



**Amirkabir University of Technology**  
**(Tehran Polytechnic)**

دانشکده ریاضی و علوم کامپیوتر

برنامه سازی پیشرفته

استاد:

مهدی قطعی – امین رحمانی

پروژه پایانی (شطرنج)

نوشته شده توسط:

امیر عباس الوند

شماره دانشجویی:

40013034

خرداد-1401

## فهرست مطالب

|                          |                                    |   |
|--------------------------|------------------------------------|---|
| 3.....                   | مقدمه:                             | 4 |
| 4.....                   | کلاس ها:                           | 4 |
| <b>iNTERFACES:</b> ..... |                                    | 4 |
| •                        | <b>Board:</b> .....                | 4 |
| •                        | <b>StartMenu:</b> .....            | 5 |
| •                        | <b>Piece:</b> .....                | 5 |
| •                        | <b>Tile:</b> .....                 | 6 |
| •                        | <b>Main:</b> .....                 | 6 |
| •                        | کلاس های مهره ها: .....            | 6 |
| •                        | <b>AIPlayer&amp;MiniMax:</b> ..... | 6 |

## مقدمه:

در این پروژه تلاش شده که به کمک زبان جاوا و تفکر شیع گرای مدل ساده ای از بازی شطرنج با قابلیت بازی کردن انفرادی و چند نفره پیاده سازی کرد.

در این پروژه از پرداختن به برخی جزئیات بازی مانند:

- محدودیت کیش شدن
- تبدیل شدن مهره سرباز در انتها به مهره دیگر
- جابجایی رخ و شاه(شاه قلعه یا قلعه بستن)

صرف نظر شده است.

## کلاس ها:

### iNTERFACES:

#### ChessMan:

به این منظور استفاده می شود تا بتوان هر مهره در شطرنج را مشخص کرد تا در مراحل بعد بتوان به کمک هر code که به یک مهره اختصاص دارد، صفحه را رسم کرد و حرکات را انجام داد همچنین امتیاز و ارزش هر مهره در اینجا مشخص شده است.

#### Player:

برای انجام حرکات استفاده میشود که سازنده آن موقعیت مهره و رنگ مهره است که با کمک آنها میتوان نوبت دهی را انجام داد و حرکات را در صفحه نمایش داد.

- **Board:**

این کلاس که از کلاس JFrame ارث بری میکند و Player را پیاده سازی میکند مسئول ساخت صفحه و انجام حرکات و تخصیص نوبت دهی به بازیکنان و همچنین اضافه کردن مهره های مرده به پل graveyard است.

در سازنده این کلاس عمل ساخت دکمه های ذخیره بازی و بازی جدید انجام میشود و ابعاد صفحه شطرنج مشخص می شود.

متد های این تابع:

```
public void clickListener()
```

در این متد فعالیت های کلیک کردن مشخص میشود و فعالیت هایی همچون نشان گذاری مهره انتخاب شده و نشان دادن خانه های قابل دسترس ان مهره در این متد انجام می شود.

```
public void buildBoard(int[][] positions)
```

این متد مسئول ساخت صفحه شطرنج بر اساس موقعیت داده شده به عنوان پارامتر است که کد موجود در آن موقعیت را شناسایی و بر اساس نقشه به دست آمده صفحه شطرنج را طراحی میکند، هر بار که مهره ای را جابجا میکنیم به دلیل تغییر سافتن موقعیت ها این متد دوباره بر اساس نقشه جدید صفحه شطرنج را بازسازی میکند و در واقع حرکت مهره ها اینگونه انجام می شود.

```
public void move(int[][] p1, int[][] p2)
```

تغییرات موقعیت در این متد انجام می شود و فیلد position در این متد بازسازی و به روز رسانی میشود، همچنین عمل کشتن مهره ها و اضافه کردن آنها به پنل graveyard در این متد انجام می شود که شامل کشته شدن شاه هم هست، که پس از کشته شدن شاه بازی به اتمام می رسد.

```
public int[][] nextMove(int[][] positions, boolean white)
```

