

پروژه میانترم درس ساختمان داده ها نیمسال تحصیلی ۱۴۰۳-۱۴۰۲


توضیحات پروژه:

میخواهیم در این پروژه بازی «Water Sort» را پیاده سازی کنیم.
جهت آشنایی بیشتر با این بازی روی [این](#) کلیک کنید.
شما باید Class ی به نام WaterSortGame بسازید که دارای یک Linked List از Stack ها باشد که هر Stack نشاندهنده یک Bottle است.

طرز کارکرد برنامه:

شما باید یک رابط کنسولی برای این برنامه درست کنید که حرکاتی که کاربر میخواند انجام دهد را با وارد کردن عباراتی که بعداً توضیح داده خواهد شد، انجام دهد و نتیجه را بعد از هر حرکت در کنسول نمایش دهد. این کار تا زمانی که کاربر برنده شود ادامه خواهد داشت و در صورت برنده شدن کاربر عبارت You Won! در کنسول چاپ خواهد شد.

توضیحات مربوط به کلاس WaterSortGame:



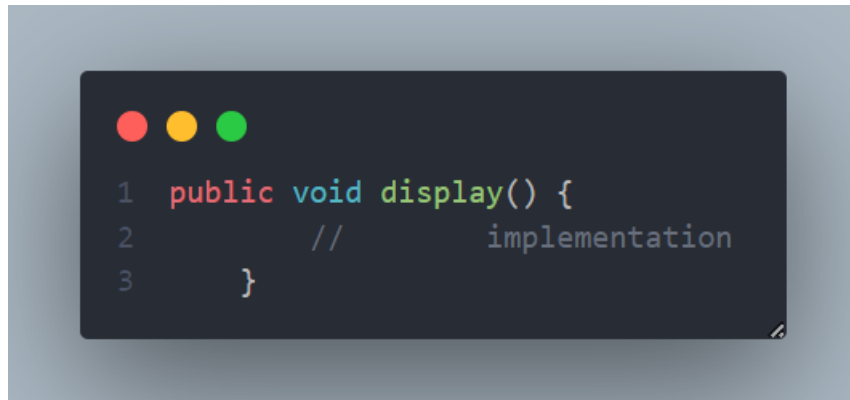
```
1 public class WaterSortGame {  
2     public WaterSortGame(String[] colors, int maxBottleSize) {  
3         // implementation  
4     }  
5 }  
6
```

:Constructor

کلاس WaterSortGame کلاس اصلی برنامه ماست که constructor آن آرایه رنگ های بازی که هر رنگ یک String متمایز است و ماکسیمم سایز هر بطری رو میگیرد. هر بطری بر اساس ماکسیمم سایز تقسیم میشود و رنگ هارا میتوان در آن ریخت. هر بطری یک یا چند قسمت از هر رنگ خواهد داشت و هدف بازی این است که یک بطری که تمام قسمت های آن از یک رنگ پر شده باشد داشته باشیم. طبق قوانین بازی اگر فرضاً

N رنگ داشته باشیم باید N + 1 بطری داشته باشیم که N تای آن با رنگ ها پر شده است و 1 بطری خالی است. در ابتدا بطری ها با رنگ ها به صورت random پر میشود.

:Display



متد display وضعیت هر یک از بطری ها را در کنسول چاپ میکند خروجی باید به شکل زیر باشد:

```
Red   Blue   Yellow Empty Blue  
Red   Yellow Green Empty Green  
Green Red    Red   Empty Green  
Yellow Blue   Blue  Empty Yellow  
---#---
```

به عبارتی خروجی را مانند یک جدول چاپ کند که هر ستون از چپ به راست نشاندهنده یک بطری است.

نیازی نیست که خروجی دقیقا به شکل بالا باشد و چاپ جدول String رنگ ها کافی است. علامت ---#--- انتخاب شدن است که در ادامه توضیح داده خواهد شد. Empty هم به معنای خالی بودن است.

در این مثال maxBottleSize 4 است.

