هدف از این پروژه پیاده سازی بازی UNO است.نکاتی که در پیاده سازی این بازی مطرح است:
1- از کارت wildDraw تنها زمانی می توان استفاده کرد که هیچ کارت دیگری قابل بازی نباشد!(اگر هیچ کارت قابل بازی در دست نداشته باشیم و تنها یک کارت wildDraw و یک کارت wildColor داشته باشیم، استفاده از کارت wildColor نسبت به wildDraw اولویت دارد!!)
2-از کارت wildColor هر زمانی که بخواهیم می توانیم استفاده کنیم.
3-در هنگام وارد کردن ورودی،زمانی که بازیکن انسان داریم، باید حرف اول کارت های حرکتی بزرگ، کارت های وحشی به صورت camelCase ورنگ ها با حروف کوچک نوشته شوند. در غیر این صورت یک پیام Incorrect input چاپ شده و بازیکن نوبت خود را از دست می دهد.
4-زمانی که کارتی که روی زمین است یک کارت Draw2 است، نفر بعدی تنها در صورتی می تواند جریمه ی کارت را بر ندارد که یک کارت Draw2 روی آن بگذارد.اگر هنگامی که بازیکن، انسان است یک کارت Draw2 داشته باشد ولی یک کارت دیگر را بخواهد بازی کند، پیغام خطا ایجاد شده و فرد مجبور میشود جریمه ی کارت را بردارد و نوبت خود را از دست بدهد.(برای کارت wildDraw نیز همین گونه است)
این پروژه حاوی 9 کلاس است و در پیاده سازی آن از ارث بری استفاده شده است. همچنین قسمت امتیازی پروژه، یعنی پیاده سازی به صورت بازی n نفره ، انجام شده است. کلاس های این پروژه شامل: کلاس Main که کلاس اصلی پروژه است، کلاس Deck که مربوط به زمین بازی است، کلاس Player که مربوط به بازیکن های کامپیوتری ، کلاس User که مربوط به بازیکن های انسان است، کلاس Card که مربوط به کارت هاست، کلاس WildCard که مربوط به کارت های وحشی، کلاس ColoredCard که مربوط به کارت های رنگی است، کلاس ActionCard که مربوط به کارت های حرکتی است و کلاس NumericCard که مربوط به کارت های عددی است.

کلاس Card:

این کلاس ، یک سوپر کلاس برای ایجاد یک کارت است و تنها یک فیلد scoreNumber دارد که مشخص کننده ی امتیاز کارت است.متد های این کلاس تنها متد های getter و setter برای scoreNumber هستند.

کلاس WildCard:

این کلاس از کلاس card ارث بری میکند برای ایجاد یک کارت وحشی است.فیلد های این کلاس شامل یک رشته برای نوع کارت(wildColor or wildDraw) ، یک رشته برای رنگ کارت های بعدی(زمانی که یک کارت وحشی بازی می شود باید رنگ کارت های بعدی مشخص شود)، یک boolean برای مشخص کردن فعال بودن یا نبودن کارت wildDraw و یک int برای مشخص کردن اندازه ی جریمه ی کارت wildDrawاست. در کانستراکتور این کلاس، امتیاز هر کارت برابر با 50 ، نوع کارت برابر با نوع داده شده ، boolean مشخص کننده ی فعال بودن یا نبودن =true و در صورتی که کارت از نوع wildDraw باشد، جریمه ی آن =4 قرار داده می شود.
متد های این کلاس شامل:
تمامی setter و getter های مورد استفاده

متد doublingWildDrawValue :
در صورتی که یک کارت wildDraw بر روی یک کارت wildDraw دیگر بازی شود، این متد فراخوانی شده و یک int value که مشخص کننده ی اندازه ی جریمه ی کارت قبلی است را به عنوان ورودی می گیرد و آن مقدار را 4 واحد افزایش داده و به عنوان میزان جریمه ی این کارت ، set می کند.

کلاس ColoredCard:

این کلاس از کلاس card ارث بری میکند و برای ایجاد یک کارت رنگی است. فیلدهای این کلاس تنها شامل یک رشته برای مشخص کردن رنگ کارت است که در کانستراکتور مقداردهی می شود.
متد این کلاس نیز تنها یک getter برای رنگ کارت است.

