

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 (rar != zip) که به قالب زیر نام‌گذاری شده باشد، بارگذاری نمایید.

StudentNumber\_FirstName\_LastName.zip

9031066\_Ehsan\_Edalat.zip

۲. در سوال‌هایی که ورودی و خروجی مطلوب آن‌ها مشخص شده است، برنامه‌ی شما به صورت ماشینی تصحیح می‌شود. بنابراین 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.

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

مهلت تحویل: تا جمعه ۱۰ آبان ۱۳۹۸ ساعت ۲۳:۵۵

سوال اول

جاهای خالی را با کلمه مناسب پر کنید.

- a) Superclass's \_\_\_\_\_ members are accessible to itself and its subclasses (and any other classes in that package, not outside).
- 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.

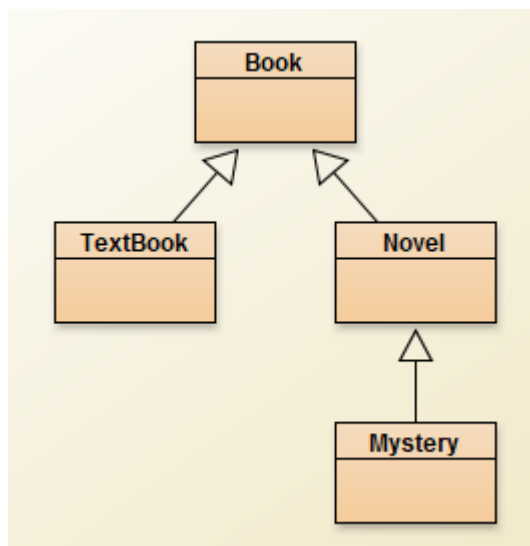
سوال سوم

مفاهیم زیر را به اختصار توضیح دهید:

- a) Polymorphism:
- b) Overriding:
- c) Substitution:
- d) Abstract Class:
- e) 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 کلاسی به عنوان وسیله نقلیه به اسم Vehicle به صورت زیر نوشته شده است. فیلدها و متدهای این کلاس را در کد زیر مشاهده می‌کنید. این کلاس را به عنوان کلاس پایه در نظر بگیرید و با استفاده از آن به سوال‌ها پاسخ دهید. توجه کنید که این کلاس را به هیچ عنوان تغییر ندهید.

```
public class Vehicle {  
  
    private int topSpeed;  
    private int currentSpeed;  
    private String name;  
  
    public Vehicle(int topSpeed, String name) {  
  
        this.topSpeed = topSpeed;  
  
        this.name = name;  
  
        this.currentSpeed = 0;  
  
    }  
    public void speedUpBy(int speedUpValue) {  
  
        if(speedUpValue >= 0) {  
  
            if ((currentSpeed + speedUpValue) <= topSpeed) {  
  
                currentSpeed += speedUpValue;  
  
            } else {  
  
                currentSpeed = topSpeed;  
  
            }  
  
        }else System.out.println("Invalid speed up value");  
  
    }  
    public void breakBy(int speedBreakValue){  
  
        if(speedBreakValue >= 0) {  
  
            if ((currentSpeed - speedBreakValue) >= 0) {  
  
                currentSpeed -= speedBreakValue;  
  
            } else {  
  
                currentSpeed = 0;  
  
            }  
  
        }  
  
    }  
}
```

```

    }else System.out.println("Invalid speed break value");

}
public final void print(){

    System.out.println(toString());

}
public String toString(){

    return "Vehicle " + name + " is moving | speed : " + currentSpeed + " / " + topSpeed;

}
public int getTopSpeed() {

    return topSpeed;

}
public int getCurrentSpeed() {

    return currentSpeed;

}
public String getName() {

    return name;

}
}

```

در این سوال می‌خواهیم یک سیستم کنترل وسایل نقلیه خیلی ساده پیاده‌سازی کنیم.

