

به نام خدا

## تمرین پنجم 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-HW5-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، که با هر ساعت تاخیر یک درصد از سقف نمره قابل کسب، کسر می‌شود. برای مثال در صورت ارسال با ۱۰ ساعت تاخیر، سقف نمره قابل دریافت ۹۰ درصد است.

مهلت تحویل: تا چهارشنبه ۳۱ فروردین ۱۴۰۱ ساعت ۲۳:۵۹ شب

صفحه

## فهرست سوالات

- سوال اول..... ۳
- سوال دوم..... ۳
- سوال سوم..... ۵
- سوال چهارم (بازی چراغ قرمز چراغ سبز)..... ۶
- سوال پنجم..... ۹

## سوال اول

به سوالات زیر پاسخ دهید

الف) مزیت‌های ارث‌بری در زبان Java چیست؟

ب) چرا در بعضی از زبان‌ها از جمله Java، یک کلاس نمی‌تواند از چند کلاس به‌طور همزمان ارث‌بری کند؟

ج) عبارت کلیدی super در زبان Java چیست و به چه منظور استفاده می‌شود؟

د) تفاوت بین Inheritance و Abstraction در زبان Java چیست؟

ه) تفاوت بین method overriding و method overloading چیست؟

و) اگر یک class به‌طور همزمان دو اینترفیس را که هر دوی آن‌ها یک متد با name و signature یکسان دارند را پیاده‌سازی کند، چه اتفاقی رخ خواهد داد؟

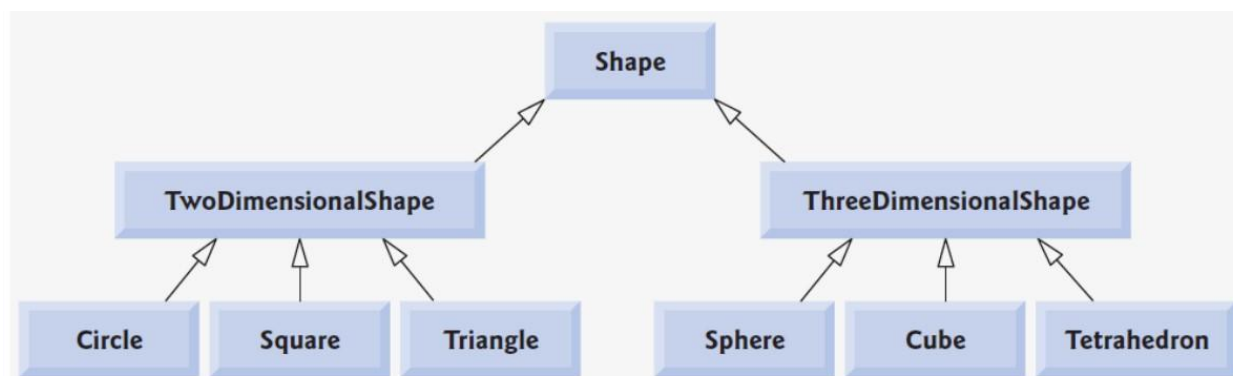
## سوال دوم

### بخش اول

Diamond Problem را با یک مثال توضیح دهید.

### بخش دوم

با توجه به نمودار زیر به سوالات ذکر شده پاسخ دهید.



نمودار ارث‌بری کلاس‌ها

الف) درستی یا نادرستی دستورات زیر را با ذکر دلیل مشخص کنید.

- Shape shape = new Sphere();
- Square square = new TwoDimensionalShape();
- ThreeDimensionalShape tds = new Triangle();
- ThreeDimensionalShape tds = new Cube();
- Tetrahedron tetrahedron = new Shape();

ب) ابتدا دستورات زیر را در نظر بگیرید:

- Circle c1 = new Circle();
- TwoDimensionalShape t1 = new Triangle();
- Shape s1 = new Shape();
- ThreeDimensionalShape t2 = new ThreeDimensionalShape();
- Shape s2 = new TwoDimensionalShape();
- Shape s3 = new Cube();
- Sphere s4 = new Sphere();
- Tetrahedron t3 = new Tetrahedron();

حال درستی یا نادرستی دستورات زیر را با ذکر دلیل مشخص کنید.

- c1 = t1;
- s1 = t2;
- s2 = t3;
- t2 = s3;
- s4 = t3;
- t2 = t1;

## سوال سوم

الف) جای خالی را با عبارت 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) A method that is declared \_\_\_\_\_ cannot be overridden in a subclass.
- e) \_\_\_\_\_ in Java is the ability of an object to take many forms..
- f) Objects of \_\_\_\_\_ classes cannot be instantiated.
- g) Methods declared in an interface are by default \_\_\_\_\_.
- h) In Java we can only use multiple inheritance for \_\_\_\_\_.
- i) In inheritance we can declare members of a class as \_\_\_\_\_. So that subclasses won't be able to inherit them.
- j) A complete class that has an implementation of all its methods and can be instantiated is called \_\_\_\_\_ class.

