

بیلیاردک!

مقدمه

در این سوال قصد داریم یک بازی بیلیارد را به صورت ساده تر شبیه سازی کنیم. شکل کلی بازی در محیطی انجام می شود که از نمای بالای میز ما با چوب خود به توپ های روی میز ضربه ای را وارد می کنیم. در مورد ابزار های موجود در بازی می دانیم:

میز یک مستطیل با ابعاد دلخواه است. (عمودی یا افقی بودن این میز هم دلخواه می باشد). این میز دارای 6 حفره در گوشه ها و وسط طول ها می باشد. توجه کنید که حفره ها دایره هایی هستند که در گوشه ها ربع دایره داخل مستطیل زمین و در طول ها نیم دایره داخل مستطیل زمین هستند. شعاع این حفره ها از شعاع توپ ها بزرگ تر و با اندازه دلخواه هستند.

توپ ها به سه دسته تقسیم می شوند:

۱. توپ سفید که ما با این توپ به دیگر توپ ها ضربه وارد می کنیم. این توپ شماره ندارد.

۲. توپ سیاه که دارای شماره 8 است.

۳. 9 توپ رنگی (با رنگ دلخواه) که با شماره های 1 تا 7 ، شماره 9 و 10 شماره گذاری شده است. توجه کنید در ادامه سوال، اصطلاح توپ رنگی فقط برای این توپ ها است و به دو توپ قبلی مرتبط نیست.

شعاع توپ ها مقادیری دلخواه اما کوچک تر از شعاع حفره ها هستند. امتیاز هر توپ رنگی که پاکت می شود (وارد حفره می شود)، ضریبی ثابت و دلخواه از شماره توپ است و برای توپ سیاه این ضریب دو برابر است.

چوب یک مستطیل با ابعاد دلخواه اما بسیار نازک است.

این بازی برای وارد کردن ضربه مجاز محدودیت تعداد ضربه دارد. (این تعداد دلخواه است). با تمام شدن این تعداد، به اندازه نصف امتیاز بیشترین توپ رنگی از امتیاز هر ضربه ی اضافی کم میشود.

مراحل بازی

در ابتدا 10 توپ شامل توپ های رنگی و سیاه به شکل مثلث با ترتیب دلخواه (می تواند رندوم یا هر ترتیب خاص دیگری باشد.) در یک طرف میز قرار دارند و توپ سفید در طرف دیگر میز قرار دارد. مانند شکل نمای بازی می باشد. ما قصد داریم با وارد کردن ضربه به توپ ها به وسیله وارد کردن ضربه به توپ سفید، توپ های رنگی را وارد حفره ها کنیم. به این منظور با برای هر ضربه دو عملیات را باید انجام بدیم؛ یک عملیات انتخاب جهت برای وارد کردن ضربه است (برای انتخاب جهت، همواره امتداد چوب از مرکز توپ سفید می گذرد) و دیگری انتخاب سرعت اولیه برای ضربه است. و به همین ترتیب بازیکن بازی را به جلو می برد. فرد زمانی می برد که تمام توپ ها را وارد حفره ها کرده باشد. و در دو حالت بازیکن می بازد؛ یا توپ سفید را وارد حفره می کند یا امتیاز فرد منفی می شود. امتیاز فرد در صفحه باید نشان داده شود و بعد از هر ضربه روزرسانی شود.

قوانین بازی

اگر توپی مثلاً سفید به یک توپ دیگر برخورد کند، توپ سفید با همان سرعت و توپ دیگر با سرعت توپ سفید در جهت مناسب بر می گردند. ولی اگر به دیواره های میز برخورد کند، مقدار ثابت دلخواهی از سرعت توپ سفید کم می شود. برای درک بهتر برای جهت مناسب برای بعد از برخورد به تصویر تعیین جهت بعد از برخورد با توپ دقت کنید. دو حالت خاص را هم برای فهم بهتر بررسی میکنیم:

۱. اگر مرکز سه دایره بالا (توپ سیاه و سفید در حالت قبل از ضربه و توپ سفید در حالت تماس با توپ سیاه) در یک راستا باشند، توپ سفید دقیقاً روی همان مسیر بر میگردد و توپ سیاه در همان راستا در جهت مخالف دور می شود.

۲. اگر توپ سفید در حالت تماس با توپ سیاه در دور ترین حالت ممکن باشد، توپ سفید به مسیر خود ادامه می دهد و توپ سیاه در راستای خط المرکزین توپ سیاه در حالت قبل از ضربه و توپ سفید در حالت تماس با توپ سیاه، دور می شود.

برای برخورد با دیوار هم برای تعیین جهت مناسب به تصویر تعیین جهت بعد از برخورد با دیوار توجه کنید. پاکت شدن توپ به معنای عبور مرکز کره ی توپ از روی میز به داخل حفره است و با عبور مرکز توپ، توپ پاکت می شود. فرض می کنیم توپ زمانی که پاکت می شود سریعاً از زمین خارج می شود.

امتیازی های سوال

مقدار نمره ی اضافی با توجه به سختی بخش تعیین می شود و حداکثر نمره ی اضافی این سوال بعدا اطلاع رسانی می شود. نمره امتیازی روی نمره تمرین های دیگر هم تاثیر دارد.

۱. ایجاد افکت گرافیکی برای چرخش توپ ها

۲. استفاده از تصاویر به جای شکل های هندسی (استفاده نکردن از شکل های هندسی نمره کامل این بخش حساب می شود ولی با توجه به نسبت استفاده از عکس در مقابل شکل ها میتوانید بخشی از این نمره را بگیرید)

۳. برقرار کردن قوانین فیزیکی به جای قوانین برخورد گفته شده (برخورد با توپ ها الاستیک و برخورد با دیواره ها غیر الاستیک و ثوابت دلخواه و منطقی انتخاب شوند).

۴. استفاده از افکت های صوتی

۵. روان بودن بازی و برخورد ها

۶. استفاده از نوار گرافیکی برای انتخاب سرعت اولیه هر ضربه

۷. اضافه کردن قابلیت نشان دادن قسمتی از ادامه ی مسیر در هنگام انتخاب جهت برای ضربه

۸. اضافه کردن ذخیره 5 امتیاز اخیر با اسم کاربر به هر روش دلخواه (در ابتدای بازی گرفتن اسم بازیکن باید اضافه شود).

۹. گرافیک جذاب و زیبا

سیستم شبیه ساز دانشگاه

دانشگاه

در این تمرین می خواهیم یک سیستم شبیه ساز دانشگاه طراحی کنیم که در آن دانشجویها و نمره هایشان را ثبت کنیم.