کلاس ActionCard:

این کلاس از کلاس ColoredCard ارث بری میکند و برای ایجاد یک کارت حرکتی است.فیلد های این کلاس شامل یک رشته برای مشخص کردن نوع حرکت، یک boolean برای مشخص کردن فعال بودن یا نبودن کارت و یک int برای مشخص کردن میزان جریمه ی کارت های Draw2 است.در کانستراکتور این کلاس، امتیاز هر کارت برابر با 20،نوع حرکت برابر با نوع حرکت داده شده، رنگ کارت برابر با رنگ داده شده، boolean مشخص کننده ی فعال بودن یا نبودن =true و در صورتی که کارت یک کارت Draw2 باشد، جریمه ی آن =2 قرار داده می شود.

متد های این کلاس شامل:

تمامی getter ها و setter های مورد استفاده

متد doublingDraw2CardValue :

در صورتی که یک کارت Draw2 بر روی یک کارت Draw2 دیگر بازی شود، این متد فراخوانی شده و یک int value که مشخص کننده ی اندازه ی جریمه ی کارت قبلی است به عنوان ورودی می گیرد و آن مقدار را 2 واحد افزایش داده و به عنوان میزان جریمه ی این کارت ، set می کند.

کلاس NumericCard :

این کلاس از کلاس ColoredCard ارث بری میکند و برای ایجاد یک کارت عددی است.فیلد این کلاس تنها شامل یک int برای مشخص کردن عدد روی کارت است. کانستراکتور این کلاس، امتیاز هر کارت را برابر عدد روی کارت قرار می دهد و رنگ کارت را نیز برابر رنگ داده شده قرار می دهد.

متد این کلاس تنها شامل یک getter برای عدد روی کارت است.

کلاس Deck :

این کلاس برای ایجاد یک deck برای بازی است.فیلد های این کلاس شامل: یک رشته برای تعیین جهت بازی،یک لیست از کارت های میز ،یک کارت رو شده به عنوان کارت وسط، یک رشته برای تعیین رنگ کارت وسط، یک int برای تعیین عدد کارت وسط و یک لیست از بازیکن هاست.کانستراکتور این کلاس یک لیست از بازیکن ها را به عنوان ورودی میگیرد و همه ی بازیکن ها را به لیست بازیکن های کلاس اضافه می کند. سپس

کارت های میز را با فراخوانی متد setDeckCards که توضیح آن در ادامه آمده است، set می کند.

سپس کارت ها را بُر می زند و به هر بازیکن 7 کارت با فراخوانی متد dealCards می دهد. در ادامه ، ابتدا اولین کارت از کارت های میز را به عنوان کارت وسط قرار می دهد سپس چک می کند که اگر این کارت از کارت های وحشی باشد، کارت ها را تا زمانی بُر می زند که اولین کارت از کارت های رنگی باشد و آن کارت را به عنوان کارت وسط قرار می دهد(فراخوانی متد setFaceUpCard) و از لیست کارت های میز آن را حذف میکند.سپس لیست بازیکن ها را بُر می زند تا یک نفر به صورت تصادفی نفر شروع کننده(نفر اول لیست) شود.همچنین direction اولیه را نیز برابر clockwise قرار می دهد.

متد های این کلاس شامل:

همه ی getter و setter های مورد استفاده

متد setDeckCards :

این متد 3 لیست از کارت های WildCard,ActionCard,NumericCard ایجاد کرده و به متد های مربوطه پاس می دهد که آنها را پر کنند. سپس همه ی آنها را به کارت های میز اضافه می کند.

متد setWildCards :

این متد یک لیست از کارت های WildCard گرفته ، 4 کارت wildColor و 4 کارت wildDraw ایجاد می کند و آنها را به لیست WildCards اضافه می کند.

متد setNumericCards :
این متد یک لیست از کارت های NumericCard گرفته و دو سری کارت عددی از هر رنگ ایجاد می کند(البته از عدد 0 فقط یکی از هر رنگ داریم) و به لیست کارت های NumericCards اضافه می کند.