این متد حرکات تغییر یافته را نوبت دهی کرده و به متد buildBoard ارسال میکند تا صفحه بازی دوباره از نو بر اساس نقشه جدید طراحی شود و سپس نوبت را تغییر می دهد.

```
public static void saveGame()
```

این متد به کمک فایل نقشه را ذخیره میکند تا پس از فراخوانی صفحه شطرنج بر اساس نقشه ذخیره شده صفحه شطرنج را طراحی کند همچنین در فایل مربوط به این کلاس در کنار نقشه بازی،نوبت و اطلاعات مربوط به حالت بازی(تک نفره یا چند نفره) نیز ذخیره می شود تا پس از فراخوانی مشکلی در نوبت دهی و حالت بازی ایجاد نشود.

```
public void close()
```

در این تابع که با کلیک بر روی new game فراخوانی می شود مسئول بسته شدن صفحه شطرنج به منظور نمایش صفحه Start می باشد.

- **StartMenu:**

این کلاس که از کلاس JFrame ارث بری می کند و مسئول نمایش صفحه اول بازی که شامل New Game,Continue,Single Player,Multi Player است.

- **Piece:**

این کلاس یک کلاس abstract است و کلاس های مربوط به مهره ها از این کلاس ارث بری میکنند زیرا هر مهره باید ارزش ها و متد های موجود در این کلاس را داشته باشد مانند:موقعیت، حرکات قانونی و ... .

به طور مثال همه مهره های موجود در بازی شطرنج موقعیت مکانی دارند و همچنین هر مهره دارای یک حرکت قانونی است مثلا نمیتوان مهره ای را از خانه هشتم فراتر برد و این حالت برای همه مهره ها صادق است.

- **Tile:**

این کلاس مشخصات هر خانه را با خود حمل میکند، مشخصاتی مانند موقعیت آن خانه و خالی بودن یا نبودن آن خانه.

- **Main:**

این کلاس فقط مسئول ساخت object از کلاس Board و StartMenu است.

- **کلاس های مهره ها:**

کلاس هایی مانند: Bishop, King, Queen و ... که مشخصات هر مهره مانند تصویر آنها و حرکات آنها را مشخص می کند.

این کلاس از کلاس Piece ارث بری می کند زیرا هر مهره باید مشخصات موجود در این کلاس را دارا باشد در غیر این صورت نمیتواند حرکات خود را انجام دهند. برای پیاده سازی حرکات قانونی هر مهره با کمک موقعیت آن مهره در صفحه بر اساس قوانین شطرنج و پیدا کردن خانه های خالی یا خانه هایی با مهره های هم رنگ یا غیر هم رنگ حرکات قانونی و ممکن آن مهره را مشخص می کنیم و ذخیره میکنیم. که در این بین مهره Pawn (پیاده) سخت ترین مهره بود، به این دلیل مه موقعیت مکانی و جهت حرکت این مهره در رنگ های مشکی و سفید متفاوت بود و به همین دلیل سخت ترین مهره برای پیاده سازی این مهره بود، باید ابتدا رنگ موقعیت مورد نظر را مشخص میکردیم و بر اساس آن حرکات مستقیم و اوریب را مشخص می کردیم و در طرف مقابل ساده ترین مهره Queen (وزیر) زیرا که اجتماع حرکات ممکن Rook (رخ) و Bishop (فیل) است و با پیاده سازی حرکات این دو مهره در واقع فقط نیاز به کپی پیست کردن موقعیت های ممکن برای حرکت این دو مهره است.

- **AIPlayer&MiniMax:**

این کلاس نقشه را از Board دریافت میکند و به کمک کلاس MiniMax نقشه را تحلیل میکند و حرکت بعدی را انجام میدهد و نقشه جدید را باز می گرداند و بر اساس نقشه های دریافتی و بازگردانی شده با هوش مصنوعی بازی یک نفره انجام می شود.