توجه داشته باشید که برای راحتی این تمرین در دو بخش اصلی توضیح داده شده بخش اول مربوط به شی گزایی مسئله است و بخش دوم مربوط به گرافیک آن ولی در نهایت خروجی گرافیکی باید تحویل داده شود.

بخش اول :

دانشجو

کلاس دانشجو با نام Student، خصوصیات زیر را دارد:

سازنده (Constructor)

- سازنده با یک ورودی از نوع String که شماره دانشجویی است.

متد های getter و setter

نوع	نام getter	نام setter	توضیح
String	getStudent	-	شماره دانشجویی
String	getName	setName	نام دانشجو
java.time.LocalDate	getBirthDate	setBirthDate	تاریخ تولد
Department	getDepartment	setDepartment	دانشکده

متدهای دیگر

- متد takeCourse برای گرفتن یک درس استفاده می شود. تنها ورودی این متد از نوع Course است.

- متد `finishCourse` زمانی صدا می شود که یک درس تمام شده باشد. این متد دو ورودی دارد یکی از نوع `Course` و دیگری از نوع `double` متد که نمره ی دانشجو در آن درس است. نمره ی وارد شده باید نامنفی بوده و حداکثر برابر 20 باشد، همچنین درس مورد نظر (ورودی اول متد) باید جزو درس هایی که دانشجو گرفته باشد و هنوز تمام نشده باشد.
- متد `getGradeReport` با خروجی از نوع `java.util.List<GradeReport>` . هر یک از اعضای لیست به ازای هر درسی که این دانشجو گرفته، یک `instance` از کلاس `GradeReport` است.
- متد `getCourses` با خروجی از نوع `java.util.List<Course>` این تابع لیست درس هایی که دانشجو اخذ کرده را خروجی می دهد.

دانشکده

کلاس دانشکده با نام `Department`، خصوصیات زیر را دارد:

سازنده (Constructor)

- سازنده با یک ورودی از نوع `String` که نام دانشکده است.

متد های `getter` و `setter`

نوع	نام <code>getter</code>	نام <code>setter</code>	توضیح
<code>String</code>	<code>getName</code>	<code>setName</code>	نام
<code>String</code>	<code>getId</code>	<code>setId</code>	کد دانشکده

متدهای دیگر

- متد `equals` با ورودی از نوع `java.lang.Object` و خروجی از نوع `boolean`. این متد در صورتی که شی ورودی، `null` نباشد و از نوع `Department` باشد، همچنین نام و کد آن با این شی برابر باشد `true` بازمی گرداند. در غیر این صورت `False` برمی گرداند.
- متد `getStudent` دانشجویهای دانشکده را در قالب `java.util.List<Student>` می دهد.
- متد `getFaculty` اساتید دانشکده را در قالب `java.util.List<Professor>` می دهد.

درس

کلاس درس با نام Course، خصوصیات زیر را دارد:

سازنده (Constructor)

- سازنده با یک ورودی از نوع String که نام درس است.

متدهای getter و setter

نوع	نام getter	نام setter	توضیح
String	getName	setName	نام درس
int	getCredits	setCredits	تعداد واحد درس
Department	getDepartment	setDepartment	دانشکده ارائه دهنده
Professor	getInstructor	setInstructor	استاد ارائه دهنده

توضیح : تعداد واحد درس باید مثبت و کوچک تر مساوی 6 باشد. مقدار پیش فرضش هم برابر 3 است.

متدهای دیگر

- متد equals با ورودی از نوع java.lang.Object و خروجی از نوع boolean. این متد در صورتی که شیء ورودی، null نباشد و از نوع Course باشد، همچنین نام و تعداد واحد و دانشکده و استاد آن با این شیء برابر باشد true باز می گرداند. در غیر این صورت False بر می گرداند.

استاد

- کلاس استاد با نام Professor، خصوصیات زیر را دارد:

سازنده (Constructor)

- سازنده با یک ورودی از نوع String که نام استاد است.

متدهای getter و setter

نوع	نام getter	نام setter	توضیح
String	getName	setName	نام استاد
java.time.LocalDate	getBirthDate	setBirthDate	تاریخ تولد
Department	getDepartment	setDepartment	دانشکده
AcademicRank	getRank	setRank	رتبه استاد

متدهای دیگر

- متد equals با ورودی از نوع java.lang.Object و خروجی از نوع boolean. این متد در صورتی که شی ورودی، null نباشد و از نوع Professor باشد، همچنین نام و تاریخ تولد و دانشکده و رتبه ی آن با این شیء برابر باشد true باز می‌گرداند. در غیر این صورت False برمی‌گرداند.
- متد getCourse با خروجی از نوع java.util.Set<Course>. این متد تمام درس هایی که تا به حال این استاد ارائه داده را برمی گرداند.

گزارش نمره

- کلاس گزارش نمره با نام GradeReport، خصوصیات زیر را دارد:

سازنده (Constructor)

- سازنده با دو ورودی، یکی از نوع Student که دانشجوی مربوطه است و دیگری از نوع Course که درس مربوطه است.

متدهای setter و getter

نوع	نام getter	نام setter	توضیح
Student	getStudent	-	دانشجوی مربوطه
Course	getCourse	-	درس مربوطه
double	getGrade	setGrade	نمره دانشجو در آن درس

تعداد صفحات: 2

مدهای دیر

- متد toString با خروجی از نوع String، یک رشته با فرمت <course_name> -> : <student_id> grade<> است. <student_id> را با شماره دانشجویی، <course_name> را با نام درس و <grade> را با نمره جایگزین کنید.

مجموعه های شمارشی (Enum)

رتبه دانشگاهی با نام AcademicRank یکی از مقادیر PROFESSOR ، ASSISTANT_PROFESSOR یا ASSOCIATE_PROFESSOR می باشد.

بخش دوم :

در گرافیک این سوال ما 4 مود یا حالت اصلی داریم. در ابتدا کاربر می تواند یکی از این 4 حالت اصلی را انتخاب کند.

1- God mode:

این حالت برای خدایان طراحی شده و هر کاری در آن می توان انجام داد. در این وضعیت تمام اشیا باید به شکلی مرتب طبقه بندی شده باشند و امکان ساختن یک شی از هر نوعی از کلاس ها (کلاس هایی که واقعا تعدادی از اشیا از آن ها وجود دارد مانند دانشجویان، اساتید، دانشکده ها و ...) وجود دارد.

با کلیک کردن بر روی هر کدام از اشیا باید پنجره ای کوچک باز شود که در اطلاعات آن شی نوشته شده باشد.