- دو کلاس GroundVehicle و AirVehicle به عنوان وسیله زمینی و وسیله هوایی پیاده‌سازی کنید. هر دوی این کلاس‌ها از کلاس Vehicle گسترش پیدا می‌کنند.
- کلاس AirVehicle علاوه بر ویژگی‌های Vehicle دارای فیلدها و متدهای زیر می‌باشد:
- فیلد maxFlyingHeight از نوع int که نشان دهنده بیشینه ارتفاع پروازی وسیله نقلیه است.
- فیلد currentFlyingHeight از نوع int که نشان دهنده ارتفاع پروازی فعلی وسیله نقلیه است.
- فیلد timeToDescendByOneMeter از نوع int که نشان دهنده مدت زمانی بر حسب ثانیه است که طول می‌کشد تا وسیله نقلیه به اندازه یک متر ارتفاع خود را کم کند.
- متدهایی با نام Ascend و Descend برای افزایش و کاهش ارتفاع وسیله نقلیه مشابه سرعت برای این کلاس پیاده‌سازی کنید.
- متدی برای محاسبه زمان مورد نیاز برای فرود این وسیله نقلیه به اسم calculateTimeToLand نیز پیاده‌سازی کنید.
- کلاس GroundVehicle علاوه بر ویژگی‌های Vehicle دارای فیلدها و متدهای زیر نیز می‌باشد:

- فیلد maxGear از نوع int که نشان دهنده مقدار بیشینه دنده ماشین است.
- فیلد currentGear از نوع int که نشان دهنده مقدار فعلی دنده ماشین است.
- برای این کلاس متدهای افزایش و کاهش سرعت را طوری override کنید که میزان دنده نیز در افزایش یا کاهش سرعت موثر باشد. یعنی با تغییر در سرعت ماشین، تغییر متناسب را در دنده نیز اعمال کنید. برای این کار می‌توانید از جدول‌های زیر کمک بگیرید. بیشینه دنده را ۵ در نظر بگیرید. محدودیتی در نوع پیاده‌سازی اطلاعات جدول و نسبت‌ها در کد وجود ندارد و با هر کالکشنی که فکر می‌کنید پیاده‌سازی بهتر انجام می‌شود، آن را پیاده کنید.

محدوده سرعت	نحوه تغییر دنده
بیشتر از صفر کیلومتر بر ساعت	۱→۲
بیشتر از ۱۳ کیلومتر بر ساعت	۲→۳
بیشتر از ۲۷ کیلومتر بر ساعت	۳→۴
بیشتر از ۷۵ کیلومتر بر ساعت	۴→۵

محدوده سرعت	نحوه تغییر دنده
کمتر از ۴۷ کیلومتر بر ساعت	۲→۱
کمتر از ۱۰۰ کیلومتر بر ساعت	۳→۲
کمتر از ۱۵۰ کیلومتر بر ساعت	۴→۳
کمتر از ۱۸۲ کیلومتر بر ساعت	۵→۴

- متد toString() را برای کلاس‌های فرزند به صورت متناسب با تغییرات در فیلدها override کنید.
- در نهایت کلاسی با نام VehicleSystem پیاده‌سازی کنید که دارای لیستی از Vehicle‌ها است. تمام اعضای این لیست می‌توانند از نوع GroundVehicle یا AirVehicle باشند. متدی به اسم addVehicle پیاده‌سازی کنید که یک Vehicle جدید را به لیست فعلی Vehicle‌ها اضافه کند.
- یک متد به نام status پیاده‌سازی کنید که با پیمایش روی لیست Vehicle‌ها :
  - برای وسایل نقلیه هوایی (AirVehicle)، تمام اطلاعات به اضافه زمان مورد نیاز برای فرود را محاسبه و چاپ کند .
  - برای وسایل نقلیه زمینی (GroundVehicle) تمام اطلاعات به اضافه میزان سرعت مورد نیاز برای رفتن به دنده بالاتر و میزان کاهش سرعت برای رفتن به دنده پایین‌تر را محاسبه و چاپ کند.

برای برنامه کلاس Main را ایجاد کرده و یک VehicleSystem در آن بسازید. حداقل ۴ تا از هر کدام از وسیله‌های نقلیه ایجاد کنید و تمامی متدها و پیاده‌سازی‌ها را با مقادیر مختلف تست کنید.

**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. علاوه بر آن، کلاس‌های این تمرین باید در پکیج HW55 پیاده‌سازی شوند.

