های دوتایی (key, value)، از کالکشن HashMap استفاده می‌شود که مانند شکل زیر است:



۳\_ روش برای پیمایش اعضای این کالکشن نام برده و به صورت خلاصه توضیح دهید.

## سوال دوم

در هر یک از سناریوهای زیر، می‌خواهیم تعدادی شیء را در کنار یکدیگر نگهداری کنیم. بسته به نیازمندی‌های ذکر شده در هر سناریو، با ذکر دلیل مشخص کنید که استفاده از کدام یک از کلاس‌های `ArrayList`، `LinkedList`، `HashMap`، `HashSet` و یا استفاده از آرایه‌ای از اشیاء برای این کار مناسب‌تر است:

الف) می‌خواهیم لیست کاربران آنلاین را در سامانه کورسز نگهداری کنیم. توجه داشته باشید که نیازی به جستجو میان این کاربران نداریم و تنها برای چاپ کردن لیست کاربران آنلاین، به ترتیب آنلاین شدن آن‌ها، از این لیست استفاده می‌کنیم.

ب) می‌خواهیم لیست دانشجویانی که در ترم اخیر حداقل در یک درس نمره قبولی کسب نکرده‌اند را داشته باشیم، به طوری که اسم هر نفر حداکثر یکبار در این لیست ظاهر شود.

ج) می‌خواهیم دانشجویان دانشگاه امیرکبیر را در یک مجموعه ذخیره کنیم. تمام اطلاعات دانشجویان از جمله نام‌کاربری و رمزعبور آن‌ها در این کلاس ذخیره می‌شوند. بنابراین، هر زمان که دانشجویی می‌خواهد به سامانه دانشگاه وارد شود، باید این لیست را جستجو کنیم.

د) در یک سیستم با حافظه محدود، می‌خواهیم اطلاعات دانشجویان تعداد زیادی کلاس را ذخیره کنیم. در ابتدای ترم حداکثر ظرفیت هر کلاس را می‌دانیم و می‌دانیم ممکن است تعدادی از دانشجویان از این کلاس انصراف دهند، اما مطمئن هستیم پس از ساخت کلاس تعداد دانشجویان اضافه نمی‌شود.

ه) در یک سیستم دانشگاهی به طور مداوم نیاز است که بتوانیم با جستجو میان شماره‌های دانشجویی، به سرعت دانشجوی مورد نظر را پیدا کرده و به اطلاعات او دسترسی داشته باشیم.

## سوال سوم

در این سوال هدف ما پیاده‌سازی یک گراف جهت‌دار است. گراف یک ساختمان داده برای ذخیره داده‌های متصل به هم است. به عبارت دیگر یک گراف از مجموعه‌ای از راس‌ها و یال‌هایی که راس‌ها را به یکدیگر متصل می‌کنند تشکیل شده است. برای آشنایی بیشتر با مفهوم گراف می‌توانید لینک زیر را مطالعه کنید:

[https://www.tutorialspoint.com/data\\_structures\\_algorithms/graph\\_data\\_structure.htm](https://www.tutorialspoint.com/data_structures_algorithms/graph_data_structure.htm)

در این جا ما یک کلاس به نام Graph داریم که شامل یک Hashmap است که یک راس را به لیستی از رئوس همسایه آن مپ می‌کند.

```
HashMap<Integer, ArrayList<Integer>>
```

این کلاس شامل متدهایی است که می‌توانند اعمال زیر را انجام دهند:

```
void addVertex (int vertex){}
```

این متد یک راس را می‌گیرد و آن را به گراف اضافه می‌کند.

شکل ورودی:

```
addVertex k
```

```
void addEdge (int source, int destination) {}
```

این متد راس مبدا و مقصد را می‌گیرد و راس مقصد را به لیست همسایه‌های راس مبدا اضافه می‌کند.

شکل ورودی:

```
addEdge s d
```

```
void removeEdge (int source, int destination) {}
```

این متد راس مبدا و مقصد را می‌گیرد و راس مقصد را از لیست همسایه‌های راس مبدا حذف می‌کند.

شکل ورودی:

```
removeEdge s d
```

```
void printGraph(){}
```

این متد رئوس را به صورت مرتب شده به همراه همسایه‌های آن‌ها چاپ می‌کند.

شکل ورودی:

```
print
```

نمونه ورودی و خروجی به شکل زیر است:

با وارد کردن دستور exit برنامه خاتمه می‌یابد.