:Select, DeSelect, SelectNext, SelectPrevoius

```
1 public Boolean select(int bottleNumber) {
2     //      implementation
3     return isSelected;
4 }
5 public void deSelect() {
6     //      implementation
7 }
8 public void selectNext() {
9     //      implementation
10 }
11 public void selectPrevois() {
12     //      implementation
13 }
```


تابع select شماره بطری را می گیرد و آن بطری Select میشود 😊 و isSelected را برمیگرداند که نشاندهنده این است که بطری انتخاب شده است یا نه. در صورتی که تمام قسمت های بطری از یک رنگ پر شده باشد، غیر قابل انتخاب است یا اینکه بطری کاملاً خالی باشد. توابع selectPrevious و selectNext به صورت circular خواهند بود.

:Pour

```
1 public Boolean pour(int bottleNumber) {
2     //      implementation
3     return isPoured;
4 }
```

رنگ بالای بطری ای که انتخاب شده است را در بطری با شماره `bottleNumber` تا جایی که جا داشته باشد میریزد. برای اینکه بتوان از یک بطری در بطری دیگری رنگی ریخت یا باید رنگ هردو یکسان باشد یا بطری دوم خالی باشد. اگر رنگی ریخته شد مقدار `isPoured` مقدار `true` خواهد داشت در غیر این صورت مقدار `false` خواهد داشت. اگر هم بطری ای هم انتخاب نشده باشد عملی صورت نمیگیرد و مقدار `false` برمی گرداند.


:Swap



```
1 public void swap(int bottleNumber) {
2     // implementation
3 }
```

جای بطری با شماره `bottleNumber` را با بطری `select` شده عوض میکند.

:ReplaceColor



```
1 public void replaceColor(String firstColor, String secondColor) {
2     // implementation
3 }
4 }
```

رنگ اول را در بطری ها در صورت وجود با رنگ دوم عوض میکند. رنگ دوم باید با رنگ های موجود متفاوت باشد.

AddEmptyBottle

```
1 public void addEmptyBottle() {  
2     //      implementation  
3 }
```

بطری ای به اندازه نصف `maxBottleSize` به انتهای لیست بطری ها اضافه میکند. کاربر تا انتهای بازی فقط یک بار میتواند از این قابلیت استفاده کند.

:hasWon

```
1 public Boolean hasWon() {  
2     //      implementation  
3     return hasWon;  
4 }
```

در صورت برنده شدن کاربر مقدار `true` برمیگرداند.

:Undo, Redo

```
1 public void undo() {  
2     //      implementation  
3 }  
4 public void redo() {  
5     //      implementation  
6 }
```

تمامی اعمال انجام شده توسط کاربر قابلیت undo , redo خواهند داشت.

رابط کنسولی: در شروع بازی یک object از کلاس از WaterSortGame میسازید که کاربر رنگ ها و maxBottleSize را مشخص خواهد کرد. بعد از هر حرکت کاربر تابع display صدا زده خواهد شد. ورودی کاربر به شکل های زیر است:

```
1 دستور شروع بازی start  
2 رنگ ها green red yellow  
3 ماکسیمم سائز بطری 4  
4  
5 دستور انتخاب بطری سوم select 3  
6 deSelect  
7 selectNext  
8 selectPrevious  
9 دستور ریختن رنگ در بطری سوم pour 3  
10 دستور عوض کردن بطری با بطری چهارم swap 4  
11 دستور عوض کردن رنگ زرد با آبی ReplaceColor yellow blue  
12 undo  
13 redo  
14 addEmptyBottle
```

کاربر در ابتدا حتما باید بازی را start کند.

نمره اضافه:

تابعی که بتواند این بازی را تا برنده شدن انجام دهد دارای نمره اضافی خواهد بود.
حالت سخت تر بازی: کاربر فقط بتواند 5 بار در طول کل بازی undo یا redo انجام دهد و تابعی پیاده سازی شود که اگر کاربر دیگر نتواند با حرکات مجاز ببرد You lost! را در کنسول چاپ کند.

نکات تکمیلی:

استفاده از هر زبان برنامه نویسی برای انجام پروژه مجاز است.
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 هر گونه شباهت و اثبات تقلب میان دو فرد، نمره ۱۰۰ - به هر دو فرد داده میشود.
فایل های نهایی پروژه خود را در قالب زیر در سامانه VU بارگزاری کنید:
FirstName_LastName_StudentNumber_PR1.zip

موفق باشید 😊