شکل گرافیکی زیر صرفا برای فهم بهتر شما است و گرافیک این بخش را می توانید به میل خود طراحی کنید (هرچند تمام اطلاعات باید در یک پنجره و به شکل یک جا نمایش داده شود).

mode: god S P

STUDENTS

1

2

3

4

+ ADD

PROFESSORS

1

2

3

4

+ ADD

courses

1

2

3

4

+ ADD

details:

2- student mode:

این حالت برای استفاده دانشجویان طراحی شده. در این بخش دانشجو با وارد کردن نام و شماره دانشجویی خود وارد پنل شخصی‌اش می‌شود. در این بخش دانشجو می‌تواند تمام درس‌ها را مشاهده و اخذ کند و همچنین علاوه بر اطلاعات خود اطلاعات درس‌های خود، واحد‌های اخذ شده و نمره‌هایش را نیز مشاهده کند. بدیهی است که دانشجو اجازه ثبت نمره برای خود را ندارد.

3- professor mode:

این حالت برای استفاده اساتید طراحی شده. در این بخش استاد با وارد کردن نام و دپارتمان خود وارد پنل شخصی‌اش می‌شود. در این بخش یک استاد علاوه بر مشاهده نام همکاران خود می‌تواند لیستی از درس‌هایی که ارائه داده را مشاهده کرده و علاوه بر آن با کلیک بر روی هر درس می‌تواند لیست دانشجویانی که درس را اخذ کرده‌اند را مشاهده کند. استاد می‌تواند به هر دانشجو نمره دهد و یا دانشجو را از آن درس حذف کند. همچنین استاد قابلیت ارائه درس جدید را نیز دارد.

4- department officer mode:

این حالت برای استفاده مسئول آموزش دانشکده‌ها طراحی شده. در این بخش مسئول آموزش با وارد کردن نام دپارتمان خود وارد پنل دانشکده می‌شود. در این بخش مسئول آموزش می‌تواند تمام اساتید و درس‌هایی که در این دانشکده وجود دارند را مشاهده کند.

همچنین مسئول آموزش می‌تواند دانشجویان را از یک درس حذف و یا به یک درس اضافه کند. همچنین مسئول آموزش می‌تواند استاد، دانشجو و یا درس جدید اضافه کند.

توضیحات: توجه داشته باشید که کلاس‌ها، متدها و متغیرهایی که در بخش اول شرح داده شده‌اند لازم هستند اما لزوماً کافی نیستند لذا این اجازه را دارید که قابلیت‌های بیشتری را برای پیاده سازی گرافیکی اعمال کنید اما کلیت بخش اول نباید حذف شود.

شطرنج (امتیازی)

ایکس و ایگرگ دو شطرنج‌باز حرفه‌ای هستند که متأسفانه در جزیره‌ای دور افتاده به همراه شما و البته یک عدد لپ تاپ و تعدادی باتری اضافی لپ تاپ (: گیر افتاده اید و از آن جایی که در جزیره غذایی نیست، تصمیم دارند شما را به عنوان غذا بخورند. با این وجود از آن جایی که آن‌ها خیلی عاشق شطرنج؛ هستند ولی هیچ مهره شطرنجی همراهشان ندارند و از طرفی هم می‌دانند که شما برنامه نویسی بلدید، به شما قول داده اند که اگر بتوانید کد یگ بازی شطرنج؛ را برای آن‌ها بنویسید تا شطرنج بازی کنند، از خیر خوردن شما می‌گذرند و آن قدر بازی خواهند کرد تا خودشان از گرسنگی بمیرند! البته آن‌ها برای این که کارشان سریع تر راه بیفتد، یک سری قوانین شطرنج را هم بی خیال شده‌اند و صرفاً اگر مواردی که آن‌ها از شما می‌خواهند را پیاده سازی بکنید، از خیر خوردن شما خواهند گذشت.

همانطور که از بند بالا مشخص است، هدف شما در این سوال پیاده سازی بازی شطرنج خواهد بود. قوانین این بازی شطرنج تا حد زیادی مشابه قوانین شطرنج واقعی است و صرفاً چند قانون آن برای سادگی حذف شده است و یک مورد ویژگی متفاوت به آن اضافه شده است.

- در این نوع شطرنج چیزی به نام کیش و مات نداریم و صرفاً با زدن شاه، بازی به پایان می‌رسد. یعنی نیازی به انجام هیچ گونه چک برای کیش شدن و یا مات شدن نیست و خود بازیکنان اگر حواسشان نباشد و شاهشان در وضعیت تهدید باشد و زده شود، بازنده خواهند بود. پس از این لحاظ شما نیاز به بررسی وضعیت کیش و مات ندارید و فقط باید زده نشدن شاه بررسی شود.
- قلعه کردن و حرکت آن پاسبان در بازی وجود ندارند و نیازی به پیاده سازی آن‌ها نیست. (پیاده سازی آن‌ها نمره امتیازی هم ندارد)
- این موضوع که سربازها با رسیدن به آخر زمین تبدیل به وزیر و... بشوند وجود ندارد.
- در ابتدای شروع بازی تعداد مشخصی حرکت به عنوان لیمیت تعیین خواهد شد. در صورتی که این محدودیت به اتمام برسد (یعنی دقیقاً به همان اندازه در مجموع دو نفر حرکت انجام داده باشند) و هیچ کسی بازی را برنده نشده باشد بازی مساوی اعلام می‌شود. مساوی به شکل رایج شطرنج (پات) که قوانین مختلفی نظیر تعداد و نوع مهره‌ها و حرکات ممکن و... باید در آن چک شود، در این جا وجود ندارد.
- امکان unselect کردن مهره انتخابی برای حرکت وجود دارد و قانونی که در بین شطرنج‌بازان معروف

به "دست به مهره، حرکت" است وجود ندارد.

