**پروژه میان ترم ساختمان داده**

**نیمسال تحصیلی 1403- 1404**

دکتر امین طوسی

---

### هدف پروژه:

هدف ما از این پروژه پیاده سازی بازی **"2048"** با استفاده از ساختمان دادهٔ Linked List است. این بازی روی یک صفحه 4x4 انجام می‌شود. در هر حرکت، بازیکن می‌تواند تمام کاشی‌ها را به یکی از چهار جهت (بالا، پایین، چپ، یا راست) حرکت دهد. وقتی دو کاشی با عدد مشابه در مسیر حرکت به هم برخورد کنند، ادغام شده و جمع می‌شوند و یک کاشی جدید با عدد بزرگ‌تر تشکیل می‌دهند. هدف بازی این است که با ادغام مکرر کاشی‌ها، به عدد **2048** برسید.
جهت آشنایی بیشتر با این بازی میتوانید به این لینک مراجعه کنید.

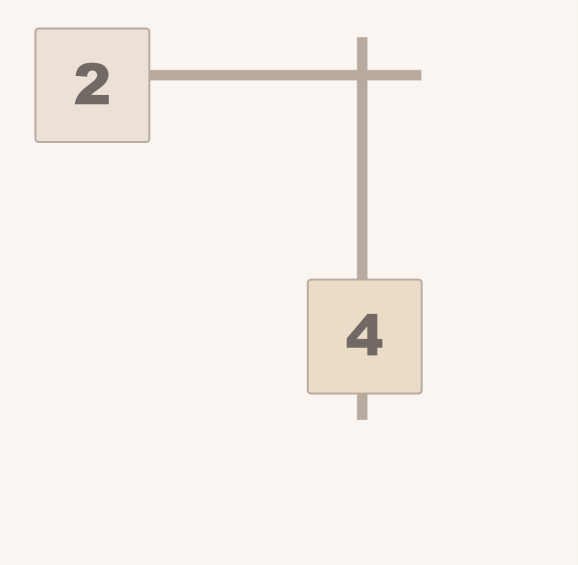
### کلاس اصلی بازی 2048:

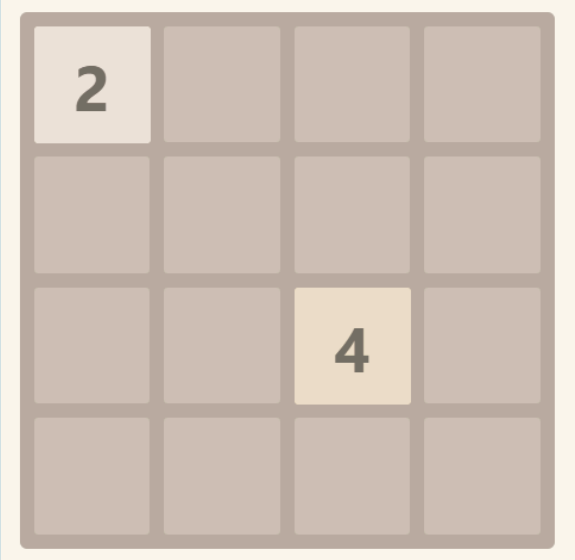
```
public class Game2048 {
  private static final int BOARD_SIZE = 4;

  public Game2048(){
    // implementation
  }
}
```

این کلاس، صفحه‌ی 4x4 بازی 2048 را با استفاده از **یک ساختار Linked List** پیاده‌سازی می‌کند که تنها **خانه‌های حاوی اعداد** را ذخیره می‌نماید.

**عملکرد کلاس:**

**ایجاد صفحه بازی**: در ابتدای بازی، دو عدد 2 یا 4 به‌صورت تصادفی در خانه‌های خالی قرار داده می‌شوند. این مقادیر اولیه، شروع بازی را ممکن می‌سازند.




