به نام خدا

تمرین سوم درس برنامه‌نویسی پیشرفته

نیم‌سال دوم 1399-1400

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

AP-HW3-Saman_Hoseini-9731079.zip

۲. در سوال‌هایی که ورودی و خروجی مطلوب آن‌ها مشخص شده است، برنامه‌ی شما به صورت ماشینی تصحیح می‌شود. بنابراین رعایت نحوه ورودی‌گرفتن و نمایش خروجی اهمیت بسیاری دارد. دقیقا همان‌طور که از شما خواسته شده است ورودی‌ها را خوانده و خروجی‌ها را تولید کنید.

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 گروه تلگرامی 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 ۳ اردیبهشت ۱۴۰۰ ساعت ۲۳:۵۹ شب

فهرست سوالات صفحه

سوال اول

الف) مفاهیم زیر را به اختصار توضیح دهید.

Wrapper class, Autoboxing and Unboxing, Garbage Collection

ب) به سوالات زیر پاسخ دهید.

۱. آیا فرایند Garbage Collection تضمین می‌کند که حافظه‌ی برنامه کم نمی‌آید؟

۲. توضیح دهید که هنگام ساخت یک Object در جاوا، چه روندی طی می‌شود؟

۳. چهار روش متفاوت برای پیمایش (Iterate) روی یک لیست را نام برده و توضیح دهید.

۴. پدیده Stack Overflow چطور رخ می‌دهد و چگونه می‌توان از رخداد آن پیشگیری کرد؟

۵. کلاس های HashMap و HashSet را از نظر ساختاری باهم مقایسه کنید.

سوال دوم

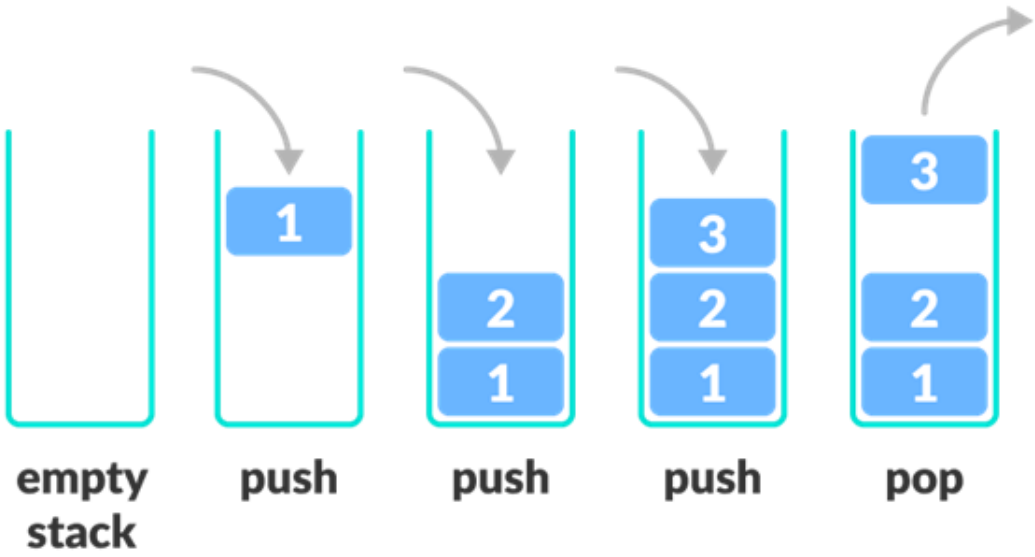
صحیح یا غلط بودن موارد زیر را با توضیح مختصر تعیین کنید.

1. Information hiding decreases the level of dependence.
2. Package access members are accessible to any class.
3. Classes can be private too like fields.
4. A map can contain duplicate keys.
5. Java is always “Pass by reference”.
6. It is possible to have a private constructor in java.

سوال سوم

پشته یا Stack یک ساختمان داده معروف است که کاربردهای زیادی در دنیای کامپیوتر دارد.
ساختمان داده پشته به صورت LIFO یا Last In First Out است. به این معنی که هر داده‌ای که در آخر وارد پشته شود، زودتر از همه خارج می‌شود.