متد setActionCards :
این متد یک لیست از کارت های ActionCard گرفته و هر action را برای هر رنگ دو بار ایجاد کرده و به لیست کارت های ActionCards اضافه می کند.

متد dealCards :
این متد یک بازیکن گرفته و یک لیست 7 تایی از کارت ها ایجاد کرده و 7 کارت از اول لیست کارت های میز را به این لیست اضافه کرده و سپس این لیست 7تایی را از لیست کارت های میز حذف کرده و بعد این لیست 7 تایی را به عنوان کارت های بازیکن set می کند.

متد setFaceUpCard :
این متد یک کارت میگیرد و ابتدا کارت وسط قبلی را به آخر لیست کارت های میز اضافه می کند و سپس این کارت را به عنوان کارت وسط set می کند و سپس اگر این کارت یک کارت ColoredCard باشد رنگ میز را= رنگ آن قرار می دهد و اگر این کارت یک کارت NumericCard باشد عدد میز را=عدد آن کارت قرار می دهد. اگر این کارت یک کارت ActionCard باشد و حرکت آن از نوع Reverse باشد، جهت میز را با فراخوانی متد changeDirection عوض می کند و فعالیت آن کارت را false می کند.اگر این کارت یک کارت WildCard باشد، رنگ میز را با فراخوانی متد getNextCardColor برابر رنگ set شده برای کارت های بعدی کارت wild ، قرار می دهد.

متد changeDirection :
این متد جهت میز را عوض می کند. اگر clockwise باشد آن را = anticlockwise و اگر anticlockwise باشد جهت میز را =clockwise قرار می دهد.

متد isEnd :
این متد چک میکند که اگر سایز کارت های حداقل یک بازیکن =0 شده است، یعنی بازی تمام شده است و true بر میگرداند. در غیر این صورت False بر میگرداند.

متد printScores :
این متد امتیاز های بازیکن ها را بعد از اتمام بازی، به ترتیب از کم به زیاد و با فراخوانی متد getScore برای هر بازیکن چاپ می کند. عملکرد به این گونه است که ابتدا امتیاز های همه ی بازیکن ها داخل یک لیست ریخته می شود و سپس sort شده و سپس با یک پیمایش بر روی این امتیاز ها و یک پیمایش دیگر بر روی لیست بازیکن ها در داخل آن پیمایش، بازیکنی که امتیاز آن برابر آن امتیاز است را پیدا کرده و اسم او و امتیازش را چاپ میکند.

متد printNumberOfCards :
این متد تعداد کارت های هر بازیکن را چاپ میکند.

کلاس Player :

این کلاس برای ایجاد یک بازیکن کامپیوتری است.فیلد های این کلاس شامل : یک لیست از کارت های بازیکن، یک رشته ی مشخص کننده ی اسم بازیکن و یک int برای مشخص کردن index بازیکن بعدی است.در کانستراکتور، اسم مقداردهی می شود.

متد های این کلاس شامل:
تمامی getter و setter های مورد نیاز

متد setNextPlayer :
این متد برای مشخص کردن این است که نفر بعدی که نوبتش است، index ش درلیست بازیکن های موجود در میز بازی چند است.
این متد deck اصلی بازی را به عنوان ورودی می گیرد و سپس اگر جهت چرخش میز ساعتگرد باشد،index این بازیکن را یک واحد افزایش می دهد و به عنوان index بازیکن بعدی قرار می دهد. اگر این مقدار بزرگ تر یا مساوی اندازه ی لیست بازیکن های میز -1 باشد، یعنی به آخر لیست رسیده ایم، پس بازیکن بعدی، نفر اول لیست است.پس index بازیکن بعدی را برابر 0 قرار می دهد.
اگر جهت چرخش میز پادساعتگرد باشد، index این بازیکن را یک واحد کاهش می دهد و به عنوان index بازیکن بعدی قرار می دهد.اگر این مقدار کوچک تر از 0 باشد یعنی به اول لیست رسیده ایم، پس بازیکن بعدی، نفر آخر لیست است.پس index را برابر (اندازه ی لیست -1) قرار می دهد.