**انجام حرکات بازی**

بازی 2048 از چهار حرکت اصلی (بالا، پایین، چپ و راست) تشکیل شده است، هدف این است که تنها **گره‌های دارای مقدار** (یعنی خانه‌های حاوی اعداد) در لیست باقی بمانند و خانه‌های خالی به عنوان گره وجود نداشته باشند.

```
public void move_up(){
  // implementation
}
public void move_dowm(){
  // implementation
}
public void move_right(){
  // implementation
}
public void move_left(){
  // implementation
}
```

هر حرکت به‌طور خلاصه به صورت زیر عمل می‌کند:

- **حرکت به بالا (moveUp)**: این تابع گره‌های حاوی عدد را در هر ستون به سمت بالاترین موقعیت خالی در همان ستون حرکت می‌دهد. در صورت وجود دو عدد یکسان به‌صورت متوالی، این گره‌ها با هم ترکیب شده و عدد حاصل در گره بالایی ذخیره می‌شود، و گره پایینی حذف می‌گردد. سپس ساختار Linked List به‌روز می‌شود تا گره‌های خالی را حذف کرده و ترتیب جدید گره‌ها را حفظ کند.
- **حرکت به پایین (moveDown)**: این حرکت مشابه حرکت به بالا عمل می‌کند، با این تفاوت که گره‌های هر ستون به سمت پایین‌ترین موقعیت خالی حرکت می‌کنند. ترکیب مقادیر و به‌روزرسانی Linked List به همان روش انجام می‌شود.
- **حرکت به چپ (moveLeft)**: این تابع گره‌های حاوی مقدار را در هر ردیف به سمت چپ‌ترین موقعیت خالی منتقل می‌کند. اگر دو گره متوالی مقدار یکسانی داشته باشند، آن‌ها با هم ترکیب شده و تنها یک گره با مقدار جدید به عنوان حاصل در جایگاه چپ‌تر باقی می‌ماند. گره‌های خالی حذف شده و ارجاعات لیست پیوندی به‌روز می‌شوند.
- **حرکت به راست (moveRight)**: در این حرکت، گره‌های هر ردیف به سمت راست‌ترین موقعیت خالی منتقل می‌شوند و در صورت وجود مقادیر مشابه، گره‌ها با هم ترکیب شده و گره‌های خالی حذف می‌شوند.

**در هر حرکت، پس از جابجایی و ترکیب گره‌های حاوی مقدار، ارجاعات در Linked List به‌روزرسانی می‌شوند تا تنها گره‌های حاوی اعداد نگهداری شوند.**

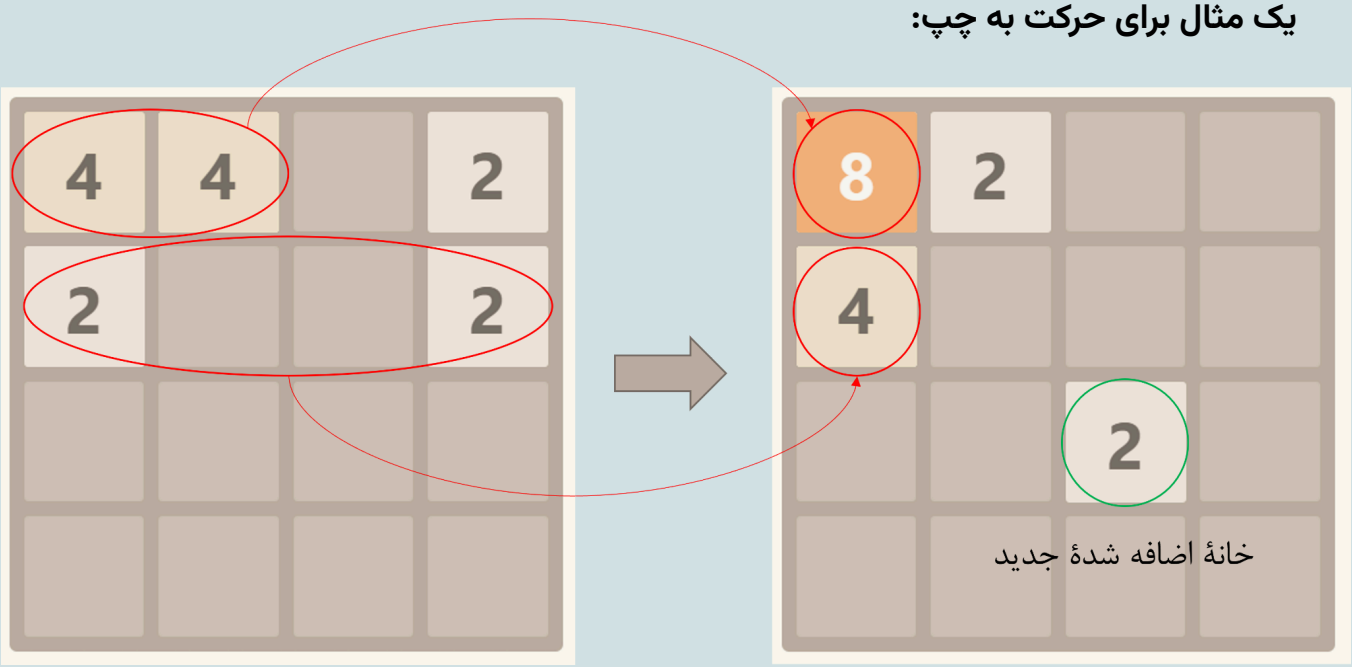
**تابع AddNewTail:**

```
public void AddNewTail(){
  // implementaation
}
```