این ساختمان داده را می‌توان به کمک آرایه پیاده‌سازی کرد.



پشته باید بتواند کارهای زیر را انجام دهد:

1. Push: یک داده را به بالای پشته اضافه می‌کند. (اگر پشته پر باشد Stack Overflow اتفاق می‌افتد.)
2. Pop: یک داده را از بالای پشته بر می‌دارد و آن را حذف می‌کند. (اگر پشته خالی باشد Stack Underflow اتفاق می‌افتد.)
3. Peek: مقدار بالاترین عضو پشته را برمی‌گرداند ولی آن را حذف نمی‌کند.
4. isEmpty: مشخص می‌کند که پشته خالی هست یا خیر.
5. isFull: مشخص می‌کند که پشته پر است یا خیر.

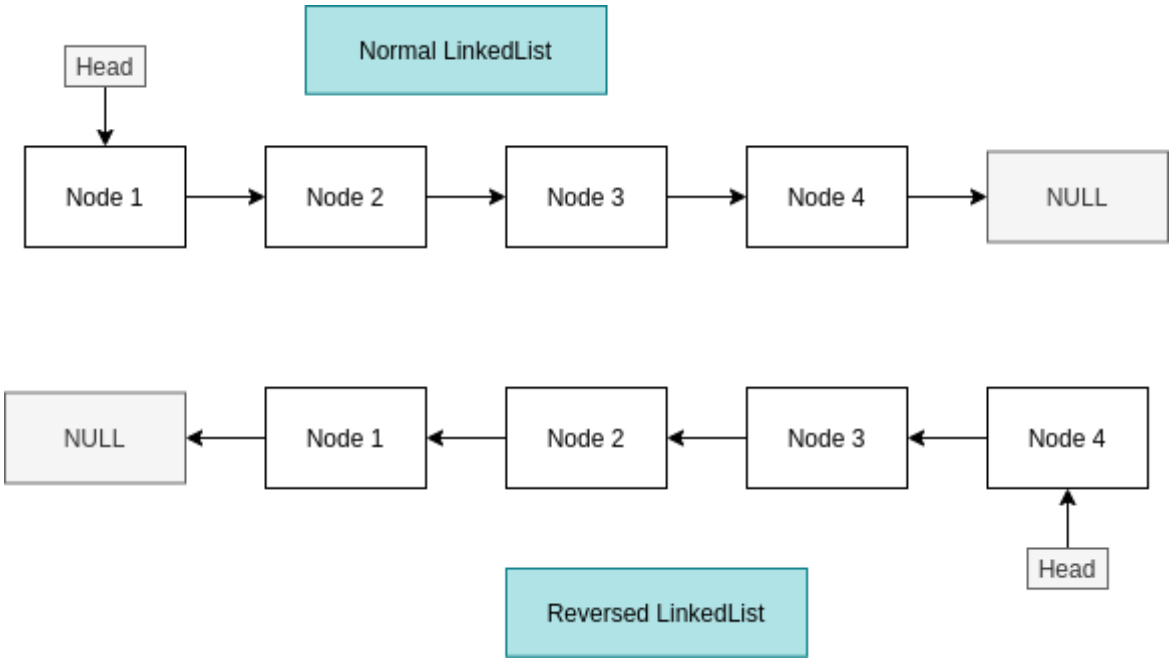
تفاوت متد pop و peek در این است که pop بالاترین عنصر پشته را برمی‌دارد و حذف می‌کند ولی peek فقط مقدار بالاترین عضو پشته را برمی‌گرداند.

برای اطلاعات بیشتر از پشته و نحوه عملکرد آن از لینک زیر استفاده کنید:

https://www.tutorialspoint.com/data_structures_algorithms/stack_algorithm.htm

از کاربردهای پشته می‌توان به معکوس کردن لیست پیوندی (Linked List) اشاره کرد.

یک لیست پیوندی را در نظر بگیرید، در صورتی که جهت اتصال همه گره (Node) ها را برعکس کنیم، لیست پیوندی ما معکوس می‌شود. به مثال زیر توجه کنید:



برنامه‌ای بنویسید که به کمک پشته یک لیست پیوندی را معکوس کند.

