

به نام خدا

## تمرین دوم 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-HW2-FirstName\_LastName-StudentNumber.zip

AP-HW2-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. دقیقاً 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 چهارشنبه ۱۸ فروردین ۱۴۰۰ ساعت ۲۳:۵۵ شب

صفحه

فهرست سوالات

- سوال اول..... ۳
- سوال دوم..... ۴
- سوال سوم..... ۵
- سوال چهارم..... ۷
- سوال پنجم..... ۱۱

## سوال اول

به سوالات زیر پاسخ دهید.

الف) هر Object شامل ۳ قسمت Identity و State, Behavior می باشد. هر کدام از آن ها به چه مفهومی اشاره می کند؟ توضیح دهید.

ب) نحوه تعامل اشیا با یکدیگر چگونه است؟

ج) مفهوم عبارات زیر را به اختصار توضیح دهید.

Overloading, Casting, Modularization, Abstraction

د) Immutable Object چیست؟ شرح دهید و یک مثال بزنید.

## سوال دوم

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

1. If we write our own constructor, the compiler also creates a default constructor in Java.
2. Actions on ArrayList may be slower than standard arrays.
3. We can have an ArrayList of primitive types like int, char and ... .
4. We can define more than one constructor in a class.
5. In an instance method or a constructor, "this" is a reference to the current object.
6. Local variables in Java must be definitely assigned to before they are accessed, or it is a compile error.
7. Variables declared in the body of a particular method are known as instance variables and can be used in all methods of the class.
8. The values assigned inside the constructor overwrite the values initialized with declaration.

## سوال سوم

الف) به سوالات داده شده به طور کامل پاسخ دهید:

1. آیا امکان عوض کردن return type یک تابع در method overloading وجود دارد؟
2. تفاوت بین method و constructor چیست؟
3. آیا امکان overload کردن constructor وجود دارد؟ توضیح دهید.

ب) قطعه کد زیر را با توضیح مختصر اصلاح کنید (مشکلاتی از قبیل نام گذاری نیز اصلاح شود):

```
public class Main {

    public static void main(String[] args) {
        person p1 = new person("Ted", "Mosby", "123456");
        p1.express();
        p1.express("Happy");
        p1.print();
    }

    static class person {
        private String FirstName;
        private String last_name;
        private int id;

        public void person(String FirstName, String last_name, int id){
            FirstName = FirstName;
            last_name = last_name;
            id = id;
        }

        public void express(){
            System.out.println("I feel neutral");
        }

        public int express(String state){
            System.out.println("I feel " + state + " today");
            return 0;
        }

        public void print(){
            System.out.println("person{" +
                "name='" + FirstName + '\'' +
                ", lname='" + last_name + '\'' +
                ", id=" + id +
                '}');
        }
    }
}
```

## سوال چهارم

الف) به سوالات زیر پاسخ دهید.

1. تفاوت بین LinkedList، ArrayList و Array در چیست؟ توضیح دهید.

2. یک بیمارستان بیماران کرونایی را پذیرش می کند و محدودیتی در تعداد بیماران ندارد. برای ذخیره اطلاعات بیماران (اطلاعات هر بیمار شامل نام و کد ملی اوست) می خواهیم از ArrayList یا LinkedList و یا Array استفاده کنیم. با توجه به اینکه همواره بیماران اضافه یا مرخص می شوند، بهتر است از کدام یک از کالکشن های مذکور استفاده کنیم؟ علت را توضیح دهید.