شکل ورودی:

```
exit
```

Input:

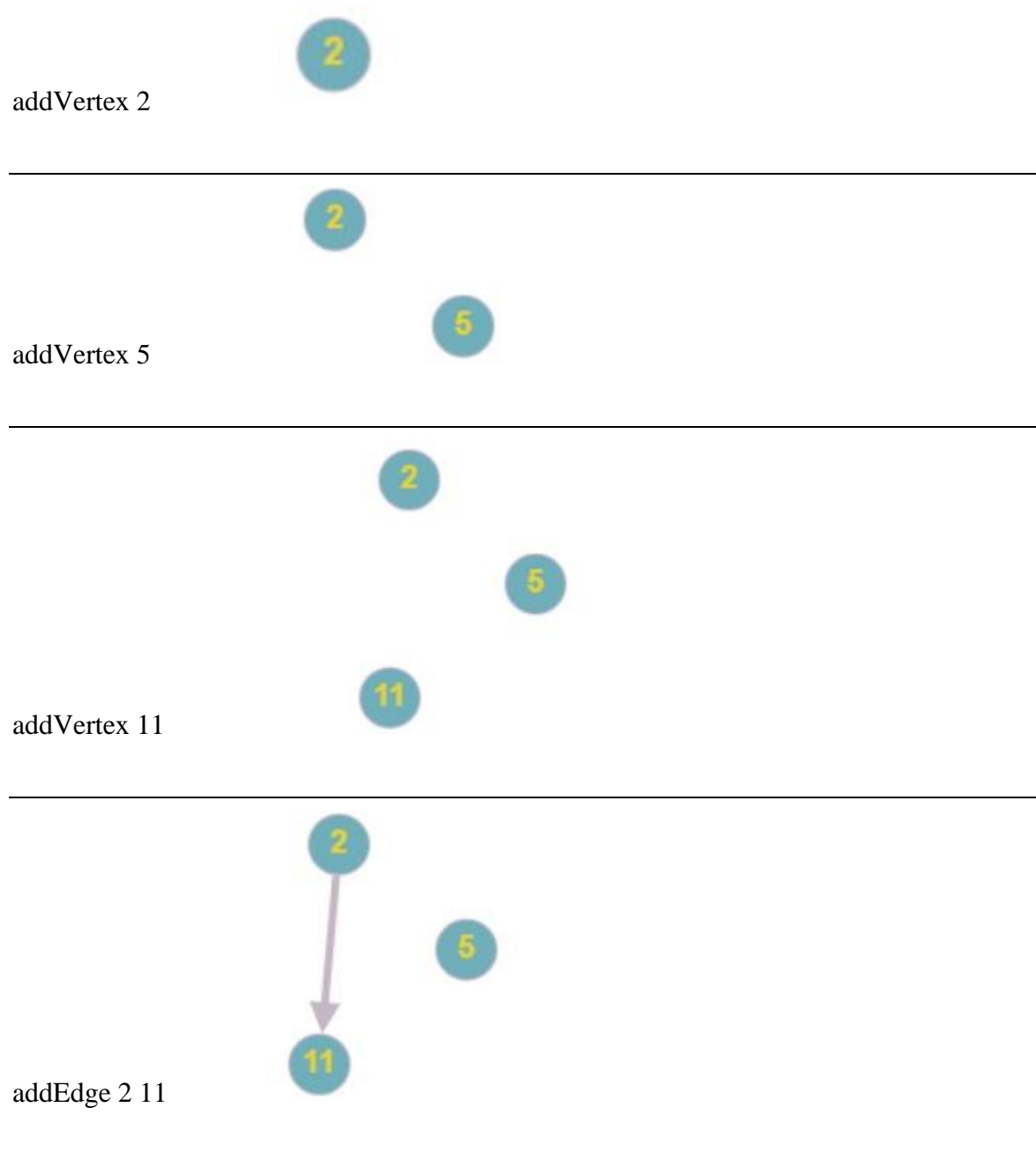
```
addVertex 2
addVertex 5
addVertex 11
addEdge 2 11
addVertex 25
addEdge 5 11
addEdge 11 2
print
removeEdge 2 11
print
addVertex 20
addEdge 2 20
addEdge 20 25
removeEdge 5 11
print
exit
```

output:

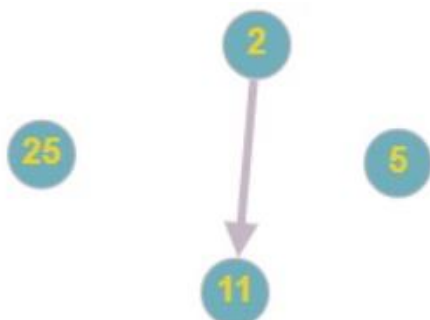
```
Graph {
2 : 11
5 : 11
11 : 2
25 :
}
Graph {
2 :
5 : 11
11 : 2
25 :
}
Graph {
2 : 20
5 :
11 : 2
20 : 25
25 :
}
```



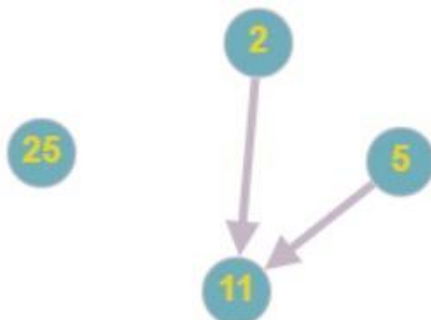
توضیح ساخته شدن گراف با توجه به نمونه ورودی که در بالا ذکر شد:



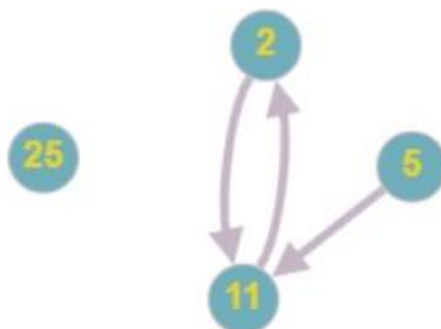
addVertex 25



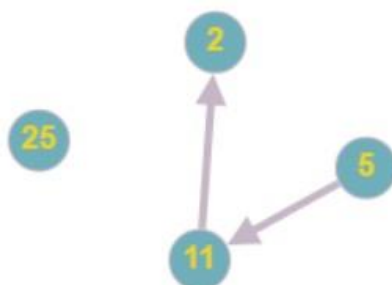
addEdge 5 11

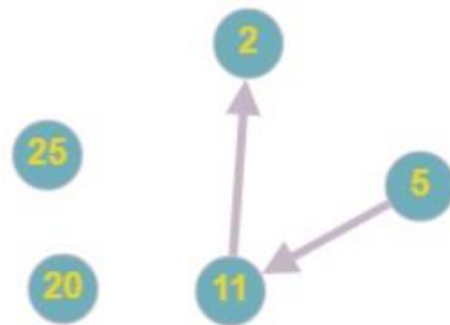


addEdge 11 2



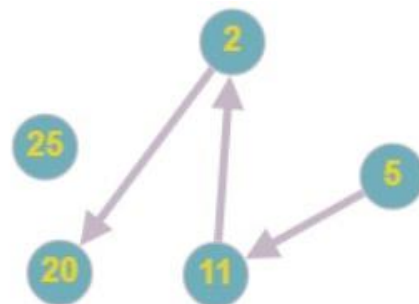
removeEdge 2 11





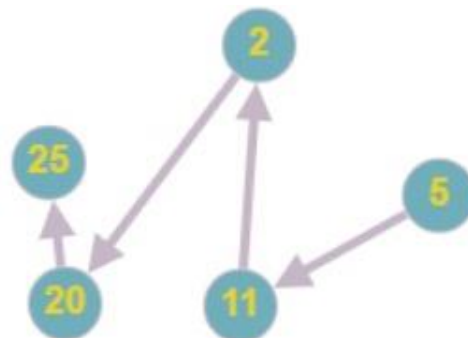
addVertex 20

---



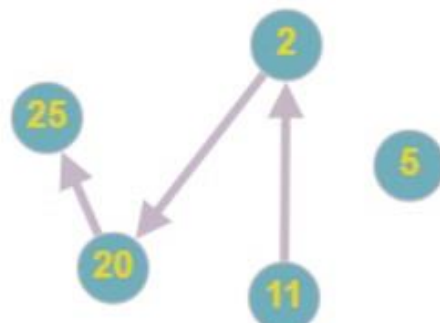
addEdge 2 20

---



addEdge 20 25

---



removeEdge 5 11

## سوال چهارم

در این سوال می‌خواهیم یک رستوران پیاده‌سازی کنیم که در ادامه توضیح کلاس‌های آن آمده است. در صورت نیاز می‌توانید متدهایی اضافه کنید که Signature آن را خودتان مشخص می‌کنید.