ورودی برنامه:

در خط اول یک آرایه از اعداد ورودی داده می‌شود.

1. اعداد را از راست به چپ بخوانید و یک لیست پیوندی با آنها بسازید.
2. روی لیست پیوندی پیمایش کنید و تمامی گره (Node) های آن را در پشته Push کنید.
3. دوباره از ابتدای لیست پیوندی، روی آن پیمایش کنید و مقادیر داخل پشته را Pop کنید و آنها را به ترتیب معکوس وصل کنید.

خروجی برنامه:

لیست پیوندی معکوس شده را چاپ کنید.

مثال:

Input:
1 2 3 4 5 6 7 8 9
Output:
9 8 7 6 5 4 3 2 1

Input
11 7 19 3 4 15 22
Output
22 15 4 3 19 7 11

توجه! در پیاده سازی پشته (Stack) حق استفاده از کلاس‌های آماده را ندارید و باید تمامی آن را دستی پیاده سازی کنید. اما استفاده از کلاس آماده لیست پیوندی (Linked List) بلامانع است، در صورتی که کلاس لیست پیوندی (Linked List) دستی پیاده شود نمره امتیازی دارد.

سوال چهارم

الف) با توجه به کلاس‌های Coordinate و Triangle که در ادامه تعریف شده‌اند، وضعیت حافظه‌های Heap و Stack را دقیقاً بعد از اجرای خط آخر متد run، در یکی از قالب‌های Pointer Model یا Address Model رسم کنید و مشخص کنید که آیا Garbage Object وجود دارد یا خیر. برای پاسخ خود دلیل بیاورید. (از حافظه اختصاص داده‌شده به متدهای Static صرف نظر کنید.)

کلاس Coordinate:

```
public class Coordinate {
    int x, y;

    public Coordinate(int x, int y)
    {
        this.x = x;
        this.y = y;
    }
}
```

کلاس Triangle:

```
public class Triangle {
    Coordinate c1, c2, c3;

    public Triangle(Coordinate c1, Coordinate c2, Coordinate c3) {
        this.c1 = c1;
        this.c2 = c2;
        this.c3 = c3;
    }
    public double area() {
        return 0.5 * (c1.x * (c2.y - c3.y) + c2.x * (c3.y - c1.y)
            + c3.x * (c1.y - c2.y));
    }
    public double perimeter() {
        return Math.sqrt(Math.pow(c1.x - c2.x, 2)
                        + Math.pow(c1.y - c2.y,    2))
                        + Math.sqrt(Math.pow(c1.x - c3.x, 2)
                        + Math.pow(c1.y - c3.y, 2))
                        + Math.sqrt(Math.pow(c3.x - c2.x, 2)
                        + Math.pow(c3.y - c2.y, 2));
    }
}
```

متد run:

```
public void run() {
        Coordinate c1 = new Coordinate(0, 0);
        Coordinate c2 = new Coordinate(3, 0);
        Coordinate c3 = new Coordinate(0, 4);
        Coordinate c4 = new Coordinate(1, 1);
        Coordinate c5 = new Coordinate(3, 0);
        Triangle t1 = new Triangle(c1, c2, c3);
        double area = t1.area();
        double perimeter = t1.perimeter();
        System.out.println("Area: " + area);
        System.out.println("Perimeter: " + perimeter);
}
```

ب) کد زیر را در نظر گرفته و خروجی نهایی کنسول را با ذکر دلیل مشخص کنید. (از IDE نباید استفاده شود.)

```
public static void main(String[] args) {
        String str1 = "CEIT";
        String str2 = "C" + "E" + "I" + "T";
        System.out.println(str1 == str2);
        System.out.println(str1.equals(str2));

        String str3 = "123";
        String str4 = Integer.toString(123);
        System.out.println(str3 == str4);
        System.out.println(str3.equals(str4));
}
```

سوال پنجم

در این سوال می‌خواهیم یک فروشگاه آنلاین مواد غذایی پیاده سازی کنیم که در ادامه توضیح کلاس‌های آن آمده است. در صورت نیاز می‌توانید متدهایی اضافه کنید که Signature آن را خودتان مشخص می کنید.