متد drawCard :
این متد برای زمانی است که بازیکن باید یک یا چند کارت از deck بردارد.این متد یک عدد مشخص کننده ی تعداد کارت ها و deck اصلی بازی را میگیرد و یک لیست از کارت ها می سازد و سپس به تعداد کارت های داده شده ،از اول لیست کارت های میز کارت بر می دارد و به لیست کارت های بازیکن اضافه می کند و آنها را داخل یک لیست دیگر ریخته و سپس همه ی کارت های آن لیست را از لیست کارت های میز حذف می کند.

متد playerTurn :
این متد برای پیاده سازی اتفاقاتی است که در هر نوبت بازیکن باید بیفتد.
این متد deck اصلی بازی و یک int مشخص کننده ی index بازیکن در لیست بازیکن های deck اصلی را میگیرد، ابتدا جهت بازی و تعداد کارت های همهی بازیکن ها را چاپ کرده و سپس لیست کارت های فرد و کارت وسط میز را به او نشان می دهد.
بعد چک میکند که اگر کارت وسط میز یک کارت wildDraw باشد، متد مربوطه را صدا می زند(wildDrawCardFaceUp) و سپس چک می کند که اگر آن متد True برگرداند، یعنی نوبت بازیکن تموم شده است و در نتیجه متد setNextPlayer فراخوانی شده و نوبت بازیکن تمام می شود.
اگر کارت وسط یک کارت wildColor باشد، متد مربوط به آن را صدا می زند.(wildColorCardFaceUp)
اگر کارت وسط یک کارت action باشد، متد actionFaceUpCard صدا می زند و سپس چک می کند که اگر آن متد True برگرداند، یعنی نوبت بازیکن تموم شده است و در نتیجه متد setNextPlayer فراخوانی شده و نوبت بازیکن تمام می شود.
اگر کارت وسط یک کارت numeric باشد، متد numericCardFaceUp را صدا می زند.
سپس متد playOneCard را فراخوانی میکند تا بازیکن یک کارت بازی کند. بعد چک می کند که اگر سایز کارت های بازیکن 1 است، UNO را چاپ می کند. در آخر متد setNextPlayer را فراخوانی می کند.

متد wildDrawCardFaceUp :
این متد زمانی اجرا می شود که کارت روی زمین یک کارت wildDraw باشد. اگر این متد true برگرداند به این معنی است که نوبت بازیکن تمام شده است و بازیکن کار دیگری نباید انجام بدهد. در غیر این صورت false بر میگرداند.
ابتدا چک میکند که این کارت هنوز فعال است یا نه.اگر فعال نیست این کارت مثل یک کارت wildColor رفتار کرده و متد wildColorCardFaceUp را صدا می زند و متد false بر میگرداند.
اگر کارت فعال است، چک میکند که در بین کارت های بازیکن کارت wildDraw وجود دارد یا نه.اگر بازیکن کارت wildDraw داشته باشد، این متد آن کارت را برای بازیکن بازی کرده و به عنوان کارت وسط قرار می دهد، اندازه ی جریمه ی آن را با صدا زدن متد doublingWildDrawValue افزایش داده و از لیست کارت های بازیکن حذف کرده و با صدا زدن متد chooseColor نیز رنگ کارت بعدی را مشخص می کند و کارت را به همراه رنگ مشخص شده چاپ می کند و متد true برگردانده و نوبت بازیکن تمام میشود.

اگر بازیکن کارت wildDraw نداشته باشد، متد drawCard را صدا زده و به اندازه ی جریمه ی کارت wildDraw ،کارت برای بازیکن بر می دارد و فعالیت کارت wildDraw را نیز false می کند ومتد True بر گردانده و نوبت بازیکن تمام می شود.

متد wildColorCardFaceUp:
این متد زمانی اجرا می شود که کارت وسط یک کارت wildColor یا یک کارت wildDraw غیر فعال باشد.
این متد با یک پیمایش بر روی لیست کارت های بازیکن، کارت هایی که رنگشان با رنگ set شده برای میز(توسط بازیکن قبلی) یکی است را به لیست کارت های قابل بازی برای بازیکن اضافه می کند.

