

به نام خدا

تمرین پنجم درس برنامه‌نویسی پیشرفته

۰. فایل مربوط به توضیحات نحوه ارسال تمرین‌ها را که در مودل قرار دارد، مطالعه کنید.

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 باشد، بارگذاری نمایید.

StudentNumber_FirstName_LastName.zip

9031806_Mohammad_Ahmadpanah.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.

مهلت تحویل: تا یک‌شنبه ۲۷ اسفند ۱۳۹۶ ساعت ۵:۲۳

سوال اول

جاهای خالی را با کلمه مناسب پر کنید.

- A) Superclass's _____ members are accessible only to itself and its subclasses.
- B) When an object of a subclass is instantiated, a superclass _____ is called implicitly or explicitly.
- C) Subclass constructors can call superclass constructors via the _____ keyword.
- D) All classes in Java inherit directly or indirectly from the _____ class.

سوال دوم

درستی یا نادرستی عبارات زیر را با ذکر دلیل مشخص کنید.

- A) Superclass constructors are not inherited by subclasses.
- B) When a subclass redefines a superclass method by using the same signature, the subclass is said to overload that superclass method.
- C) Inheritance improves code reusability in subclasses while substitution reduces code duplication in client classes.

سوال سوم

مفاهیم زیر را به اختصار توضیح دهید:

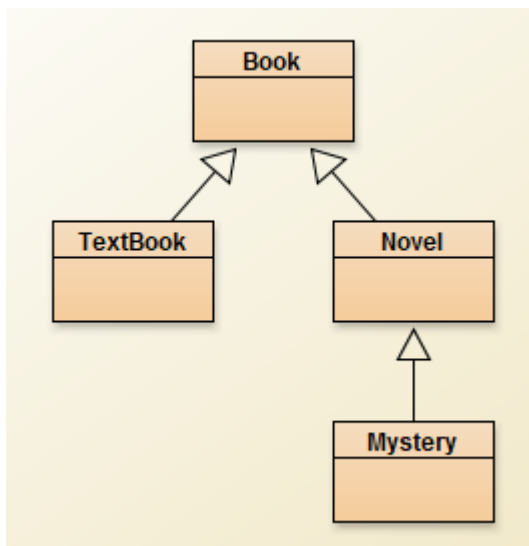
Polymorphism:

Overriding:

Substitution:

Abstract Class:

Interface:



سوال چهارم

با در نظر گرفتن نمودار کلاس (Class Diagram) مقابل به پرسش‌های زیر پاسخ دهید.

الف) مجاز بودن یا نبودن هر یک از دستورات زیر را با ذکر دلیل مشخص کنید.

- i. `Book b1 = new TextBook();`
- ii. `Book b2 = new Mystery();`
- iii. `Novel n1 = new Book();`
- iv. `Novel n2 = new Mystery();`
- v. `Mystery m1 = new Novel();`

ب) با فرض دستورات زیر

```
Book b1 = new Book();  
Book b2 = new Book();  
Mystery m1 = new Mystery();  
Novel n1 = new Novel();  
TextBook t1 = new TextBook();
```

مجاز بودن یا نبودن هر یک از دستورات زیر را با ذکر دلیل مشخص کنید.

- i. `b1 = n1;`
- ii. `n1 = b1;`
- iii. `b2 = m1;`
- iv. `t1 = n1;`
- v. `n1 = m1;`
- vi. `m1 = b1;`

سوال پنجم

فرض کنید کلاس Account به صورت زیر نوشته شده است. از این کلاس به عنوان کلاس پایه استفاده کرده و این سوال را حل کنید. تغییر کلاس پایه مجاز نیست.

```
public class Account {
    private long balance;      // The current balance
    private long accountNumber; // The account number

    public Account(long num) {
        balance = 0;
        accountNumber = num;
    }

    public void deposit(long amount) {
        if (amount > 0)
            balance += amount;
        else
            System.err.println("Invalid deposit amount!");
    }

    public void withdraw(long amount) {
        if (amount > 0 && amount <= balance)
            balance -= amount;
        else
            System.err.println("Invalid withdraw amount!");
    }

    public long getBalance() {
        return balance;
    }

    public long getAccountNumber() {
        return accountNumber;
    }

    public String toString() {
        return "Account Number #" + accountNumber
            + "--> balance = " + balance;
    }
}
```

```
public final void print() {
    // Don't override this.
    // Override the toString method.
    System.out.println( toString() );
}
}
```

در این سوال می‌خواهیم یک برنامه برای یک بانک ساده و کوچک بنویسیم.