- کلاس Product:

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

- تاریخ تولید و انقضا باید از کلاس LocalDate یا Date باشند.
- فرمت تاریخ باید به صورت dd-mm-yyyy باشد.
- نباید تاریخ تولید بعد از تاریخ انقضا باشد.
- قیمت محصولات می‌تواند عددی اعشاری باشد.

- کلاس Inventory:

یک HashMap دارد و تمامی محصولات موجود در فروشگاه 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 قبلا 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.

کلاس‌های Product و Inventory و Basket را پیاده سازی کنید. همه کلاس ها باید متد toString را override کنند و در صورت نیاز با صدا زدن این متد اطلاعات مورد نیاز را چاپ کنند.

کلاس Main را به صورت زیر پیاده سازی کنید:

۱. یک Instance از کلاس Inventory با نام inventory بسازید.

محصولات زیر را با اطلاعات داده شده مربوط به هر یک بسازید و همه آنها را با موجودی داده شده در HashMap ذخیره کنید.

اطلاعات محصولات:

NAME	CATEGORY	WEIGHT	PRICE	MANUFACTURE	EXPIRATION	STOCK
Carrot	Vegetables	5	20	15-3-2020	15-3-2021	10
Apple	Fruits	10	50	1-4-2020	1-8-2020	50
12xEggs	Egg	100	40	1-1-2020	1-6-2020	20
Oats	Grains	70	100	1-6-2020	1-1-2021	45
Salmon	Seafood	150	250	1-1-2020	1-2-2020	5
Stake	Meat	800	1000	1-3-2020	1-9-2020	5
Milk	Dairy	100	20	10-1-2020	25-1-2020	20
Cheese	Dairy	150	10	1-2-2020	15-3-2020	50

۲. یک Instance از کلاس Basket با نام basket بسازید. در آینده برای خرید محصول از این کلاس استفاده می‌کنیم.

رابطه کنسولی با کاربر:
در شروع برنامه لیستی از تمام محصولات موجود در انبار با میزان موجودی آنها به کاربر نمایش دهید.

قالب خروجی باید به فرمت JSON باشد. برای اطلاعات بیشتر به لینک زیر مراجعه کنید.

https://stackabuse.com/reading-and-writing-json-in-java/

مانند مثال زیر:

```
1){
  "Product": {
    "NAME": "Salmon",
    "CATEGORY": "Seafood",
    "WEIGHT": "150.0",
    "PRICE": "250.0",
    "MANUFACTURE_DATE": 2020-01-01,
    "EXPIRATION_DATE": 2020-02-01
  }
}instock: 5
2){
  "Product": {
    "NAME": "Milk",
    "CATEGORY": "Dairy",
    "WEIGHT": "100.0",
    "PRICE": "20.0",
    "MANUFACTURE_DATE": 2020-01-10,
    "EXPIRATION_DATE": 2020-01-25
  }
}instock: 20
3){
  "Product": {
    "NAME": "Fetacheese",
    "CATEGORY": "Dairy",
    "WEIGHT": "150.0",
    "PRICE": "10.0",
    "MANUFACTURE_DATE": 2020-02-01,
    "EXPIRATION_DATE": 2020-03-15
  }
}instock: 50
```

```
4){
  "Product": {
    "NAME": "Redmeat",
    "CATEGORY": "Meat",
    "WEIGHT": "800.0",
    "PRICE": "1000.0",
    "MANUFACTURE_DATE": 2020-03-01,
    "EXPIRATION_DATE": 2020-09-01
  }
}instock: 5
5){
  "Product": {
    "NAME": "Eggshell",
    "CATEGORY": "Eggs",
    "WEIGHT": "100.0",
    "PRICE": "40.0",
    "MANUFACTURE_DATE": 2020-01-01,
    "EXPIRATION_DATE": 2020-06-01
  }
}instock: 20
6){
  "Product": {
    "NAME": "Apple",
    "CATEGORY": "Vegetables",
    "WEIGHT": "10.0",
    "PRICE": "50.0",
    "MANUFACTURE_DATE": 2020-04-01,
    "EXPIRATION_DATE": 2020-08-01
  }
}instock: 50
```

```
7){
  "Product": {
    "NAME": "Oats",
    "CATEGORY": "Grains",
    "WEIGHT": "70.0",
    "PRICE": "100.0",
    "MANUFACTURE_DATE": 2020-06-01,
    "EXPIRATION_DATE": 2021-01-01
  }
}instock: 45
8){
  "Product": {
    "NAME": "Carrot",
    "CATEGORY": "Vegetables",
    "WEIGHT": "5.0",
    "PRICE": "20.0",
    "MANUFACTURE_DATE": 2020-03-15,
    "EXPIRATION_DATE": 2021-03-15
  }
}instock: 10
```