متد actionCardFaceUp :
این متد زمانی اجرا می شود که کارت وسط یک کارت action باشد. اگر این متد true برگرداند به این معنی است که نوبت بازیکن تمام شده است و بازیکن کار دیگری نباید انجام بدهد. در غیر این صورت false بر میگرداند.
ابتدا چک می کند که اگر این کارت نوعش از نوع Skip باشد و کارت فعال باشد، فعالیت کارت را false کرده و متد true بر میگرداند و نوبت بازیکن تمام می شود.(بازیکن نوبتش را از دست می دهد)
اگر فعال نبود، متد actionNotActiveCardFaceUp را صدا می زند و False بر میگرداند.
اگر نوع کارت از نوع کارت Reverse بود، قبلا همه ی کار ها انجام شده، پس متد actionNotActiveCardFaceUp را صدا می زند و False بر میگرداند.
اگر نوع کارت از نوع Draw2 باشد، ابتدا چک میکند که کارت فعال است یا خیر.اگر فعال بود در بین کارت های بازیکن جست و جو میکند که آیا کارت Draw2 دارد یا نه.اگر داشت آن کارت را برای شخص بازی کرده و جریمه ی آن کارت را با فراخوانی متد doublingDraw2CardValue افزایش داده و این کارت را به عنوان کارت وسط قرار داده و از لیست کارت های بازیکن حذف می کند و کارت را چاپ می کند و نوبت بازیکن به پایان رسیده و true بر میگرداند.
اگر بازیکن کارت Draw2 نداشت، متد drawCard را فراخوانی کرده و به اندازه ی جریمه ی کارت Draw2 برای فرد کارت بر میدارد و نوبت فرد به پایان رسیده و true بر میگرداند.
اگر کارت Draw2 فعال نبود، متد actionNotActiveCardFaceUp را صدا می زند و False بر میگرداند.

متد actionNotActiveCardFaceUp:
این متد زمانی اجرا می شود که کارت وسط یک کارت action غیر فعال باشد.(مثلا نفر a یک کارت Skip گذاشته،نفر b نوبتش را از دست داده و حالا نوبت نفر c است. کارت وسط هنوز Skip است ولی غیرفعال) این متد با یک پیمایش بر روی لیست، کارت هایی که رنگشان با رنگ کارت وسط یکی باشد یا action آنها با action کارت وسط یکی باشد را به لیست کارت های قابل بازی برای فرد اضافه می کند.

متد numericCardFaceUp :
این متد زمانی اجرا می شود که کارت وسط یک کارت عددی باشد.
این متد با یک پیمایش بر روی لیست کارت ها، کارت هایی که رنگ یا عددشان با رنگ یا عدد کارت وسط یکی است را به لیست کارت های قابل بازی اضافه می کند.

متد playOneCard :
این متد ابتدا با یک پیمایش بر روی لیست کارت ها، کارت های wildColor را به لیست کارت های قابل بازی اضافه میکند.(زیرا کارت های wildColor را هر زمانی میتوانیم بازی کنیم) سپس چک میکند که اگر سایز کارت های قابل بازی برای بازیکن =0 است یعنی بازیکن هیچ کارتی برای بازی کردن ندارد، پس متد noPlayableCards را صدا می زند.

اگر بازیکن کارت قابل بازی داشته باشد، یک عدد رندم بین 0 تا size-1 تولید کرده و آن کارت را از لیست کارت های قابل بازی انتخاب کرده و بازی میکند و از لیست کارت های بازیکن حذف کرده و به عنوان کارت وسط set می کند و سپس آنرا print می کند .(اگر کارت یک کارت wildColor بود یک رنگ نیز با فراخوانی متد chooseColor برای کارت های بعدی set می کند.)

متد noPlayableCards :
این متد زمانی فراخوانی می شود که هیچ کارت قابل بازی برای فرد وجود نداشته باشد.ابتدا با یک پیمایش بر روی لیست کارت های فرد چک میکند که اگر فرد کارت wildDraw دارد آن کارت را برای شخص بازی میکند و یک رنگ نیز با فراخوانی متد chooseColor برای کارت های بعدی set می کند و کارت را به عنوان کارت وسط set کرده و از لیست کارت های فرد حذف می کند و کار متد به پایان می رسد.
اگر کارت wildDraw نداشت، با فراخوانی متد drawCard یک کارت برای شخص بر میدارد و سپس چک میکند که اگر این کارت از نوع کارت های wild بود یا از نظر color,action,number با کارت وسط یکی بود، آن را بازی میکند و کارت را به عنوان کارت وسط set کرده و از لیست کارت های فرد حذف می کند.(اگر کارت یک کارت wildColor بود یک رنگ نیز با فراخوانی متد chooseColor برای کارت های بعدی set می کند.)