ب) صحیح یا غلط بودن عبارات زیر را مشخص کنید.

- 1) When a subclass redefines a superclass method by using the same signature, the subclass is said to overload that superclass method.
- 2) An abstract class has no use unless it is extended by some other class.
- 3) A constructor of an abstract class is called when an instance of an inherited class is created.
- 4) In case of inheritance, constructors are called from the top to down of the inheritance hierarchy.
- 5) We access subclass members if we create an object of superclass.
- 6) Trying to invoke a subclass-only method with a superclass variable is not allowed.
- 7) Java is 100% object oriented.

## سوال چهارم (بازی چراغ قرمز چراغ سبز)

### توضیح بازی:

بازی شامل ۵ شخصیت است که در ابتدای بازی به صورت تصادفی بین بازیکنان پخش می‌شود (به هر بازیکن یک شخصیت). هر شخصیت دارای سرعت، جان اولیه و قابلیت‌های بخصوص خود است. با اعلام چراغ سبز بازیکنان می‌توانند حرکت مستقیم خود را در مسیر بازی شروع کنند. با اعلام چراغ قرمز همه بازیکنان متوقف می‌شوند و سپس ربات بازی، یکی از بازیکنان را به صورت تصادفی انتخاب کرده و به او ۲۵ واحد آسیب می‌رساند.

بازیکنانی که تا پایان ۱۰ راند بازی بتوانند مسافت ۲۰۰ متر را طی کنند، برنده می‌شوند.

### یک دور از بازی:

با گفته شدن عبارت «چراغ سبز» به بازیکنان فرصت دو ثانیه‌ای داده می‌شود تا حرکت مستقیم خود را انجام دهند. هر شخصیت به اندازه‌ی سرعتی که دارد مسافتی طی می‌کند و متوقف می‌شود. با گفتن عبارت «چراغ قرمز» زمان حرکت بازیکنان تمام می‌شود و ربات بازی یک بازیکن را بطور تصادفی انتخاب می‌کند و به او ۲۵ واحد آسیب می‌رساند.

سپس به یکی از بازیکنان اجازه داده می‌شود که از مهارت‌های ویژه خود استفاده کند. در ابتدا باید چک شود که بازیکن بیشتر از ظرفیت مهارت‌هایش استفاده نکرده باشد و سپس از کاربری که بازیکن را کنترل می‌کند پرسیده شود که می‌خواهد از مهارت خود را در این راند استفاده کند یا خیر.

بازیکنی که ۲۰۰ متر مسافت را طی کند و جان او برابر صفر شده باشد، دیگر نمی‌تواند به حرکت خود ادامه دهد و یا از مهارتش استفاده کند. توجه کنید که بازیکن نمی‌تواند بیشتر از ۲۰۰ متر طی کند.

زمانی که تمام بازیکنان زنده مسافت ۲۰۰ متر را طی کنند یا تعداد راندها ۱۰ شده باشد، بازی تمام می‌شود.

در پایان هر راند بازیکنان زنده و بازیکنانی که موفق به طی کردن ۲۰۰ متر شده‌اند، به تمامی بازیکنان اعلام شود.

### شخصیت‌های بازی:

• مرد قوی:

○ میزان جان: ۱۰۰ واحد

- سرعت: ۲۰ واحد در ۲ ثانیه
- مهارت ویژه: این شخصیت می‌تواند جان خود را از آسیب‌های ربات و بازیکن انتخابی دیگری، نجات دهد (با توجه به دفعات مجاز، در تنها ۲ دور از بازی).
- تعداد دفعات مجاز به استفاده از مهارت ویژه: ۲
- دختر جوان:
  - میزان جان: ۷۵ واحد
  - سرعت: ۲۵ واحد
  - مهارت ویژه: این شخصیت می‌تواند به یکی از بازیکنان انتخابی کاربر ۲۵ واحد آسیب برساند.
  - این شخصیت جز شخصیت‌های منفی بازی محسوب می‌شود.
  - تعداد دفعات مجاز به استفاده از مهارت ویژه: ۱
- مرد عصبانی:
  - میزان جان: ۷۵ واحد
  - سرعت: ۲۰ واحد
  - مهارت ویژه: این شخصیت می‌تواند به یکی از بازیکنان که در کمتر از ۱۰ متری او قرار دارند ۵۰ واحد آسیب برساند.
  - تعداد دفعات مجاز به استفاده از مهارت ویژه: ۱
  - این شخصیت جز شخصیت‌های منفی بازی محسوب می‌شود.
- پیرمرد:
  - میزان جان: ۷۵ واحد
  - سرعت: ۲۰ واحد
  - مهارت ویژه: این شخصیت می‌تواند سرعت خود را در راند بعد بازی دو برابر کند ولی پس از آن سرعت او دو واحد کمتر از سرعت اولیه می‌شود.
  - تعداد دفعات مجاز به استفاده از مهارت ویژه: ۲