حال منتظر دستورات ورودی کاربر باشید:

- دستور add:

بعد از دستور add یک عدد داده می‌شود، این عدد بیانگر Index محصولی است که کاربر قصد اضافه کردن آن را به سبد خود دارد. منظور از ایندکس، عدد محصول در نمایش بالا است. به این صورت:

```
add k
```

توجه داشته باشید هر محصولی که به سبد خرید کاربر اضافه شود، یکی از موجودی انبار آن محصول کم می‌شود، همچنین امکان منفی شدن موجودی محصولی در انبار وجود ندارد.

- دستور remove:

بعد از دستور remove یک عدد داده می‌شود، این عدد بیانگر Index محصولی است که کاربر قصد حذف کردن ان را از سبد خود دارد. به این صورت:

```
remove k
```

Index ای که کاربر وارد می‌کند در واقع Index محصولات داخل سبد خرید خود است، به عنوان مثال اگر لیست محصولات داخل سبد خرید کاربر به صورت زیر باشد:

```
Itemsincart:
1){
  "Product": {
    "NAME": "Salmon",
    "CATEGORY": "Seafood",
    "WEIGHT": "150.0",
    "PRICE": "250.0",
    "MANUFACTURE_DATE": 2020-01-01,
    "EXPIRATION_DATE": 2020-02-01
  }
}
2){
  "Product": {
    "NAME": "Milk",
    "CATEGORY": "Dairy",
    "WEIGHT": "100.0",
    "PRICE": "20.0",
    "MANUFACTURE_DATE": 2020-01-10,
    "EXPIRATION_DATE": 2020-01-25
  }
}
3){
  "Product": {
    "NAME": "Fetacheese",
    "CATEGORY": "Dairy",
    "WEIGHT": "150.0",
    "PRICE": "10.0",
    "MANUFACTURE_DATE": 2020-02-01,
    "EXPIRATION_DATE": 2020-03-15
  }
}
```

و کاربر عدد ۲ را به عنوان Index ورودی بدهد، منظور کاربر حذف کردن محصول دوم از سبد خریدش است.

توجه داشته باشید هر محصولی که از سبد خرید حذف شود به انبار محصولات برمی‌گردد و باید موجودی آن محصول در انبار افزایش یابد.

• دستور cart:
این دستور لیست تمامی محصولات داخل سبد خرید را چاپ می‌کند. به صورت زیر:

```
Itemsincart:
1){
  "Product": {
    "NAME": "Salmon",
    "CATEGORY": "Seafood",
    "WEIGHT": "150.0",
    "PRICE": "250.0",
    "MANUFACTURE_DATE": 2020-01-01,
    "EXPIRATION_DATE": 2020-02-01
  }
}
2){
  "Product": {
    "NAME": "Milk",
    "CATEGORY": "Dairy",
    "WEIGHT": "100.0",
    "PRICE": "20.0",
    "MANUFACTURE_DATE": 2020-01-10,
    "EXPIRATION_DATE": 2020-01-25
  }
}
```

```
3){
  "Product": {
    "NAME": "Feta cheese",
    "CATEGORY": "Dairy",
    "WEIGHT": "150.0",
    "PRICE": "10.0",
    "MANUFACTURE_DATE": 2020-02-01,
    "EXPIRATION_DATE": 2020-03-15
  }
}
4){
  "Product": {
    "NAME": "Red meat",
    "CATEGORY": "Meat",
    "WEIGHT": "800.0",
    "PRICE": "1000.0",
    "MANUFACTURE_DATE": 2020-03-01,
    "EXPIRATION_DATE": 2020-09-01
  }
}

5){
  "Product": {
    "NAME": "Eggshell",
    "CATEGORY": "Eggs",
    "WEIGHT": "100.0",
    "PRICE": "40.0",
    "MANUFACTURE_DATE": 2020-01-01,
    "EXPIRATION_DATE": 2020-06-01
  }
}
```