این تابع وظیفه دارد پس از هر حرکت موفق در بازی، یک خانه جدید با مقدار 2 یا 4 را به‌طور تصادفی به صفحه اضافه کند. عملکرد این تابع باید شامل موارد زیر باشد:

- انتخاب خانهٔ تصادفی برای افزودن عدد جدید
- انتخاب مقدار عددی به این صورت که با احتمال 70 درصد مقدار 2، و 30 درصد مقدار 4 اضافه شود.
- افزودن خانه جدید به Linked List

**یک مثال برای حرکت به چپ:**



**تابع امتیازدهی**

در هر حرکت، با ترکیب کاشی‌های مشابه، به امتیاز بازیکن افزوده می‌شود. این تابع به گونه‌ای عمل می‌کند که هر زمان دو کاشی با مقدار یکسان با هم ترکیب می‌شوند، مقدار حاصل به امتیاز کل اضافه می‌شود.

- به عنوان مثال، اگر دو کاشی با مقدار 16 با هم ترکیب شوند و کاشی جدیدی با مقدار 32 ایجاد شود، 32 امتیاز به امتیاز کل اضافه می‌شود
- در پایان هر حرکت، امتیاز کل به روز می‌شود تا بازیکن بتواند پیشرفت خود را ببیند.

**نمایش وضعیت بازی (Display):**

```
public void Display(){
  // implementation

}
```

تابع Display وظیفه دارد وضعیت فعلی بازی را در قالب یک ماتریس 4*4 که هر عنصر آن شامل مقدار هر خانه است را نمایش دهد. علاوه بر نمایش کاشی‌ها، امتیاز فعلی بازیکن نیز در کنار وضعیت صفحه نشان داده می‌شود.

**اعلام نتیجه بازی:**

```
public boolean hasWon(){
  // implementation
  return hasWon;
}
```

کاربر زمانی برنده می‌شود که به امتیاز 2048 برسد. در این صورت تابع hasWon مقدار true برمی‌گرداند.

باخت زمانی اتفاق می‌افتد که هیچ حرکت ممکنی برای ترکیب یا جابجایی اعداد وجود نداشته باشد. در این صورت تابع hasWon مقدار false برمیگرداند.

**قابلیت Undo و Redo:**

```
public void Undo(){
  // implementation
}
public void Redo(){
    // implementation
}
```

برای افزودن قابلیت‌های Undo و Redo به بازی 2048، وضعیت‌های قبلی صفحه بازی ذخیره می‌شوند تا کاربر بتواند به عقب برگردد یا تغییرات انجام شده را بازگرداند. با این ویژگی‌ها، کاربر می‌تواند در صورت اشتباه یا تغییر نظر در مورد حرکت‌ها، به حالت قبلی بازی بازگردد یا تصمیمات قبلی را دوباره اعمال کند.