- قابلیت undo کردن حرکات وجود دارد (در این مورد توضیح بیشتری خواهیم داد)

قوانین کلی شطرنجی که باید پیاده سازی کنید:

- شروع حرکت با مهره سفید است.
- در صورت زده شدن شاه در این شطرنجی که قرار است پیاده سازی کنید، بازی به پایان رسیده و فردی که شاه را از دست داده بازی را می‌بازد. مهم ترین تفاوت این شطرنج با شطرنج واقعی هم همین مورد و نبودن وضعیت کیش است.
- سربازها در ابتدا می‌توانند یک یا دو خانه به جلو بروند و پس از آن تنها یک خانه به جلو می‌روند. نحوه زدن مهره های حریف توسط سرباز به صورت ضربدری است. در حین حرکت در جلوی سرباز نباید هیچ مهره ای باشد.
- فیل به صورت مورب حرکت می‌کند و مهره های دیگر را هم به همین صورت می‌زند. در حین حرکت در مسیر حرکت مورب فیل نباید هیچ مهره ای باشد.
- حرکت رخ به صورت افقی و عمودی است و مهره های دیگر را هم به همین صورت می‌زند. در حین حرکت در مسیر حرکت رخ نباید هیچ مهره ای باشد.
- حرکت وزیر مانند ترکیب حرکت رخ و فیل است و هم مورب و هم عمودی و افقی حرکت می‌کند. مهره‌های دیگر را هم به همین صورت می‌زند. در حین حرکت در مسیر حرکت وزیر نباید هیچ مهره ای باشد.
- حرکت اسب به صورت L شکل است. یعنی دو خانه به جلو و یک خانه به راست یا دو خانه به جلو و یک خانه به چپ یا دو خانه به راست و یک خانه بالا و... و مهره‌های دیگر را هم به همین صورت می‌زند. در حین حرکت می‌تواند از روی سایر مهره ها بپرد و قرار داشتن مهره‌های دیگر در سر راه اسب مشکلی برای حرکت ایجاد نمی‌کند.
- البته برنامه ای که باید بنویسید، صرفاً خود بازی شطرنج نیست. بلکه باید مکانیزم حساب کاربری و ثبت نام و لاگین و... را مطابق مواردی که در ادامه برای شما گفته می‌شود، پیاده سازی کنید.

در این برنامه، در اصل سه منو (menu) داریم. یکی منوی ابتدای بازی یا منوی ثبت‌نام است. یکی منوی

اصلی و یکی هم منوی خود بازی. در ادامه به توضیح دستوراتی که در هرکدام از این منوها وارد می‌شود می‌پردازیم. توجه کنید که در هنگام شروع برنامه، به طور پیش فرض، کاربر در منوی ثبت‌نام قرار دارد. در تمامی منوها دستور help وجود دارد که توضیح آن را در هر منو قرار داده‌ایم. ضمناً توجه کنید تمامی پیغام‌های انجام موفقیت آموز دستورات یا خطاها، در یک خط مجزا چاپ می‌شوند و بعد به خط بعدی می‌رود. همچنین توجه کنید که نماد [] در دستورات پایین در خود دستور نوشته نمی‌شوند و صرفاً برای نمایش این که در این قسمت عبارتی از سمت کاربر باید وارد شود نوشته شده‌اند. (یعنی مثلاً یک عبارت معتبر برای register [username] [password] به صورت register ali 1234 می‌تواند باشد)

**منوی ثبت‌نام:*

در ابتدای شروع برنامه کاربر در این منو قرار دارد. دستور:

```
register [username] [password]
```

همانطور که مشخص است ریال یک کاربر با نام کاربری و پسورد مشخص شده را ایجاد می‌کند. نام کاربری و پسورد باید فقط شامل حروف الفبای انگلیسی، اعداد و کاراکتر آندرلاین _ باشند. خطاهای مربوط به این دستور به این ترتیب چک می‌شوند و هر خطا که رخ داده بود، پیغام همان خطا چاپ شده و سایر خطاها بررسی نمی‌شود. اگر هیچ خطایی رخ نداد و عملیات موفقیت‌آمیز بود، پیغام زیر چاپ خواهد شد:

```
register successful
```

خطاها: اگر نام کاربری شامل کاراکترهایی به جز کاراکترهای ذکر شده بود، پیغام زیر چاپ خواهد شد:

```
username format is invalid
```

اگر پسورد شامل کاراکترهایی به جز کاراکترهای ذکر شده بود، پیغام زیر چاپ خواهد شد:

```
password format is invalid
```

اگر کاربری با username گفته شده از قبل وجود داشت:

```
a user exists with this username
```

دستور:

```
login [username] [password]
```

برای لاگین کردن به حساب کاربری مشخص با نام کاربری و پسورد داده شده استفاده می‌شود. نام کاربری و پسورد باید فقط شامل حروف الفبای انگلیسی، اعداد و کاراکتر آندرلاین _ باشند. خطاهای مربوط به این دستور به این ترتیب چک می‌شوند و هر خطا که رخ داده بود، پیغام همان خطا چاپ شده و سایر خطاها بررسی نمی‌شود. اگر هیچ خطایی رخ نداد و عملیات موفقیت‌آمیز بود، پیغام زیر چاپ خواهد شد:

```
login successful
```

خطاها: اگر نام کاربری شامل کاراکترهایی به جز کاراکترهای ذکر شده بود، پیغام زیر چاپ خواهد شد:

```
username format is invalid
```

اگر پسورد شامل کاراکترهایی به جز کاراکترهای ذکر شده بود، پیغام زیر چاپ خواهد شد:

```
password format is invalid
```

اگر کاربری با username گفته شده وجود نداشت:

```
no user exists with this username
```

اگر پسورد غلط بود:

```
incorrect password
```

دستور:

..

```
remove [username] [password]
```

برای حذف کردن به حساب کاربری مشخص با نام کاربری و پسورد داده شده استفاده می‌شود. نام کاربری و پسورد باید فقط شامل حروف الفبای انگلیسی، اعداد و کاراکتر آندرلاین _ باشند. خطاهای مربوط به این دستور به این ترتیب چک می‌شوند و هرخطا که رخ داده بود، پیغام همان خطا چاپ شده و سایر خطاها بررسی نمی‌شود. اگر هیچ خطایی رخ نداد و عملیات موفقیت‌آمیز بود، پیغام زیر چاپ خواهد شد که به جای username نام کاربری فرد حذف شده باید نشان داده شود:

```
removed [username] successfully
```

خطاها: اگر نام کاربری شامل کاراکترهایی به جز کاراکترهای ذکر شده بود، پیغام زیر چاپ خواهد شد:

```
username format is invalid
```

اگر پسورد شامل کاراکترهایی به جز کاراکترهای ذکرشده بود، پیغام زیر چاپ خواهد شد:

```
password format is invalid
```

اگر کاربری با username گفته شده وجود نداشت:

```
no user exists with this username
```

اگر پسورد غلط بود:

```
incorrect password
```

دستور:

```
list users
```

—

این دستور لیست تمامی کاربرانی که وجود دارند را به ترتیب الفبایی (Lexicograph) نمایش می‌دهد.