### • کلاس Person:

اطلاعات هر فرد مانند نام، نام کاربری، موجودی پول و سبد خرید کاربر را در فیلدهای خود ذخیره می‌کند.

- نام دو فرد می‌تواند یکسان باشد اما نام کاربری یکتا است.
- موجودی کاربر می‌تواند عددی اعشاری باشد.
- سبد خرید، یک شی از کلاس Basket است.

### • کلاس Product:

اطلاعات هر محصول از جمله نام، دسته بندی، قیمت، تاریخ و ساعت تولید و تاریخ و ساعت انقضا را در فیلدهای خود ذخیره می‌کند.

- نام یکتا است.

- تاریخ تولید و انقضا باید از کلاس LocalDateTime باشند.

- فرمت تاریخ باید به صورت yyyy-MM-dd HH:mm باشد.

- نباید تاریخ تولید بعد از تاریخ انقضا باشد.

- قیمت محصولات می‌تواند عددی اعشاری باشد.

- دسته بندی تنها شامل سه نوع می‌شود و انواع دیگر قابل پذیرش نیست:

Drink -

Food -

Snack -

### • کلاس Inventory:

یک HashMap دارد و تمامی محصولات موجود در رستوران را به تعداد موجودی آن محصول map می‌کند. این کلاس باید توانایی کم و اضافه کردن محصولات جدید به انبار را داشته باشد و همچنین بتواند میزان موجودی یک محصول که قبلاً به فروشگاه اضافه شده است را تغییر دهد.

### • کلاس Basket:

سبد خرید شما است و باید قابلیت کم و اضافه کردن محصولات به سبد خرید را داشته باشد و همچنین مجموع قیمت محصولات داخل سبد خرید را نیز محاسبه کند.

• در پیاده‌سازی این کلاس از ArrayList استفاده کنید.

کلاس‌های Product، Person، Inventory و Basket را پیاده‌سازی کنید. همه کلاس‌ها باید متد toString را override کنند و در صورت نیاز با صدا زدن این متد اطلاعات مورد نیاز را چاپ کنند.

### • کلاس Store:

تقریباً تمام پردازش‌ها و مدیریت منابع در این کلاس انجام می‌شود. به نوعی این کلاس، مدیریت رستوران شما را بر عهده دارد.

• یک Instance از کلاس Inventory با نام inventory دارد.

• یک HashSet از کاربران دارد.

### • کلاس Main:

در این کلاس، تعامل با کاربر صورت می‌گیرد. تمام دستورات قابل فهم در این کلاس قرار دارد. تمام اطلاعات از کاربر گرفته می‌شوند و برای انجام پردازش به کلاس Store داده می‌شوند.

• با هربار شروع برنامه باید محصولات زیر را با اطلاعات داده شده مربوط به هر یک بسازید و همه آنها را با موجودی داده شده در HashMap موجود در Inventory ذخیره کنید.

Name	Short Name	Category	Price	Manufacture	Expiration	In Stock
Sheldon's Pizza	Pizza	Food	\$19.99	Apr 5, 2022 09:00 PM	Apr 9, 2022 09:00 PM	10
Tom & Jerries Steak	Steak	Food	\$100.00	Mar 6, 2022 09:30 PM	Mar 9, 2022 12:00 AM	5
Sponge Bob's Burger	Burger	Food	\$15.75	Mar 3, 2022 08:30 AM	Mar 3, 2022 08:30 PM	100
Central Perk's Coffee	Coffee	Drink	\$8.50	Mar 1, 2022 01:20 PM	Mar 1, 2022 03:20 PM	50
Ninja Turtles Chocolate Bar	Chocolate	Snack	\$17.99	Jan 12, 2020 10:50 AM	Jan 12, 2023 10:50 AM	15
Scooby Doo's Scooby Snacks	Snack	Snack	\$3.49	Jun 14, 2021 04:30 PM	Jun 14, 2022 04:30 PM	12
MacLaren's Soda	Soda	Drink	\$7.00	Mar 11, 2022 03:30 PM	Jun 11, 2022 03:30 PM	80
Kung Fu Panda's Noodle Soup	Noodle	Food	\$35.00	Feb 28, 2022 01:20 PM	Mar 1, 2022 12:00 AM	40

در صورت عدم وضوح تصویر، [مشاهده](#) نسخه اصلی

توجه کنید که برای ذخیره کردن نام productها از ستون Short Name این جدول استفاده کنید؛ به عبارتی فیلد نام در کلاس Product همان value ستون Short Name جدول بالا است.