3. آیا امکان اضافه کردن چند باره ی یک Object به ArrayList در جاوا وجود دارد؟

4. تفاوت بین تابع length() برای یک آرایه و تابع size() برای یک ArrayList چیست؟

ب) با مفهوم لیست پیوندی (LinkedList) در درس مبانی برنامه نویسی آشنا شدید. با توجه به مفاهیم برنامه نویسی شی گرای (کلاس)، لیست پیوندی ای از اعداد صحیح پیاده سازی کنید که قابلیت انجام اعمال زیر را داشته باشد:

```
void add(int k)
```

یک عدد می گیرد و آن را به انتهای لیست اضافه می کند.  
شکل ورودی:

```
add k
```

```
void addFirst(int k)
```

یک عدد می گیرد و آن را به ابتدای لیست اضافه می کند.  
شکل ورودی:

```
addFirst k
```

```
void findMiddle()
```

عنصر وسط لیست چاپ می شود. در صورت زوج بودن تعداد، عنصری که اندیس کوچکتری دارد چاپ می شود. (مثلا اگر ۴ عنصر داریم، عنصر با اندیس ۱ چاپ می شود) اگر لیست خالی بود، عدد ۱- چاپ می شود.

شکل ورودی:

```
addFirst k
```

```
void removeMiddle()
```

عنصر وسط لیست حذف می‌شود. در صورت خالی بودن لیست اتفاقی نمی‌افتد.  
شکل ورودی:

```
removeMiddle
```

```
void print()
```

لیست پیوندی را چاپ می‌کند. در صورت خالی بودن لیست، یک خط خالی چاپ شود.  
شکل ورودی:

```
print
```

```
void contains(int k)
```

یک عدد می‌گیرد و در صورت پیدا شدن داده در لیست پیوندی، شماره‌ی اندیس آن و در صورت نبودن، عدد ۱- را چاپ می‌کند.  
شکل ورودی:

```
contains k
```

```
void removeIndex(int i)
```

یک اندیس (index) می‌گیرد و عنصر موجود در آن مکان را حذف می‌کند. در صورت خالی بودن لیست یا نبودن عنصری در آن مکان، اتفاقی نمی‌افتد.  
شکل ورودی:

```
removeIndex i
```

```
finish
```

گرفتن ورودی‌ها به اتمام می‌رسد و خروجی‌ها چاپ می‌شوند.

توجه: اندیس‌ها از صفر شروع می‌شوند. همچنین برای سادگی برنامه، ورودی تکراری داده نمی‌شود.



نمونه ورودی و خروجی:

```
Input 1:
add 15
add 34
add 17
findMiddle
add 46
print
removeIndex 2
print
addFirst 4
print
findMiddle
removeIndex 20
contains 46
removeIndex 0
print
contains 4
removeMiddle
print
finish
```

Output 1:

34

15 34 17 46

15 34 46

4 15 34 46

15

3

15 34 46

-1

15 46

## سوال پنجم

طراحی و مدل سازی یک سیستم جامع اینترنت بانک در این سوال شما باید طراحی و پیاده سازی یک سیستم اینترنت بانک را بر عهده بگیرید. در ادامه سوال به طور کامل ویژگی‌های این سیستم برای شما شرح داده می‌شود.

این سیستم قابلیت ثبت نام (register) کردن کاربران را دارد. کاربرانی که ثبت نام کردند، می‌توانند با وارد شدن به سیستم (login)، اعمال مختلفی را انجام دهند و همچنین مدیر سیستم (system admin) نیز قابلیت ورود به سیستم و مشاهده اطلاعات تمامی کاربران و حساب‌های آن‌ها را دارد.

توضیحات کلاس‌ها و متدهای هر کدام:

### تراکنش (Transaction)

تراکنش به هرگونه تغییر در موجودی حساب اعم از برداشت، واریز و یا انتقال وجه گفته می‌شود. این کلاس باید ویژگی immutability داشته باشد.

فیلدهای این کلاس عبارتند از:

- هر تراکنش یک مقدار (amount) دارد. (می‌تواند مثبت یا منفی باشد)
- هر تراکنش یک تاریخ (date) دارد.

مقدار فیلد amount در constructor مقداردهی می‌شود.