دستور:

```
help
```

این فرمان انواع دستوراتی که در این بخش قابل نمایش هستند را نمایش می‌دهد. از این دستور صرفاً برای اطمینان حاصل کردن از این که در منوی درست قرار دارید. خروجی این دستور در این منو به صورت زیر است:

```
register [username] [password]
login [username] [password]
remove [username] [password]
list_users
help
exit
```

دستور:

```
exit
```

بیانگر اتمام اجرای برنامه است و بعد از چاپ پیام زیر، اجرای برنامه پایان می‌پذیرد:

```
program ended
```

**منوی اصلی:*

همانطور که گفته شد، در صورت موفقیت آمیز بودن login وارد این منو خواهید شد. دستور:

```
new_game [username] [limit]
```


با این دستور یک بازی جدید شروع می‌شود. در این بخش باید username یک کاربر دیگر وارد شود و بدین ترتیب بازی با آن کاربر شروع خواهد شد. کاربری که اکنون لاگین کرده و دستور را زده است و در حین بازی به عنوان مهره سفید و بازیکنی که در این دستور نام او وارد شده و به عنوان مهره سیاه خواهد بود. توجه کنید که در این برنامه قرار نیست چیزی نظیر لاگین شدن همزمان دو کاربر و... را هندل کنید و همه کارها در یک برنامه و توسط یک کاربر انجام می‌شود. limit هم یک عدد است و بیانگر محدودیت تعداد حرکات (نوبت‌های بازی) است. اگر این عدد 0 باشد به معنی نبودن هیچ محدودیتی در بازی است. جزئیات بیشتر مربوط به limit در بخش مربوط به بازی توضیح داده شده‌اند. خطاها: اگر username از کارکترهایی که درمنوی لاگین توضیح دادیم تشکیل نشده بود، پیام زیر چاپ شود:

```
username format is invalid
```

اگر limit عددی کوچک‌تر از 0 بود:

```
number should be positive to have a limit or 0 for no limit
```

اگر کاربری که دستور را زده است، نام کاربری خودش را وارد کند:

```
you must choose another player to start a game
```

اگر کاربری با این نام کاربری وجود نداشت:

```
no user exists with this username
```

در صورت اجرای موفقیت آمیز دستور، عبارت زیر چاپ می‌شود که در آن first نام کاربری بازیکن سفید و second نام کاربری بازیکن سیاه و limit هم عدد محدودیت بازی است. (در حالتی که صفر وارد شده بود به معنی عدم محدودیت، هم عدد 0 نمایش داده خواهد شد و استثنایی وجود ندارد)

دستور:

scoreboard

پیش از توضیح خروجی این دستور باید درمورد امتیازدهی در بازی صحبت کنیم. هر برد معمولی (از طریق زدن شاه) 3 امتیاز و باخت به این شکل 0 امتیاز دارد. در صورتی که یکی از بازیکنان انصراف بدهد (با دستوری که در بخش مربوط به بازی توضیح می‌دهیم)، برنده بازی 2 امتیاز دریافت کرده و شخصی که انصراف داده 1 امتیاز منفی کسب می‌کند. در صورت تساوی بازی (که در اثر اتمام limit اتفاق می‌افتد) هر بازیکن 1 امتیاز می‌گیرد. بعد از اجرای این دستور باید کاربران با این فرمت نوشته بشوند:

```
[username] [score] [wins] [draws] [loses]
```

منظور از wins و draws و losses تعداد بردها، تساوی‌ها و باخت‌هاست. ترتیب مرتب سازی کاربران هم به ترتیب از بالاترین اولویت به کمترین به این صورت است: بیشترین امتیاز در صورت برابری امتیاز بیشترین تعداد برد در صورت برابری تعداد برد بیشتری تعداد تساوی در صورت برابری تساوی کمترین باخت و در صورت برابری تمامی موارد، براساس حروف الفبا (lexicographical) به طور صعودی (یعنی a زودتر از z می‌آید و...) می‌آید و...

دستور:

list_users

کاملاً مشابه همین دستور که در منوی قبلی توضیح داده شد، عمل می‌کند.

دستور:

help

مشابه همان چیزی است که در بخش قبل توضیح دادیم. خروجی آن عیناً به این شکل است:

```
new_game [username] [limit]
```

```
scoreboard  
list_users  
help  
logout
```

دستور:

```
logout
```

با این دستور کاربر از حساب کاربری خود خارج می‌شود. پس از اجرای این دستور پیام زیر چاپ شده و کاربر وارد منوی ثبت‌نام که پیش‌تر توضیح دادیم، می‌شود.

```
logout successful
```

منوی بازی.

در این بخش علاوه بر این که به دستورات اشاره می‌کنیم، به موارد خاصی که باید در منطق به درستی پیاده سازی شوند هم اشاره می‌شود. به عنوان اولین نکته به تصویر زیر که از صفحه شطرنج است توجه کنید:





نحوه قرار گیری مهره‌ها در صفحه بازی به همین شکل خواهد بود.

در این جا اولین دستور مطرح می‌شود:

```
select [x],[y]
```

به این دستور یک مهره را انتخاب می‌کنیم. توجه کنید که برای اشاره به یک خانه در اینجا به جای حروف A تا H از اعداد استفاده خواهیم کرد. یعنی مثلا در ابتدای بازی، وزیر سفید به صورت `select 4,1` و شاه سیاه با `select 8,5` انتخاب می‌شود. خطاها: اگر مختصات به طور کلی از محدوده خارج بود (یعنی هر مولفه در بازه 1 تا 8 قرار نمی‌گرفت):

```
wrong coordination
```

اگر مهره قرار گرفته در خانه مشخص شده مربوط به حریف بود:

```
you can only select one of your pieces
```

اگر هیچ مهره ای در خانه نبود:

no piece on this spot

اگر هم مهره خودی در آن موقعیت بود، پیام زیر چاپ می‌شود:

selected

توجه کنید که اگر یک مهره از قبل انتخاب شده باشد و بخواهیم مهره جدیدی را انتخاب کنیم، صرفاً با نوشتن دوباره دستور برای مهره جدید، مهره جدید انتخاب خواهد شد و دوباره همان خروجی selected را خواهیم داشت حتی با انتخاب دوباره مهره انتخاب شده هم با همین نتیجه رو به رو خواهیم بود.

دستور:

deselect

این دستور باعث می‌شود مهره‌ای که در وضعیت انتخاب شده بوده، از وضعیت انتخاب شده خارج شود. تنها خطای این دستور مربوط به وقتی است که هیچ مهره‌ای انتخاب نشده باشد و در آن صورت عبارت زیر چاپ می‌شود:

no piece is selected

در صورتی هم که به درستی مهره از حالت انتخاب خارج شود، پیام زیر چاپ خواهد شد:

deselected

دستور:

move [x],[y]

این حرکت همان طور که واضح است، باعث حرکت مهره به نقطه مشخص شده می‌شود.