#### • نحوه تعامل با کاربر:

در هنگام شروع برنامه لیستی از دستورات قابل فهم برنامه به کاربر نشان داده می شود:

```
[ add.p balance people menu order checkout basket remove.b inventory(remove & change) exit ]
```

## دستورات و نحوه استفاده از آنها

### • add.p :

برای اضافه کردن کاربر، به این صورت به برنامه داده می‌شود:

```
add.p name username balance
```

```
add.p Amirerfan amirerfant 100.2
```

اگر اضافه کردن کاربر موفقیت آمیز بود، پیام زیر

```
Successfully added user.
```

و در غیر این صورت پیام زیر چاپ می‌شود.

```
This username already exist!
```

### • balance :

برای تغییر مقدار موجودی حساب کاربران استفاده می‌شود. به این صورت به برنامه داده می‌شود:

```
balance username amount
```

```
balance farhad1380 525
```

اگر جابجا کردن پول موفقیت آمیز بود، پیام زیر

```
Successfully transmitted.
```

و اگر کاربر وجود نداشت، پیام زیر چاپ می‌شود.

```
Invalid username.
```

## • people:

اگر کاربر این دستور را وارد کند، لیستی از تمام کاربران شامل نام، موجودی و username آنها نشان داده می شود.

نمونه:

```
1. name= 'Farhad' username= 'farhad1380' balance= 112.75
2. name= 'Amirerfan' username= 'amirerfant' balance= 51.0
```

اگر لیست افراد خالی بود، پیغام زیر چاپ می شود:

There is no one here!

## • menu:

اگر کاربر این دستور را وارد کند، لیستی از تمام آیتم های منو شامل نام، دسته بندی، قیمت، تاریخ تولید و انقضاء آنها نشان داده می شود.

نمونه:

```
1. name= 'Burger' category= 'Food' price= 15.75$ manufactureDate= 2022-03-03 08:30 expirationDate= 2022-03-03 20:30 instock: 100
2. name= 'Noodle' category= 'Food' price= 35.0$ manufactureDate= 2022-02-28 13:20 expirationDate= 2022-04-01 00:00 instock: 40
3. name= 'Snack' category= 'Snack' price= 3.49$ manufactureDate= 2021-01-14 16:30 expirationDate= 2022-01-14 16:30 instock: 12
4. name= 'Steak' category= 'Food' price= 100.0$ manufactureDate= 2022-03-06 21:30 expirationDate= 2022-03-09 00:00 instock: 5
5. name= 'Coffee' category= 'Drink' price= 8.5$ manufactureDate= 2022-03-01 13:20 expirationDate= 2022-03-01 15:20 instock: 55
6. name= 'Chocolate' category= 'Snack' price= 17.99$ manufactureDate= 2020-01-12 10:50 expirationDate= 2023-01-12 10:50 instock: 15
7. name= 'Soda' category= 'Drink' price= 7.0$ manufactureDate= 2022-03-11 15:30 expirationDate= 2022-06-11 15:30 instock: 80
8. name= 'Pizza' category= 'Food' price= 19.99$ manufactureDate= 2022-04-05 21:00 expirationDate= 2022-05-09 21:00 instock: 10
```

## • order:

این دستور برای سفارش دادن استفاده می شود و زمانی که کاربر این دستور را وارد می کند، قصد دارد محصولی را به سبد خرید خود اضافه کند.

به این صورت به برنامه داده می شود:

order username food



order amirerfant Chocolate

اگر سفارش موفقیت آمیز بود، پیغام زیر چاپ می شود:

Successfully added to your basket.

اگر username اشتباه بود، پیغام زیر چاپ می شود:

Invalid username.

اگر اسم غذا اشتباه بود، پیغام زیر چاپ می شود:

Invalid food.

در نهایت اگر غذا موجود نبود، پیغام زیر چاپ می شود:

Out of stock.