تاریخ تراکنش باید تاریخ دقیق همان لحظه باشد. برای این کار از لینک زیر کمک بگیرید:

[https://www.tutorialspoint.com/java/java\\_date\\_time.htm](https://www.tutorialspoint.com/java/java_date_time.htm)

این کلاس دارای متد زیر است:

```
print()
```

توضیح متد:

اطلاعات مربوط به تراکنش را چاپ می‌کند.

### حساب (Account)

فیلدهای این کلاس عبارتند از:

- هر حساب یک سریال (serial) از جنس UUID دارد. (کلاس java.util.UUID)
- هر حساب یک کد ملی (id) دارنده (صاحب) حساب را دارد.
- هر حساب یک نام (firstName) و نام خانوادگی (lastName) دارنده (صاحب) را دارد.
- هر حساب یک نوع (type) دارد. (مانند سپرده، اندوخته و ...)
- هر حساب یک مانده حساب (balance) دارد.
- هر حساب لیستی از تراکنش‌ها (transactions) دارد.

مقادیر فیلدهای `id`، `firstName`، `lastName`، `type` و `balance` اولیه در `constructor` مقداردهی می‌شود.  
سریال هر حساب باید منحصر به فرد باشد و توسط خود برنامه تولید شده باشد. برای این کار از لینک زیر کمک بگیرید:

<https://www.javatpoint.com/java-uuid>

این کلاس دارای متدهای زیر است:

```
updateBalance(int amount)
addTransaction(Transaction transaction)
printTransactions()
printAccountData()
```

متدهای بالا به ترتیب:

۱. به روز رسانی کردن مانده حساب بعد از هر تراکنشی اعم از برداشت، واریز و یا انتقال.
۲. اضافه کردن یک تراکنش به لیست تراکنش‌ها.
۳. چاپ کردن لیست تراکنش‌ها.
۴. چاپ کردن اطلاعات حساب.

کاربر (**User**)

فیلدهای این کلاس عبارتند از:

- هر کاربر یک نام (`firstName`) و یک نام خانوادگی (`lastName`) دارد.
- هر کاربر یک کد ملی (`id`) دارد که قابل تغییر دادن نیست.
- هر کاربر یک رمز عبور (`password`) دارد.
- هر کاربر لیستی از حساب‌ها (`accountList`) دارد.

مقادیر فیلدهای `firstName`، `lastName` و `id` و `password` کاربر در `constructor` مقداردهی می‌شود.  
این کلاس دارای متدهای زیر است:

```
addAccount(Account account)
removeAccount(Account account)
deposit(Account account, int amount)
withdrawal(Account account, int amount)
transfer(Account srcAccount, Account destAccount, int amount)
checkBalance(Account account)
printAllAvailableAccounts()
printUserData()
```

متدهای بالا به ترتیب:

۱. حساب مورد نظر را به لیست حسابها اضافه می کند.
  ۲. حساب مورد نظر را از لیست حسابها حذف می کند.
  ۳. یک حساب و یک مقدار وجه می گیرد و از آن حساب مقداری برداشت می کند.
  ۴. یک حساب و یک مقدار وجه می گیرد و به آن حساب مقداری واریز می کند.
  ۵. مقدار وجه مورد نظر را از srcAccount (حساب مبدا) به destAccount (حساب مقصد) واریز می کند. توجه داشته باشید که destAccount می تواند یکی از حسابهای خود فرد یا فرد دیگری باشد و امکان منفی شدن باقی مانده حساب مبدا (srcAccount) نیز وجود ندارد.
  ۶. یک حساب می گیرد و موجودی آن را چاپ می کند.
  ۷. اطلاعات تمامی حسابهای موجود را چاپ می کند.
  ۸. اطلاعات کاربر از جمله نام، نام خانوادگی و کد ملی را چاپ می کند.
- در متدهای بالا امکان منفی شدن باقی مانده حساب وجود ندارد و در صورت برداشت (withdrawal) یا انتقال (transfer) بیش از حد مجاز پیغامی مناسب چاپ شود.
  - در متدهای بالا در صورتی که حساب مبدا (در transfer) و یا حسابی که در پارامتر ورودی متدهای برداشت وجه، واریز وجه، مانده حساب داده می شود در لیست حسابهای کاربر نباشد یا به عبارتی دیگر متعلق به او نباشد پیغامی مناسب چاپ شود.
  - در متدهای addAccount و removeAccount باید تغییرات اعم حذف یا اضافه کردن حساب در لیست حسابهای سیستم بانکی نیز اعمال شود.
  - در متدهای بالا در صورتی که تراکنشی (deposit, withdrawal, transfer) بر روی حسابی در حال انجام است، آن را به لیست تراکنشهای داخل آن حساب اضافه کنید. به عنوان مثال اگر \$۲۰ از حساب با سریال xxxxxx برداشته شده برای این تغییر در موجودی (کاهش \$۲۰ از حساب) یک آبجکت از کلاس Transaction ساخته شود و به لیست تراکنشهای آن حساب (Account) اضافه شود.