خطاها: اگر قبل از این حرکتی در این نوبت انجام شده بود:

already moved

اگر مختصات به طور کلی از محدوده خارج بود (یعنی هر مولفه در بازه 1 تا 8 قرار نمی‌گرفت):

wrong coordination

اگر هیچ مهره‌ای انتخاب نشده بود:

do not have any selected piece

اگر خانه مورد نظر توسط مهره‌های خودی از قبل پر بود یا مانعی در مسیر برای حرکت به آن نقطه وجود داشت (مطابق قوانین شطرنج که در ابتدای داک توضیح داده شده‌اند)، پیام زیر چاپ خواهد شد:

cannot move to the spot

در صورت اجرای موفقیت آمیز، اگر حرکت به درستی انجام شد و در خانه مدنظر مهره حریف قرار داشت و زده شد، عبارت:

rival piece destroyed

و اگر صرفاً حرکت انجام شد و مهره ای زده نشد عبارت:

moved

چاپ خواهد شد.

دستور:

next_turn

نوبت را به حریف منتقل می‌کند و در صورت اجرای موفقیت آمیز پیام زیر چاپ می‌شود:

```
turn completed
```

توجه کنید که طبق قوانین شطرنج هر نفر باید در نوبت خود حتما یک حرکت انجام بدهد. بنابراین اگر هیچ حرکتی انجام نشده بود، پیام خطای:

```
you must move then proceed to next turn
```

نوشته شود.

دستور:

```
show_turn
```

این دستور برای مشخص شدن این است که نوبت کدام بازیکن است. خروجی آن به فرم زیر است:

```
it is player [username] turn with color [white/black]
```

دستور:

```
undo
```

یکی از ویژگی‌های منحصر به فرد شطرنجی که باید پیاده سازی کنید، قابلیت undo است. هر بازیکن در کل طول یک بازی می‌تواند دوبار از این قابلیت و در هر نوبت حداکثر یک بار استفاده کند. این قابلیت بدین صورت است که اگر بازیکنی حرکتی انجام داده باشد، در صورتی که بخواهد و طبق شرایط بالا تعداد دفعات استفاده‌اش تمام نشده باشد، حرکت خود را برگرداند. توجه کنید که چون تغییر نوبت با next_turn انجام می‌شود، همچنان بعد از حرکت بازیکن نوبت با بازیکن فعلی است تا زمانی که دستور next_turn زده شود. خطاها: اگر بیشتر به اندازه تعداد کل undoهای مجاز (یعنی دوبار) از این قابلیت استفاده شده باشد:

you cannot undo anymore

اگر بازیکن در این نوبت حرکتی انجام نداده باشد که بخواند undo کند:

you must move before undo

اگر در همین نوبت از undo استفاده کرده باشد:

you have used your undo for this turn

در صورت اجرای درست دستور نیز باید عبارت:

undo completed

چاپ شود. توجه کنید که بعد از اجرای دستور undo مهره ای که از قبل select شده همچنان در همین حالت باقی می ماند.

دستور:

undo_number

تعداد دفعات باقی مانده undo یک بازیکن را نشان می دهد. در ابتدای بازی این عدد دو است و با انجام حرکت undo کاهش می یابد تا در نهایت بعد از دوبار انجام این حرکت، به صفر می رسد. فرم پیام خروجی به صورت زیر خواهد بود که n تعداد undo های باقی مانده است.

you have [n] undo moves

برای دستورات بعدی باید توضیحی درمورد نحوه نمایش نام هر مهره در بازی بدهیم. نام مهره ها در بازی به این صورت است که ابتدا حرف اختصاری نوع مهره یا حرف رنگ و سپس رنگ مهره یا حرف که حرکت آورده

این سورت است که ابتدا، حرف اختصاری نوع مهره با حرف بررت و سپس رنگ مهره با حرف سوپت، آورده می‌شود. حروف اختصاری مهره‌ها بدین شرح است:

مهره	حرف اختصاری
سرباز	P
رخ	R
اسب	N
فیل	B
وزیر	Q
شاه	K

توضیحات تکمیلی
رنگ سفید با w نشان داده می‌شود
رنگ سیاه با b نشاده داده می‌شود

به عنوان نمونه شاه سفید با علامت KW شناخته می‌شود.

حال به توضیح دیگر دستورها می‌پردازیم:

`show_moves`

این دستور کل حرکاتی را که بازیکن فعلی انجام داده است، به ترتیب با این فرمت نمایش می‌دهد: (منظور از بازیکن فعلی، بازیکنی است که در هنگام زده شدن دستور نوبت اوست) اگر در طول فرآیند حرکت مهره‌ای زده نشده باشد:

`[Name of Piece that moved] [x],[y] to [x],[y]`

اگر مهره‌ای زده شده باشد:

[Name of Piece that moved] [x],[y] to [x],[y] destroyed [Name of piece destroy

به عنوان مثال اگر سرباز سفید با حرکت از ردیف 4 و ستون 1 به ردیف 5 و ستون 2 رفته باشد و در این فرآیند سرباز سیاه را از بین برده باشد، حرکت به صورت زیر نمایش داده خواهد شد:

Pw 4,1 to 5,2 destroyed Pb

دستور:

show_moves -all

این دستور مشابه دستور قبلی است با این تفاوت که حرکات تمامی بازیکنان را به ترتیب از ابتدای شروع بازی نمایش می‌دهد.

دستور:

show_killed

این دستور مهره‌های زده شده بازیکن فعلی را نشان می‌دهد. (یعنی مهره‌هایی از او که حریف آن‌ها را زده است) فرمت نمایشی به این صورت است:

[Name of Piece] killed in spot [x],[y]

ترتیب نمایش مهره‌ها به ترتیب زمان زده شدن است. (مهره‌ای که زودتر زده شده، زودتر نمایش داده می‌شود)

دستور:

`show_killed -all`

مشابه دستور بالاست ولی همه مهره‌های زده شده در بازی را به ترتیب زده شدن نمایش می‌دهد

دستور:

`show_board`

این دستور صفحه شطرنج را به همراه مهره‌های حاضر در آن نمایش می‌دهد. برای مثال در ابتدای بازی فرمت نمایش صفحه به صورت زیر است و به طور کلی باید به همین شکل نمایش بدهید:

```
Rb|Nb|Bb|Qb|Kb|Bb|Nb|Rb|
Pb|Pb|Pb|Pb|Pb|Pb|Pb|Pb|
 | | | | | | |
 | | | | | | |
 | | | | | | |
 | | | | | | |
Pw|Pw|Pw|Pw|Pw|Pw|Pw|Pw|
Rw|Nw|Bw|Qw|Kw|Bw|Nw|Rw|
```

یعنی زاویه دید، مشابه عکسی است که بالاتر از بازی شطرنج نمایش دادیم و سفید در پایین تصویر و ساع در بالای تصویر خواهد بود. توجه کنید که خانه‌های خالی با دو اسپیس نمایش داده می‌شوند. ساختار بالا را عیناً از سامانه کوئرا کپی پیست کنید.