در صورتی که لیست خالی باشد پیغام زیر چاپ شود:

```
List is empty.
```

توجه داشته باشید برای چاپ کردن اطلاعات محصولات باید متد toString را صدا بزنید.

• دستور products:
این دستور لیستی از تمام محصولات موجود در انبار با میزان موجودی آنها را چاپ می‌کند، به صورت زیر:

```
1){
  "Product": {
    "NAME": "Salmon",
    "CATEGORY": "Seafood",
    "WEIGHT": "150.0",
    "PRICE": "250.0",
    "MANUFACTURE_DATE": 2020-01-01,
    "EXPIRATION_DATE": 2020-02-01
  }
}instock: 5
2){
  "Product": {
    "NAME": "Milk",
    "CATEGORY": "Dairy",
    "WEIGHT": "100.0",
    "PRICE": "20.0",
    "MANUFACTURE_DATE": 2020-01-10,
    "EXPIRATION_DATE": 2020-01-25
  }
}instock: 20
3)
{
  "Product": {
    "NAME": "Fetacheese",
    "CATEGORY": "Dairy",
    "WEIGHT": "150.0",
    "PRICE": "10.0",
    "MANUFACTURE_DATE": 2020-02-01,
    "EXPIRATION_DATE": 2020-03-15
  }
}instock: 50
```

```
4){
  "Product": {
    "NAME": "Redmeat",
    "CATEGORY": "Meat",
    "WEIGHT": "800.0",
    "PRICE": "1000.0",
    "MANUFACTURE_DATE": 2020-03-01,
    "EXPIRATION_DATE": 2020-09-01
  }
}instock: 5
5){
  "Product": {
    "NAME": "Eggshell",
    "CATEGORY": "Eggs",
    "WEIGHT": "100.0",
    "PRICE": "40.0",
    "MANUFACTURE_DATE": 2020-01-01,
    "EXPIRATION_DATE": 2020-06-01
  }
}instock: 20
6){
  "Product": {
    "NAME": "Apple",
    "CATEGORY": "Vegetables",
    "WEIGHT": "10.0",
    "PRICE": "50.0",
    "MANUFACTURE_DATE": 2020-04-01,
    "EXPIRATION_DATE": 2020-08-01
  }
}instock: 50
```

```
7){
  "Product": {
    "NAME": "Oats",
    "CATEGORY": "Grains",
    "WEIGHT": "70.0",
    "PRICE": "100.0",
    "MANUFACTURE_DATE": 2020-06-01,
    "EXPIRATION_DATE": 2021-01-01
  }
}instock: 45
8){
  "Product": {
    "NAME": "Carrot",
    "CATEGORY": "Vegetables",
    "WEIGHT": "5.0",
    "PRICE": "20.0",
    "MANUFACTURE_DATE": 2020-03-15,
    "EXPIRATION_DATE": 2021-03-15
  }
}instock: 10
```

در صورتی که تمامی محصولات تمام شده باشند پیغام زیر چاپ شود:

```
We are out of stock.
```

توجه داشته باشید برای چاپ کردن اطلاعات محصولات باید متد toString را صدا بزنید.

- دستور checkout:

با آمدن این دستور محصولات داخل سبد خرید شما خریداری شده و برنامه با چاپ کردن پیام زیر به اتمام می‌رسد:

```
It was a pleasure doing business with you.
```

توجه! مستندسازی به کمک Javadoc، کامنت گذاری و رعایت اصول کد نویسی خوانا برای همه کلاس‌های پیاده سازی شده الزامی است. همچنین برای این تمرین علاوه بر فایل‌های کد، یک فایل متنی در قالب PDF برای قسمت تشریحی ارائه کنید.