سامانه کلی مدیریت بانکی (BankingSystem)

فیلدهای این کلاس عبارتند از:

- این کلاس لیستی از کاربران (users) دارد.
- این کلاس لیستی از حسابها (accounts) دارد.

این کلاس دارای متدهای زیر است:

```
register(User user)
login(String id, String password)
addUser(User user)
removeUser(User user)
displayUsers()
addAccount(Account account)
removeAccount(Account account)
displayAccounts()
findAccount(String serial)
```

متدهای بالا به ترتیب:

۱. یک کاربر جدید می‌گیرد و به لیست کاربران اضافه می‌کند.
۲. یک ID و رمز عبور می‌گیرد و کاربر مورد نظر را با توجه به ID داده شده از لیست کاربران پیدا می‌کند، در صورت مطابقت رمز عبور داده شده با رمز عبور کاربر پیغامی مناسب چاپ می‌کند و در غیر این صورت خطایی را گزارش می‌کند.
۳. کاربر مورد نظر را به لیست کاربران اضافه می‌کند.
۴. کاربر مورد نظر را از لیست کاربران حذف می‌کند.
۵. تمامی کاربران موجود در سیستم را چاپ می‌کند.
۶. حساب مورد نظر را به لیست حساب‌ها اضافه می‌کند.
۷. حساب مورد نظر را از لیست حساب‌های حذف می‌کند.
۸. تمامی حساب‌های موجود در سیستم را چاپ می‌کند.
۹. یک رشته می‌گیرد و در بین لیست حساب‌ها به دنبال حسابی می‌گردد که سریالش با سریال ورودی یکسان باشد و آن را برگرداند. اگر همچین حسابی وجود نداشت مقدار null را برمیگرداند.

برای پیمایش روی لیست کاربران از کلاس Iterator استفاده کنید. برای اطلاعات بیشتر به لینک زیر مراجعه کنید:

[https://www.w3schools.com/java/java\\_iterator.asp](https://www.w3schools.com/java/java_iterator.asp)

- در ادامه برای دسترسی به متدهای سیستم از ادمین سیستم استفاده می‌کنیم. ادمین سیستم به کل سیستم بانکی دسترسی دارد و نیازی به تعریف کلاس جدیدی ندارد.
- دقت داشته باشید تمامی فیلدها باید private باشند و در صورت نیاز توابع getter و setter را تعریف کنید.
- در صورت نیاز می‌توان توابع دیگری نیز تعریف و استفاده کرد. ولی از تعریف توابع غیر کاربردی خودداری کنید.

روند کار با سیستم:  
با اجرای برنامه، منوی اصلی سیستم که به این شکل است چاپ می‌شود:

```
1.Sign up
  Enter info
2.Log in
  Enter user ID and password
    1.Existing accounts
    Choose from list of accounts
      1.WithDrawal
      2.Deposit
      3.Transfer
      4.Check balance
      5.Back
    2.Add new account
      Enter info
    3.Log out
3.System Admin
  Enter userID and password
    1.display users
    2.display accounts
    3.remove user
4.Exit
```

توجه: در هر قسمت باید امکان بازگشت به صفحه اصلی منو وجود داشته باشد.