دستور:

`help`

مشابه دستور help در سایر بخش‌هاست. در اینجا باید عیناً این خروجی نمایش داده شود:

```
select [x],[y]
deselect
```

```

move [x],[y]
next_turn
show_turn
undo
undo_number
show_moves [-all]
show_killed [-all]
show_board
help
forfeit

```

دستور:

forfeit

این دستور مربوط به انصراف از بازی می‌شود و بازیکنی که این دستور را وارد کند، بازنده بازی خواهد بود و نفر دیگر برنده بازی خواهد بود. همانطور که بالاتر گفتیم، با این اتفاق، حریف که برنده بازی شده، 2 امتیاز می‌گیرد و کسی که انصراف داده منفی 1 امتیاز دریافت می‌کند. (یعنی 1 امتیاز از او کم می‌شود) بعد از اجرای این دستور باید این دو پیام در دو خط پشت سرهم نمایش داده شوند:

```

you have forfeited
player [username] with color [black/white] won

```

و به جای username و black/white هم نام کاربری و رنگ بازیکن برنده قرار می‌گیرد. پس از آن هم کاربر به منوی اصلی بازگردانده می‌شود.

بدین ترتیب توصیف دستورات به پایان رسید. صرفاً باید مواردی را در مورد پایان بازی ذکر کنیم.

- در صورتی که شاه حریف زده شود، بازی به نفع شخصی که شاه حریف را زده است، تمام می‌شود. این موضوع که شاه حریف زده شده در دستور next_turn بررسی می‌شود و در صورت اتمام بازی با برد یکی از طرفین پیامی به صورت زیر چاپ می‌شود:

```

player [username] with color [black/white] won

```

- در مورد شرایط تساوی، همانطور که گفته شد، تنها باید وضعیت limit تعداد حرکات را بررسی کنید. با گذشت هرنوبت به نوعی یک واحد به limit مدنظر نزدیکتر می‌شویم یعنی به نوعی اگر limit عدد 20 باشد، هر یک از طرفین می‌تواند 10 نوبت بازی کند. در این مثال، اگر بعد از گذشت 20 نوبت در کل، هیچ یک از طرفین برنده بازی نشد، بازی با تساوی به اتمام می‌رسد. توجه کنید که اگر limit برابر 0 باشد، یعنی هیچ محدودیتی نیست و بازی ملا با تساوی به اتمام نخواهد رسید. در صورت تساوی عبارت زیر چاپ می‌شود و کاربر به منو اصلی باز گردانده می‌شود:

draw

- همانطور که پیش‌تر هم گفتیم، در اثر تساوی به هرکدام از بازیکنان 1 امتیاز داده می‌شود. با توجه به توضیحات بالا نیازی به چک کردن شرایط واقعی تساوی در شطرنج ندارید.
- در نهایت توجه کنید که اگر در هر کدام از منوها، دستوری زده شود که با دستورات آن منو تطابق نداشته باشد، بیاد پیام زیر در یک خط چاپ شود و به خط بعدی برویم:

invalid command

- درمورد خطاهای دستورات، توجه کنید که ترتیب چک شدن آن‌ها به ترتیبی است که در این داک نوشته شده‌اند و در صورت رخ دادن اولین خطا و چاپ پیام خطا، سایر خطاها بررسی نمی‌شوند.

ورودی نمونه

```
register ali abbasi
register abbas alipoor
login ali abbasi
new_game abbas 0
help
select 2,4
move 4,4
next_turn
select 7,7

move 6,7
next_turn
select 1.2
```

```
move 3,3
next_turn
select 7,3
move 6,3
next_turn
select 1,3
move 4,6
next_turn
select 8,2
move 6,1
next_turn
select 1,7
move 3,6
next_turn
select 8,7
move 6,6
next_turn
select 2,5
move 3,5
next_turn
select 7,2
move 5,2
next_turn
select 2,2
move 3,2
next_turn
select 5,2
move 4,2
next_turn
select 3,3
move 4,1
next_turn
select 6,6
move 4,5
next_turn
select 1,4
move 2,5

next_turn
select 4,5
move 3,3
```

```
next_turn
select 4,1
move 3,3
next_turn
select 4,2
move 3,3
next_turn
select 2,5
move 3,4
next_turn
select 7,5
move 5,5
next_turn
select 4,4
move 5,5
next_turn
select 6,1
move 5,3
next_turn
select 3,4
move 3,3
next_turn
select 5,3
move 4,5
next_turn
select 3,3
move 4,4
next_turn
select 8,4
move 5,1
next_turn
select 2,3
move 3,3
next_turn
select 5,1
move 3,3
next_turn

select 4,4
move 3,3
next_turn
```

```
select 4,5
move 3,3
next_turn
select 2,1
move 4,1
next_turn
select 8,1
move 8,2
next_turn
select 1,6
move 4,3
next_turn
select 7,4
move 5,4
next_turn
select 1,1
move 1,3
next_turn
select 3,3
move 4,5
next_turn
select 4,3
move 3,4
next_turn
select 8,6
move 4,2
next_turn
select 1,5
move 2,5
next_turn
select 8,3
move 7,2
next_turn
select 3,4
move 4,5
next_turn
select 5,4

move 4,5
next_turn
select 3,6
```



```
move 4,4
next_turn
select 7,2
move 6,1
next_turn
select 2,5
move 1,4
next_turn
select 7,8
move 5,8
next_turn
select 4,4
move 6,3
next_turn
select 8,2
move 5,2
next_turn
select 6,3
move 4,2
next_turn
select 5,2
move 4,2
next_turn
select 1,3
move 6,3
next_turn
select 4,2
move 6,2
next_turn
select 6,3
move 6,2
next_turn
select 7,1
move 6,2
next_turn
select 1,4
move 1,3

next_turn
select 6,1
move 3,4
```

```
next_turn
select 2,6
move 3,6
next_turn
select 3,4
move 2,5
next_turn
select 3,6
move 4,5
next_turn
select 7,6
move 6,6
next_turn
select 5,5
move 6,6
next_turn
select 8,8
move 8,6
next_turn
select 2,7
move 4,7
next_turn
select 5,8
move 4,7
next_turn
select 2,8
move 3,8
next_turn
select 4,7
move 3,8
next_turn
select 1,8
move 3,8
next_turn
select 8,6
move 6,6
next_turn

select 3,8
move 6,8
next_turn
```

```
select 2,5
move 3,4
next_turn
select 6,8
move 6,7
next_turn
select 3,4
move 4,5
next_turn
select 4,6
move 7,3
next_turn
select 4,5
move 6,3
next_turn
select 7,3
move 6,2
next_turn
select 6,3
move 4,1
next_turn
select 3,5
move 4,5
next_turn
select 6,6
move 6,5
next_turn
select 6,7
move 6,8
next_turn
select 6,5
move 4,5
next_turn
select 1,3
move 1,4
next_turn
select 4,1

move 3,2
next_turn
select 6,8
```