- مرد باهوش:

- میزان جان: ۷۵ واحد
- سرعت: ۲۰ واحد
- مهارت ویژه: این شخصیت می‌تواند جان خود را ۲۵ واحد بهبود ببخشد.
- تعداد دفعات مجاز به استفاده از مهارت ویژه: ۲

### نکات پیاده‌سازی

- در پیاده‌سازی شخصیت‌ها باید از ساختارهای چندریختی استفاده شود.
- در صورتی که ورودی غیر معتبر بود پیغام مناسب چاپ شود ولی برنامه تمام نشود.
- مستندسازی به کمک Javadoc، کامنت‌گذاری و رعایت اصول کدنویسی خوانا برای همه‌ی کلاس‌های پیاده‌سازی شده الزامی است.



## سوال پنجم

در این تمرین می‌خواهیم یک وبسایت دانلود فیلم و سریال را شبیه‌سازی کنیم. در این وبسایت یک موجودیت با عنوان کاربر کلی (GeneralUser) داریم. این موجودیت یک نام کاربری، گذرواژه و تاریخ عضویت دارد. همچنین می‌تواند لیست فیلم‌ها را به تفکیک ژانر مشاهده کند. این کاربر کلی خود به دو دسته ادمین (Admin) و کاربر (User) تقسیم می‌شود (بنابراین توجه داشته باشید که از موجودیت GeneralUser به طور مستقیم شیءای ساخته نمی‌شود، بلکه هر کاربر کلی یا Admin است یا User).

۱. ادمین: قابلیت حذف کردن و افزودن فیلم‌ها را دارد.
۲. کاربر: قابلیت دانلود فیلم‌ها را دارد. خود به دو دسته کاربر عادی و VIP تقسیم می‌شود. الف) کاربر عادی (NormalUser): در کل امکان دانلود ۵ فیلم را داشته و برای دانلود فیلم‌های بیش‌تر باید به کاربر VIP ارتقاء پیدا کند. به علاوه پیش از دانلود هر فیلم باید ۱۰ ثانیه منتظر بماند. ب) کاربر VIP (VIPUser): محدودیتی برای دانلود فیلم‌ها ندارد.

فیلم (Film) موجودیت دیگری است که در این وبسایت نقش مهمی دارد و می‌تواند از دو منظر دسته‌بندی شود:

۱. هر فیلم می‌تواند فیلم سینمایی (Movie) و یا سریال (Series) باشد.
  ۲. هر فیلم می‌تواند انیمیشن (Animation) و یا غیر انیمیشن (LiveAction) باشد.
- به علاوه هر فیلم به طور کلی ویژگی‌های زیر را دارد:
- نام فیلم
  - خلاصه‌ای از فیلم
  - ژانر فیلم: یکی از مقادیر [COMEDY - ACTION - DRAMA - DOCUMENTARY - ADVENTURE] را دارا خواهد بود.
  - نام کارگردان
  - لیست هنرپیشه‌ها (شامل بازیگران و یا صدایپیشه کاراکترهای انیمیشن)
  - رده سنی: یکی از مقادیر [A - B - C - D] را دارا خواهد بود. A برای افراد زیر ۵ سال، B برای افراد ۱۰ تا ۱۵ سال، C برای افراد ۱۵ تا ۱۸ سال و D برای افراد بالای ۱۸ سال.
- هر فیلم سینمایی یک مدت داشته (برای تعریف این قسمت به صورت String مانعی وجود ندارد اما می‌توانید آن را با هر کلاس دلخواه دیگر نیز تعریف کنید) و هر سریال تعدادی فصل و هر فصل تعدادی قسمت دارد.

برای دانلود انیمیشن‌ها پیش از دانلود گزینه‌ای وجود دارد که به کاربران کمک می‌کند تا با وارد کردن سن بررسی کنند که انیمیشن فوق برای آن رده سنی مناسب است یا خیر. همچنین برای فیلم‌های غیر انیمیشن امکان دانلود زبان اصلی، دوبله و یا با زیرنویس چسبیده وجود دارد. برای هر یک از انواع این دانلودها متد جداگانه‌ای با پیغام متفاوت در نظر بگیرید. حال موجودیت‌های توضیح داده شده در بالا را پیاده‌سازی کرده و در کلاس Main برنامه منوهای مناسبی برای استفاده از قابلیت‌های بالا ایجاد کنید. در ابتدای برنامه یک منوی Sign in و Sign up نمایش داده می‌شود که کاربر می‌تواند با استفاده از آن به عنوان یکی از حالات کاربر یا ادمین ثبت‌نام کرده یا وارد حساب کاربری خود شود. برای مثال:

```
[1] Sign up
```

```
[2] Sign in
```

```
> 1
```

```
[1] As user
```

```
[2] As admin
```

```
> 2
```

```
> Username: adminusername
```

```
> Password: 12345
```

```
> Confirm Password: 12345
```

```
Admin account created successfully!
```

