مقدمه

شما باید بازی شطرنج را پیاده سازی کنید. برای اطلاعات بیشتر در زمینه‌ی این بازی می‌توانید به آدرس http://en.wikipedia.org/wiki/Chess مراجعه فرمائید. کلیه‌ی کلاسها و پکیجهای این برنامه باید داخل پکیج ir.ac.mahshahriau.chess باشند.

برنامه‌ای که می‌نویسید باید قابلیت انجام دادن بازی بصورت انسان با انسان و هوش مصنوعی با هوش مصنوعی را داشته باشد.

واسط کاربری

کلیه‌ی کلاس‌هایی که مربوط به واسط کاربری می‌باشند باید داخل پکیج ir.ac.mahshahriau.chess.ui باشند. (و همچنین کلاس‌هایی که مربوط به واسط کاربری نیستند نباید در این پکیج باشند!) دقت کنید که واسط کاربری تنها برای حالت بازی انسان با انسان است و برای حالت بازی هوش مصنوعی با هوش مصنوعی نیاز به واسط کاربری نیست.

شما باید یک رابط کاربر در نظر بگیرید که کابر بتواند با آن کار کند. برای این کار پیشنهاد می‌شود از JFrame استفاده کنید.(در صورتیکه خواستید، مانعی ندارد که از Applet نیز استفاده نمائید.) برای حرکت دادن یک مهره، کاربر باید روی خانه‌ی آن مهره کلیک کند و سپس روی مقصد کلیک کند. در صورتیکه مجددا روی همان خانه کلیک کرد باید آن مهره از حالت انتخاب بیرون بیاید. همچنین در صورتیکه خارج از صفحه‌ی شطرنج کلیک کرد، باید آن مهره از حالت انتخاب بیرون بیاید.

دقت کنید که وضعیت فعلی بازی (اعم از نوبت و اینکه بازیکن در حالت کیش هست یا نه) باید به کاربر نشان داده شود.

این قسمت در تحویل حضوری بررسی می‌شود.

هوش مصنوعی

هوش مصنوعی که به بازی شما داده می‌شود (نحوه‌ی داده شدن هوش مصنوعی به بازی بعدا توضیح داده می‌شود)، شی-ای است که interface زیر را پیاده‌سازی (implement) می‌کند.

```
public interface IPlayer {
      public String move(Chess chess);
}
```

در هنگامی که نوبت به هوش مصنوعی برسد، شما با فراخوانی متد move هوش مصنوعی، حرکت مورد نظرش را دریافت می‌کنید. خروجی این متد یک رشته است که مشخص می‌کند مهره‌ی درون چه خانه‌ای، به چه خانه‌ای برود. برای مثال اگر خروجی این متد، رشته‌ی "A2->A4" باشد، باید مهره‌ی درون خانه‌ی A2 به خانه‌ی A4 برود. ورودی این تابع یک شی از جنس Chess است که همان بازی در حال اجراست که بعدا توضیح داده خواهد شد. (دلیل اینکه این گونه ورودی باید به متد move داد این است که متد move، در هنگام اجرای یک حرکت، باید موقعیت تمامی مهره‌ها را بداند.)

مهره‌ها

برای مهره‌ها باید یک پکیج تحت نام ir.ac.mahshahriau.chess.pieces در نظر بگیرید که باید در آن یک کلاس abstract تحت نام Piece وجود داشته باشد که ۲ متد abstract دارد. این متدها عبارتند از: (پارامترهای این متدها بر عهده‌ی خودتان است.)

- draw: این تابع وظیفه‌ی رسم مهره را بر عهده دارد.
- canMoveTo: این تابع آدرس یک خانه را می‌گیرد و چک می‌کند که آیا این مهره می‌تواند به آن خانه برود یا نه. (نیاز به پیاده سازی حرکت شاه-قلعه[۳] و ارتقاء سرباز[۴] نیست)

سپس به ازای هر نوع مهره از شطرنج، باید یک کلاس بگیرید که بچه این کلاس Piece باشد. برای هر نوع مهره، باید متد toString را override کنید که خروجی آن بصورت "XY" می‌باشد. Y نشان‌دهنده‌ی رنگ مهره می‌باشد (در صورتیکه سفید بود، W و در صورتیکه سیاه بود، B خواهد بود) و X، نوع مهره می‌باشد که مطابق با جدول زیر مشخص می‌گردد: (برای مثال سرباز سیاه بصورت "PB" و اسب سفید بصورت "NW" مشخص می‌شوند.)

اسب	فیل	وزیر	شاه	سرباز	قلعه (رخ)
N	B	Q	K	P	R

خطا[۵]ها

کلیه‌ی کلاسهای مربوط به خطاها باید داخل پکیج ir.ac.mahshahriau.chess.exceptions باشند.