[illegible]

rival piece destroyed
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
rival piece destroyed

turn completed
selected
rival piece destroyed
turn completed

turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
rival piece destroyed
turn completed

selected
moved
turn completed
selected

moved
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected

moved
turn completed
selected
moved

turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
rival piece destroyed
turn completed
selected
rival piece destroyed
turn completed
selected
moved

turn completed
selected
moved
turn completed

selected
rival piece destroyed
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed
selected
moved
turn completed
selected
rival piece destroyed
turn completed

selected
moved
turn completed
selected

rival piece destroyed
it is player abbas turn with color black
Nb killed in spot 3,3
Nw killed in spot 3,3
Pb killed in spot 5,5
Pb killed in spot 3,3
Pw killed in spot 3,3
Qb killed in spot 3,3
Qw killed in spot 3,3
Nb killed in spot 4,5
Bw killed in spot 4,5
Pb killed in spot 6,3
Bb killed in spot 4,2
Nw killed in spot 4,2
Rb killed in spot 6,2
Rw killed in spot 6,2
Pb killed in spot 4,5
Pb killed in spot 6,6
Pw killed in spot 4,7
Pw killed in spot 3,8
Pb killed in spot 3,8
Pw killed in spot 6,6
Pb killed in spot 6,7
Pw killed in spot 4,5
Pb killed in spot 6,2
Pw killed in spot 4,1
Pw killed in spot 4,5
Pw killed in spot 3,2
Kw killed in spot 1,4
Nb killed in spot 3,3
Pb killed in spot 5,5
Pb killed in spot 3,3
Qb killed in spot 3,3
Nb killed in spot 4,5
Pb killed in spot 6,3
Bb killed in spot 4,2

Rb killed in spot 6,2
Pb killed in spot 4,5
Pb killed in spot 6,6
Pb killed in spot 3,8

```

Pb killed in spot 6,7
Pb killed in spot 6,2
  |  |  |  |Kb|  |  |  |
  |  |  |  |  |  |  |  |
  |Bw|  |  |  |  |  |  |
  |  |  |  |  |  |  |Rw|
  |  |  |  |Rb|  |  |  |
  |  |  |  |  |  |  |  |
  |  |  |  |  |  |  |  |
  |  |  |  |  |  |  |  |
  |  |  |Bb|  |  |  |  |
turn completed
player abbas with color black won
logout successful
program ended

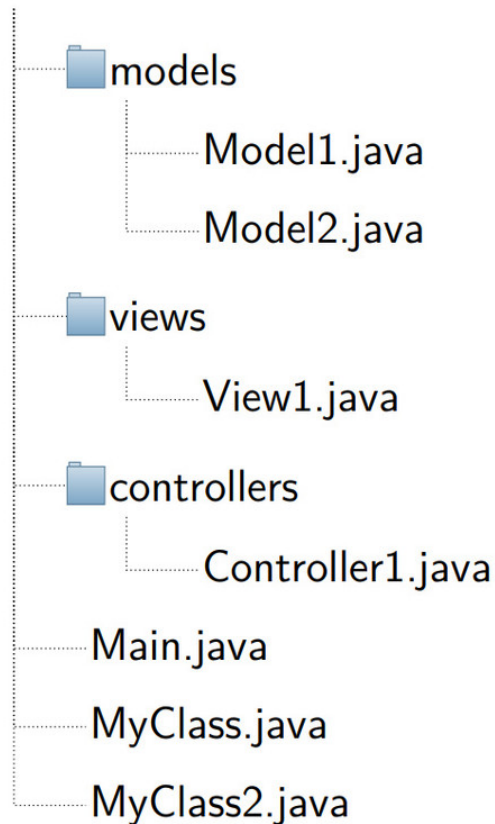
```

این سوال توسط سامانه کوئرا داوری و نمره‌دهی خواهد شد. توجه داشته باشید به هیچ‌وجه تصحیح این سوال به صورت دستی انجام نخواهد شد. لطفاً به موارد زیر توجه کنید:

- با توجه به اینکه SDK کوئرا نسخه 1.8 می‌باشد لطفاً از نسخه استفاده کنید.
- در کل پروژه فقط یک شی Scanner داشته باشید.
- جهت آپلود کدهای خود به یکی از دو روش زیر اقدام کنید:
 - برای آپلود همه‌ی کلاس‌ها رو کپی کنید زیر یک کلاس (توی یک فایل java) فقط توجه کنید که توی یک فایل فقط می‌تونید یک کلاس public داشته باشید. بعد از کپی کردن کلاس‌ها توی یک فایل پیشوند پابلیک کلاس‌هایی که کپی کردید رو بردارید تا فقط یک کلاس پابلیک باقی بماند. (حتماً کلاسی که حاوی تابع `public static void main(string[] args)` بصورت public باشد)
 - یک نمونه مثال ساده برای اینکه چقدر کلاس توی یک فایل داشته باشید.
 - شما باید تمامی فایل‌های کلاس‌هایی که با فرمت java نوشته‌اید را در قالب فشرده با فرمت zip درآورده و در سامانه کنید. توجه کنید که حتماً فایل حاوی کلاسی که تابع `public static void main(string[] args)` در آن قرار دارد و اجرای برنامه از آنجا شروع می‌شود، باید بدون قرار گرفتن در هیچ پوشه‌ای درون فایل zip قرار بگیرد و نام آن هم Main.java باشد. امکان پوشه بندی و ایجاد package‌های مختلف برای سایر فایل‌ها امکان پذیر است. به

عنوان نمونه می‌توانید به ساختار قابل قبولی و درستی که در زیر رسم شده است، توجه کنید.
توجه کنید که پوشه src که فایل‌ها در آن قرار دارد، نباید در فایل zip باشد.

answer.zip



- ورودی و خروجی شما باید عیناً شبیه به نمونه‌های ورودی و خروجی باشد؛ لذا عبارتهایی همچون Enter your commend را قبل از گرفتن ورودی نباید چاپ کنید.