- کلاس‌های SavingsAccount و CurrentAccount را برای مدل‌سازی حساب‌های پس‌انداز و جاری پیاده‌سازی کنید. این کلاس‌ها از کلاس Account گسترش می‌یابند. کلاس SavingsAccount علاوه بر ویژگی‌های کلاس Account، یک فیلد به عنوان نرخ سود (interest rate) و یک متد برای واریز سود به مقدار موجودی حساب دارد. کلاس CurrentAccount یک فیلد به عنوان حداقل میزان موجودی (overdraft limit) دارد. مطمئن شوید متدهای لازم از کلاس Account را override کرده‌اید.
- کلاس Bank را طوری بنویسید که یک آرایه از Account‌ها داشته باشد. تمام نمونه‌ها در این آرایه می‌توانند از نوع SavingsAccount یا CurrentAccount باشند. متد addAccount را برای کلاس Bank به گونه‌ای پیاده‌سازی کنید که یک Account جدید را به آرایه‌ی Account‌ها اضافه کند. در کلاس Bank یک متد update نیز وجود دارد که با پیمایش روی آرایه Account‌ها:
 - برای Account‌های از نوع SavingsAccount، سود را واریز می‌کند.
 - برای Account‌های از نوع CurrentAccount، در صورتی که موجودی حساب از overdraft limit کمتر باشد، اطلاعات حساب را چاپ می‌کند.
- کلاس Main‌ای بنویسید که شامل متد main باشد. در متد main یک نمونه از Bank ساخته و دست‌کم پنج نمونه Account از هر یک از انواع SavingAccount و CurrentAccount ساخته و به آرایه Account‌ها اضافه کنید. برنامه خود را برای همه متدهای پیاده‌سازی‌شده با مقادیر مختلف تست کنید. تنوع در تست‌ها و در نظر گرفتن حالت‌های مختلف برای همه متدها الزامی است.

توجه! مستندسازی به کمک JavaDoc، کامنت‌گذاری و رعایت اصول کدنویسی خوانا برای همه کلاس‌های پیاده‌سازی‌شده الزامی است.

سوال ششم

یک کلاس abstract به نام Student و کلاس‌های GraduateStudent (دانشجوی کارشناسی ارشد) و UndergraduateStudent (دانشجوی کارشناسی) را تعریف کنید که کلاس Student subclass کلاس Student هستند. موارد زیر را در این کلاس‌ها پیاده‌سازی کنید:

- ۱) کلاس Student شامل فیلدهای نام، نام خانوادگی و شماره دانشجویی باشد.
- ۲) کلاس UndergraduateStudent شامل یک فیلد ثابت برای تعداد کل واحدهایی است که دانشجو باید پاس کند (۱۴۰ واحد) و یک فیلد برای سهمیه ورودی وی است. دانشجوی کارشناسی می‌تواند از طریق کنکور، سهمیه المپیاد، انتقال هیئت علمی دائم، انتقال هیئت علمی موقت و مهمان وارد دانشگاه شود.
- ۳) کلاس GraduateStudent شامل یک فیلد ثابت برای تعداد کل واحدهایی است که دانشجو باید پاس کند (۳۲ واحد)، یک فیلد برای نام استاد راهنما و یک فیلد برای نام دانشگاه مقطع کارشناسی است.
- ۴) تمام فیلدها را به صورت private تعریف کنید و constructorهای متناسب با این ۳ کلاس را پیاده‌سازی کنید. (آیا می‌توان برای کلاس Student نیز constructor نوشت؟)
- ۵) توابع getter و setter را در تمام کلاس‌ها پیاده‌سازی کنید. (آیا برای فیلدهای ثابت نیازی به setter وجود دارد؟)
- ۶) یک متد displayInfo در کلاس Student تعریف کنید که نام، نام خانوادگی و شماره دانشجویی را نمایش دهد.
- ۷) متد displayStudentInformation را در دو کلاس UndergraduateStudent و GraduateStudent ایجاد نمایید که تمام مشخصات (همه فیلدها) را چاپ کند.
- ۸) آیا نمونه‌های ساخته‌شده از کلاس‌های UndergraduateStudent و GraduateStudent می‌توانند مقدار شماره دانشجویی (که به صورت private در کلاس Student تعریف شده است) را تغییر دهند؟ از طریق setter آن چطور؟
- * در کلاس Main، برنامه‌ای بنویسید تا همه موارد گفته‌شده در بالا را با حداقل دو دانشجوی مختلف از هر مقطع تحصیلی تست کند. تنوع در تست‌ها و در نظر گرفتن حالت‌های مختلف برای همه متدها الزامی است.

توجه! مستندسازی به کمک JavaDoc، کامنت‌گذاری و رعایت اصول کدنویسی خوانا برای همه کلاس‌های پیاده‌سازی شده الزامی است. همچنین برای این تمرین علاوه بر فایل‌های کد، یک فایل متنی در قالب PDF برای توضیحات و شرح دلایل ارائه کنید.

سوال هفتم

یکی از راه‌های فراهم کردن پیاده‌سازی‌های استاندارد در جاوا، استفاده از اینترفیس (interface)‌هایی است که این زبان معرفی کرده است. در سوال چهارم تمرین چهارم یک صف را پیاده‌سازی کرده بودید. در این تمرین، صفی بر اساس کلاس abstract پیوسته شده پیاده‌سازی کنید. این کلاس نوعی از اینترفیس استاندارد جاوا ([Queue Interface](#)) است. برای پیاده‌سازی صف، کافیت تا کلاس SimplePersonQueue را تغییر دهید. به همه کامنت‌های موجود در این کلاس توجه کنید، قسمت‌های لازم را تغییر داده و پیاده‌سازی این کلاس را تکمیل کنید. ضمناً به کلاس PersonQueue دست نزنید!

سپس در کلاس Main، علاوه بر کدهای نوشته شده، دستورات دیگری را نیز اضافه کنید تا مطمئن شوید که همه متدهای موجود در کلاس SimplePersonQueue را به طور کامل تست کرده‌اید. تنوع در تست‌ها و در نظر گرفتن حالت‌های مختلف برای همه متدها الزامی است.

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