اولین کلاسی که باید در این پکیج در نظر بگیرید، کلاس ChessException است که بچه‌ی Exception می‌باشد.

---

[۳] Castling

[۴] Promotion: وقتی که سرباز یک بازیکن به خانه‌ی آخر رسید، حق دارد بجای آن یک مهره‌ی مطلوب بیاورد.

[۵] Exception

برنامه‌ی شما باید خطاهای لازم را به کاربر بدهد. (دقت کنید که این خطاها در حالتی که ۲ انسان با هم بازی می کنند، باید به صورت گرافیکی به کاربر نشان داده شوند[۶] و در حالتی که ۲ هوش مصنوعی با هم بازی می‌کنند نباید به صورت گرافیکی نشان داده شوند) خطاهایی که پیاده سازی آنها ضروری است عبارتند از:

- حرکت اشتباه مهره‌ها
- کیش بودن(موقعی که در حالت کیش بودن، حرکتی انجام دهد که هنوز کیش بماند یا در وضعیت عادی، حرکتی کند که به حالت کیش برود.)
- پرش از روی مهره‌ها (برای مهره‌های غیر اسب!)
- عدم رعایت نوبت
- خالی بودن خانه
- زدن مهره‌ی خودی(!)

دقت کنید که برای هر خطا باید یک کلاس جداگانه بگیرید که بچه‌ی ChessException باشد.

کلاس Chess

در واقع وظیفه‌ی شما، پیاده‌سازی کلاسی همانند کلاس زیر است:

```
Public class Chess{
      public Chess(); // human vs. human game
      public Chess(IPlayer ai1, IPlayer ai2); // AI vs. AI game
      public void nextMove() throws ChessException;
      public char getTurn();
      public String getPieceIn(String place);
}
```

Constructor اول، برای حالتی است که انسان با انسان بازی می‌کند، با صدا زده شدن این constructor، واسط کاربری نمایش داده می‌شود و آماده‌ی گرفتن حرکت کاربر می‌شود.

Constructor دوم، برای حالتی است که هوش مصنوعی با هوش مصنوعی بازی می‌کند و بوسیله‌ی دو پارامتر ai1 و ai2 آن دو هوش مصنوعی را به بازی می‌دهیم.

---

[۶] برای راحتی می‌توانید از توابع ایستای کلاس JOptionPane استفاده کنید.

متدهای بعدی برای تست خودکار برنامه در حالت بازی هوش مصنوعی با هوش مصنوعی می‌باشند، و برای حالتی که انسان با انسان در حال بازی کردن است، کابردی ندارند.

متد nextMove، متد move را برای بازیکنی که اکنون نوبت اوست، فراخوانی می‌کند. در صورتیکه هر کدام از حالت‌های استثنا رخ دهد، مستقل از نوع خطا، باید یک ChessException حاوی پیغام "Exception for X" بیندازد که در آن X مشخص کننده‌ی رنگ مهره‌ی بازیکنی است که خطا برای آن رخ داده است. برای مثال اگر این خطا برای بازیکن با رنگ سفید رخ دهد، باید حاوی پیغام "Exception for W" باشد و اگر برای بازیکن با رنگ سیاه رخ دهد، باید حاوی پیغام "Exception for B" باشد.

متد getTurn، مشخص می‌کند که نوبت کدام بازیکن است. در صورتیکه نوبت بازیکن سیاه باشد، کاراکتر 'B' و در صورتیکه نوبت بازیکن سفید باشد، کاراکتر 'W' را برمی‌گرداند.

متد getPieceIn، این متد مشخص می‌کند که در خانه‌ی place چه مهره‌ای قرار دارد. در صورتیکه مهره‌ای در این خانه قرار داشت، مقدار خروجی این متد، همان مقدار خروجی متد toString برای آن مهره خواهد بود و در صورتیکه مهره‌ای در این خانه قرار نداشت، مقدار خروجی برای این متد، "E" خواهد بود. دقت کنید که رشته‌ی place، رشته‌ای است که کاراکتر اول آن ستون را مشخص می‌کند و کاراکتر دوم آن، سطر را مشخص می‌کند. برای مثال اگر رشته‌ی palce، بصورت "F5" بود، یعنی خانه‌ی با ردیف 5 و ستون F. دقت کنید در این متد در صورتیکه رشته‌ی مربوط به خانه‌ی ورودی، مجاز نبود، می‌توانید از IllegalArgumentException استفاده کنید.

جمع بندی

در نهایت داشتن کلاسهای زیر برای این تمرین اجباری است:

Package	Class / Interface
ir.ac.mahshahriau.chess	Chess
ir.ac.mahshahriau.chess.ui	IPlayer … …
ir.ac.mahshahriau.chess.pieces	Piece
ir.ac.mahshahriau.chess.exceptions	… 6 more ChessException
	… 6 more