## سوال ششم

یک کلاس abstract به نام Computer و کلاس‌های Mobile (تلفن همراه) و PC (کامپیوتر شخصی) را تعریف کنید که subclass کلاس Computer هستند. موارد زیر را در این کلاس‌ها پیاده‌سازی کنید:

- (۱) کلاس Computer شامل فیلدهای فرکانس cpu، حجم رم، حجم حافظه دیسک است.
  - (۲) کلاس Mobile شامل فیلدهای کیفیت دوربین (به مگاپیکسل)، تعداد درگاه سیم‌کارت و بیشینه حجم SD Card (به گیگابایت) است.
  - (۳) کلاس PC شامل فیلد ثابت برای نوع معماری کامپیوتر با مقدار رشته‌ی «CISC» است. همچنین، دارای فیلد رشته‌ای سازنده پردازنده گرافیکی (برای مثال NVIDIA) است.
  - (۴) تمام فیلدها را به صورت private تعریف کنید و constructorهای متناسب با این ۳ کلاس را پیاده‌سازی کنید. (آیا می‌توان برای کلاس Computer 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 Computer تعریف کنید که حجم رم، حجم حافظه دیسک را نمایش دهد.
  - (۷) متد displayComputerInformation را در دو کلاس Mobile و PC ایجاد نمایید که تمام مشخصات سیستم (همه فیلدها) را چاپ کند.
  - (۸) آیا نمونه‌های ساخته‌شده از کلاس‌های PC و Mobile می‌توانند مقدار فرکانس cpu (که به صورت private در کلاس Computer تعریف شده است) را تغییر دهند؟ از طریق setter آن چگونه؟
- \* در کلاس Main، برنامه‌ای بنویسید تا همه موارد گفته‌شده در بالا را با حداقل دو mobile مختلف و دو PC مختلف 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 پیاده‌سازی‌شده الزامی است. همچنین برای این تمرین علاوه بر فایل‌های کد، یک فایل متنی در قالب PDF برای توضیحات و شرح دلایل ارائه کنید. علاوه بر آن، کلاس‌های این تمرین باید در پکیج HW56 پیاده‌سازی شوند.



## سوال هفتم

یکی از راه‌های فراهم کردن پیاده‌سازی‌های استاندارد در جاوا، استفاده از اینترفیس (interface)هایی است که این زبان معرفی کرده است. در این تمرین، صفی بر اساس کلاس abstract پیوست‌شده 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 نوشته‌شده، 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 پیاده‌سازی‌شده الزامی است. همچنین برای این تمرین علاوه بر فایل‌های کد، یک فایل متنی در قالب PDF برای توضیحات و شرح دلایل ارائه کنید. علاوه بر آن، کلاس‌های این تمرین باید در پکیج HW57 پیاده‌سازی شوند.

## سوال هشتم (امتیازی)

در این سوال شما باید در مورد الگوهای طراحی (Design Pattern)های گفته‌شده تحقیق کنید. برای هر مورد باید علت استفاده از الگوی طراحی (Intent و Motivation) را بیان کنید. همچنین، با یک مثال مزیت استفاده از الگوی مورد نظر را توضیح دهید و آن را پیاده‌سازی کنید. علاوه بر آن، نمودار کلاس (Class Diagram) مرتبط با آن را رسم کنید. در نهایت، تفاوت پیاده‌سازی با استفاده از الگو و بدون آن را بیان کنید.

- Adapter
- Builder
- Façade

**توجه:** برای انجام این تمرین می‌توانید به کتاب [Design Patterns](#) مراجعه کنید. کپی کردن مستقیم مطالب کتاب و یا اسکرین‌شات از آن، هیچ نمره‌ای نخواهد داشت. پاسخ شما باید حاصل مطالعه موضوعات کتاب و درک شما از آن‌ها باشد.

**توجه!** مستندسازی به کمک JavaDoc، کامنت‌گذاری و رعایت اصول کدنویسی خوانا برای همه کلاس‌های پیاده‌سازی‌شده الزامی است. همچنین برای این تمرین علاوه بر فایل‌های کد، یک فایل متنی در قالب PDF برای توضیحات و شرح دلایل ارائه کنید. علاوه بر آن، کلاس‌های این تمرین باید در پکیج HW58 پیاده‌سازی شوند.