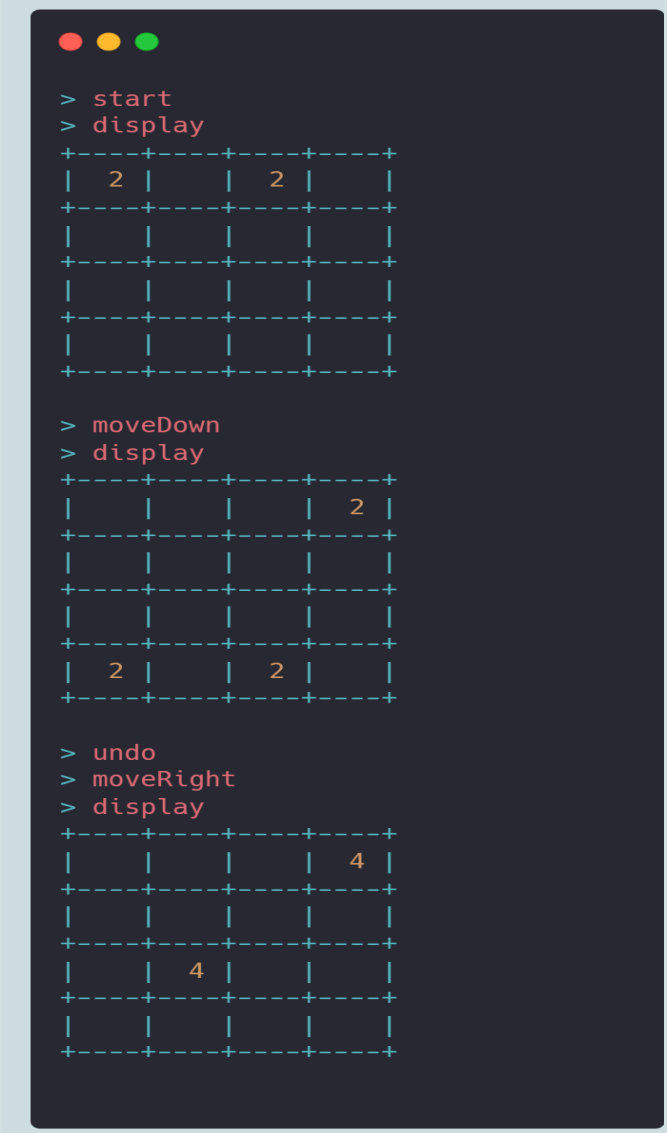
1. عملیات Undo: هر بار که کاربر حرکتی انجام می‌دهد، وضعیت فعلی صفحه ذخیره می‌شود. اگر کاربر بخواهد حرکت خود را لغو کند، می‌تواند وضعیت قبلی را بازیابی کند.
2. عملیات Redo: پس از انجام یک عملیات Undo، کاربر می‌تواند وضعیت قبلی لغو شده را دوباره به صفحه بازگرداند.
3. محدودیت تعداد عملیات‌ها: برای جلوگیری از استفاده بیش از حد، کاربر تنها قادر به انجام 5 عملیات Undo و 5 عملیات Redo در طول بازی خواهد بود.

**پیاده سازی رابط کاربری متنی بازی:**

این بازی دارای یک رابط متنی است که به کاربر امکان می‌دهد از طریق وارد کردن دستورات متنی، حرکات خود را در بازی کنترل کند.

- بازی با وارد کردن دستور start شروع می‌شود.
- کاربر می‌تواند با وارد کردن یکی از دستورات moveDown، moveRight، moveLeft، moveUp، تمام کاشی‌ها را در آن جهت حرکت دهد.
- برنامه با دستور display به‌روزرسانی جدیدی از صفحه بازی نمایش می‌دهد و نتیجه حرکت (مانند ادغام و تغییر موقعیت کاشی‌ها) و همچنین امتیاز کاربر را به او نشان می‌دهد.

این فرآیند تا زمانی ادامه می‌یابد که کاربر به هدف نهایی (یعنی رسیدن به عدد 2048) برسد یا دیگر حرکت ممکنی باقی نماند.



**نمرات اضافه:**

- پیاده سازی بازی به صورت گرافیکی
- پیاده سازی یک پلیر به صورت هوش مصنوعی

**نکات تکمیلی:**

- استفاده از هر زبان برنامه نویسی برای انجام پروژه مجاز است.
- در صورت نیاز می‌توانید متدهای کمکی پیاده‌سازی کنید.
- پیاده سازی ساختمان داده های ذکر شده به عهده شماست و نباید از کتابخانه های آماده زبان استفاده شود.
- انجام پروژه به صورت انفرادی است.
- در صورت مشاهده شباهت غیر متعارف میان پروژه افراد، نمره 100- برای هر دو نفر در نظر گرفته میشود.
- تسلط به بخش‌های مختلف پروژه در هنگام تحویل الزامی است.
-در صورت هرگونه ابهام می‌توانید با fatemeh_dehbashii یا Hanie_ghp در ارتباط باشید.
- فایل‌های نهایی پروژه خود را در قالب زیر در سامانه vu بارگذاری کنید:

FirstNameLastName_StudentNumber_PR1.zip

**موفق باشید**