توضیحات هر قسمت:  
• قسمت Sign up:

```
Input:
firstname
lastname
ID
Password
Output:
```

در صورتی که User تکراری باشد پیغام

```
user already exists.
```

و در غیر این صورت، پیغام زیر چاپ شود.

```
user created.
```

- برای مقایسه کاربران تابع **equals** را **override** کنید. (امتیازی)

**Method overriding** کردن را در آینده یاد خواهید گرفت ولی برای اطلاعات بیشتر می‌توانید به لینک زیر مراجعه کنید:

<https://www.geeksforgeeks.org/overriding-in-java/>

- قسمت Log in:

Input:

ID

Password

Output:

در صورتی که user، وجود نداشته باشد یا رمز عبور اشتباه باشد پیغام

```
user doesn't exists or password is incorrect.
```

و در غیر این صورت، پیغام زیر چاپ شود.

```
Logged in.
```

- قسمت Existing accounts:

ابتدا لیستی از تمامی حساب‌های کاربر به صورت زیر چاپ شود:

Output:

Account 1: Serial, user id, username, balance

Account 2: Serial, user id, username, balance

...



سپس شماره حساب مورد نظر (شماره از لیست چاپ شده) وارد شود:

Input:

Account number x

Output:

Logged into account.

1.Withdrawal

2.Deposit

3.Transfer

4.Check balance

5.Back

حال شماره ی مربوط به عملیات دلخواه باید وارد شود.

• قسمت Withdrawal:

Input:

amountOfWithdrawal

Output:

در صورتی که موجودی کافی باشد پیغام

Completed.

و در غیر این صورت پیغام زیر چاپ شود.

Not enough money.

• قسمت Deposit:

Input:

amountOfDeposit

Output:

Completed.

• قسمت Transfer:

Input:

destAccountSerial amount

Output:

اگر حساب مقصد وجود نداشته باشد یا موجودی کافی نباشد پیغام

Destination account doesn't exist or there is not enough money  
in your account.

و در غیر این صورت پیغام زیر چاپ شود.

Completed.

• قسمت Check balance:

Output:

availableMoney

• قسمت Back:

Output:

Logged out of account.

در منوی یک مرحله به عقب می‌رود.

• قسمت Add new account:

Input:

User ID

Account type

Output:

New account opened.

- قسمت System Admin:

ادمین سیستم بانکی دسترسی به تمامی متد ها و اطلاعات سیستم بانکی دارد. برای ورود به سیستم به علت مسائل امنیتی باید یک نام کاربری (username) و پسورد (password) گرفته شود تا هرکسی به جز افراد دارای صلاحیت وارد سیستم نشوند.

- نام کاربری سیستم ادمین sysadmin است و برای راحتی کار پسوردش را ۱۲۳۴ در نظر بگیرید.

Input:

username

password

Output:

در صورتی که رمز عبور یا نام کاربری درست نباشد پیغام

Username or password is incorrect.

و در غیر این صورت پیغام زیر چاپ شود.

Logged in as sysadmin.

سپس شماره عملیات دلخواه وارد شود :

- 1.display users
- 2.display accounts
- 3.remove user
- 4.remove account

- قسمت display users:

Output:

User 1: firstName lastName ID

User 2: firstName lastName ID

...

- قسمت `display accounts`:

Output:

Account 1: Serial, type, balance

Account 2: Serial, type, balance

...

- قسمت `remove user`:

Input:

User ID

Output:

در صورتی که کاربر وجود نداشته باشد پیغام

User doesn't exist.

و در غیر این صورت پیغام زیر چاپ شود.

User removed.

- قسمت `remove account`:

Input:

Account serial

Output:

در صورتی که حساب وجود نداشته باشد پیغام

Account doesn't exist.

و در غیر این صورت پیغام زیر چاپ شود.

Account removed.