و به همین ترتیب برای قسمت‌های دیگر. حال در ادامه:

۱. در صورتی که کاربر به عنوان ادمین وارد شد گزینه‌های زیر به او نمایش داده می‌شوند:  
الف. افزودن فیلم: اطلاعات فیلم به عنوان ورودی از کاربر دریافت شده و فیلم جدید ثبت می‌شود.  
ب. حذف فیلم: نام فیلم از کاربر دریافت شده و فیلم مشخص شده حذف می‌شود.  
ج. مشاهده لیست فیلم‌ها: ژانر مورد نظر دریافت شده و تمام فیلم‌های آن ژانر با اطلاعات کامل آن‌ها نمایش داده می‌شوند (برای پیاده‌سازی این قسمت از متد toString استفاده کنید). در صورت وارد کردن عبارت NONE تمام فیلم‌ها نمایش داده می‌شوند.  
د. خروج: کاربر از حساب کاربری خود خارج می‌شود.

```
[1] Add film
```

```
[2] Delete film
```

```
[3] List films by genre
```

```
[4] Sign out
```

۲. در صورتی که کاربر به صورت کاربر وارد شد گزینه‌های زیر به او نمایش داده می‌شوند:
- الف. مشاهده لیست فیلم‌ها: ژانر مورد نظر دریافت شده و تمام فیلم‌های آن ژانر با اطلاعات کامل آن‌ها نمایش داده می‌شوند. در صورت وارد کردن عبارت NONE تمام فیلم‌ها نمایش داده می‌شوند.
- ب. دانلود فیلم: نام فیلم مورد نظر از کاربر دریافت شده و در صورتی که کاربر شرایط دانلود فیلم جدید را داشت فیلم دانلود می‌شود (برای شبیه سازی دانلود فیلم تنها چاپ یک عبارت که حاوی نام فیلم و همچنین نوع دانلود برای فیلم‌های غیر انیمیشن باشد کافیت و نیاز نیست کار خاصی انجام شود).
- ج. ارتقاء به کاربر VIP: این گزینه تنها در صورتی که کاربر یک کاربر عادی باشد نمایش داده خواهد شد. با انتخاب این گزینه کاربر می‌تواند کد "AUTAP" را وارد کرده و به کاربر VIP ارتقاء پیدا کند. در صورت نادرست بودن کد خطای مناسبی داده می‌شود.
- د. خروج: کاربر از حساب کاربری خود خارج می‌شود.

- [1] List films by genre
- [2] Download film
- [3] Upgrade to VIP account
- [4] Sign out

**امتیازی:** بخشی به برنامه خود اضافه کنید که هر یک از کاربران بتوانند به هر یک از فیلم‌ها امتیازی از ۱ تا ۵ بدهند و این امتیاز زمان چاپ شدن اطلاعات هر فیلم نمایش داده شود. توجه کنید که هر کاربر تنها یک‌بار می‌تواند به هر فیلم امتیاز دهد.

**امتیازی:** بخشی به برنامه خود اضافه کنید که زمان دانلود هر فیلم، لیست فیلم‌های مشابه نیز به کاربر نمایش داده شود. این مشابهت می‌تواند بر اساس نام کارگردان، لیست هنرپیشه‌ها، ژانر فیلم و نام فیلم باشد. پس از جستجوی لیست فیلم‌ها بر این اساس، باید لیست ۵ فیلمی که بیشترین مشابهت با فیلم دانلود شده را دارند نمایش داده شوند (برای پیاده سازی این قسمت الگوریتم یکتایی مدنظر نیست و هر الگوریتمی که به درستی لیست فیلم‌ها را بررسی کند قابل قبول است).

\* برای یک پیاده‌سازی مناسب در هر مرحله فکر کنید که برای تعریف هر یک از فیلدها، از میان primitive type، کلاس‌ها، enum، interface و یا ... کدام یک برای استفاده مناسب‌تر است. رعایت اصول شیء‌گرایی، استفاده مناسب از ارث‌بری و interface و همچنین رعایت کامنت‌گذاری و جاوا‌داک برای این بخش الزامی بوده و در صورت عدم رعایت آن، نمره‌ای تعلق نخواهد گرفت.