#### • checkout:

این دستور برای تسویه حساب است. اگر کاربر این دستور را وارد کرد، باید مجموع ارزش سبد خرید او را با مالیت حساب کنید.

#### • نحوه محاسبه مالیات:

همانطور که گفته شد هر یک از محصولات عضوی از یکی از سه دسته ی Drink-Food-Snack است.

دسته بندی Drink شامل ۳۵ درصد، دسته بندی Food شامل ۱۰ درصد و دسته بندی Snack شامل ۲۰ درصد مالیات می شود.

دستور «checkout» به صورت زیر به برنامه داده می شود:

checkout username

اگر username اشتباه بود، پیغام زیر

```
Invalid username.
```

اگر موجودی کاربر کافی نبود، پیغام زیر

```
You can't afford your bill.
```

و در نهایت اگر تسویه حساب موفقیت آمیز بود، پیغامی شبیه به پیغام زیر چاپ شده و سبد خرید کاربر خالی می شود:

```
Amirerfan's bill= 119.45  
tax = 12.45 net = 107.0  
Your new balance= 130.55  
Have a good day.
```

• basket:

محصولات موجود در سبد خرید کاربر را نشان می دهد که به صورت زیر به برنامه داده می شود:

```
basket username
```

```
basket amirerfant
```

اگر username اشتباه بود، پیغام زیر چاپ می شود:

```
Invalid username.
```

نمونه نمایش سبد خرید:

Farhad's Basket:

1. Noodle Food 35.0\$
2. Pizza Food 19.99\$

• **remove.b**

این دستور برای حذف جنس موجود در سبد خرید استفاده می شود که به صورت زیر به برنامه داده می شود:

```
remove.b username index
```

- توجه داشته باشید که **index**، همان شماره ردیف محصول در دستور مشاهده سبد خرید است. برای مثال شماره ردیف **Pizza**، دو است.

```
remove.b amirerfant 2
```

اگر **username** اشتباه بود، پیام زیر

```
Invalid username.
```

اگر ایندکس معتبر نبود، پیام زیر

```
Invalid index.
```

در غیر این صورت، پیام زیر چاپ می شود.

```
Successfully removed.
```

- **inventory**

این دستور برای مدیریت انبار استفاده می‌شود. توجه داشته باشید که این بخش از دسترسی کاربران عادی خارج است و برای ورود به این بخش کاربر باید رمز عبور وارد کند.

- رمز عبور این بخش «ceit-2022» است.

نکته‌ی دیگر این است که بخش inventory دو دستور زیر مجموعه به نام‌های change و remove دارد.

- **دستور زیرمجموعه remove:**

برای حذف کردن یک محصول از منو استفاده می‌شود که به صورت زیر به برنامه داده می‌شود:

```
inventory password remove food
```

```
inventory ceit-2022 remove Coffee
```

اگر حذف کردن موفقیت آمیز بود، پیام زیر نشان داده می‌شود:

```
Successfully removed.
```

- **دستور زیرمجموعه change:**

برای تغییر دادن موجودی یک محصول در منو استفاده می‌شود که به صورت زیر به برنامه داده می‌شود:

```
inventory password change food amount
```

```
inventory ceit-2022 change Coffee 5
```

اگر تغییر موفقیت آمیز بود، پیام زیر نشان داده می‌شود:

Successfully changed the amount.

اگر تغییر تعداد، منفی بود و بزرگتر از مقدار موجودی رستوران بود، پیغام زیر نشان داده می‌شود:

Greater than available amount in stock

در هر کدام از دو دستور زیر مجموعه بالا اگر رمز عبور اشتباه وارد شد، پیغام زیر نشان داده می‌شود و اجازه ورود داده نمی‌شود:

Invalid password.

به همین ترتیب اگر در هر کدام از دو دستور بالا نام غذا نیز غیر معتبر باشد، پیغام زیر نشان داده می‌شود:

Invalid food.

• exit:

این دستور برای پایان دادن برنامه است. تا وقتی که کاربر این دستور را وارد نکرده، برنامه ادامه پیدا می‌کند و از کاربر ورودی دریافت می‌کند. نمونه خروجی دستور:

Have a nice day, chief.

**توجه!** مستندسازی به کمک Javadoc، کامنت‌گذاری و رعایت اصول کدنویسی خوانا برای همه کلاس‌های پیاده‌سازی شده الزامی است.