متد printOneCard :
یک کارت میگیرد و آن راچاپ می کند.

متد printAllCards:
همه ی کارت های در دست فرد را به صورت یک ردیف چاپ میکند، به گونه ای که ابتدا با صدا زدن متد printFirstLine ، خط اول ، سپس با صدا زدن print2and4Line خط دوم و سپس با صدا زدن متد printThirdLine خط سوم و سپس دوباره متد های print2and4Line و بعد printFirstLine را صدا می زند.

متد printFirstLine :
خط اول همه ی کارت ها را چاپ می کند.

متد print2and4Line :
خط دو یا چهارم(شبیه به هم اند) همه ی کارت ها را چاپ می کند.

متد printThirdLine :
خط سوم همه ی کارت ها را چاپ می کند.

متد getColor :
یک رنگ گرفته و کد پرینتی آن را بر میگرداند(برای چاپ رنگی)

متد getScore :
این متد برای محاسبه ی امتیاز بازیکن است. با یک پیمایش بر روی لیست کارت ها، امتیاز ها را با هم جمع کرده و یک int برمیگرداند.

متد chooseColor :
این متد زمانی اجرا می شود که بازیکن بخواهد یک کارت wild بازی کند.**(این متد برای بهینه سازی بازی است!)**
این متد یک پیمایش بر روی لیست کارت های بازیکن انجام داده و رنگی که بیشترین تکرار را دارد ، بر میگرداند.

کلاس User :

این کلاس از کلاس Player ارث بری می کند و برای ایجاد یک بازیکن انسان است.این کلاس برای خودش فیلدی نداشته و در کانستراکتور فیلد سوپر کلاس مقداردهی می شود.
متد های override شده شامل:

متد wildDrawCardFaceUp:
عملکرد این متد دقیقا مشابه با متد سوپر کلاسش است ولی تفاوت در این است که اگر کارت وسط یک کارت wildDraw فعال بود و بازیکن کارت wildDraw داشت باید ابتدا از بازیکن بپرسد که کدام یک از کارت های خود را می خواهد بازی کند و در صورتی که بازیکن کارت wildDraw را انتخاب کرد، آن کارت wildDraw را برای بازیکن بازی میکند، متد playWildCard رافراخوانی کرده، جریمه ی کارت را افزایش داده و متد true برگردانده و نوبت بازیکن تمام میشود.
در غیر این صورت پیغام wrong input چاپ شده و بازیکن باید به اندازه ی جریمه ی کارت wildDraw وسط، کارت بردارد و متد true برگردانده و نوبت بازیکن تمام میشود.
سایر عملکرد های این متد شبیه به متد سوپرکلاسش است.

متد actionCardFaceUp :
عملکرد این متد دقیقا مشابه با متد سوپر کلاسش است ولی تفاوت در این است که اگر کارت وسط یک کارت Draw2 فعال بود و بازیکن کارت Draw2 داشت، یک لیست از کارت های Draw2 بازیکن تشکیل داده شده و سپس ابتدا از بازیکن بپرسد که کدام یک از کارت های خود را می خواهد بازی کند و در صورتی که بازیکن کارت درستی(یکی از کارت های Draw2 خود) را انتخاب کرد، آن کارت Draw2 را برای بازیکن بازی میکند، جریمه ی کارت را افزایش داده و متد true برگردانده و نوبت بازیکن تمام میشود.
در غیر این صورت پیغام wrong input چاپ شده و بازیکن باید به اندازه ی جریمه ی کارت Draw2 وسط، کارت بردارد و متد true برگردانده و نوبت بازیکن تمام میشود.
سایر عملکرد های این متد شبیه به متد سوپرکلاسش است.

متد playOneCard :
عملکرد این متد نیز مشابه متد سوپر کلاسش است با این تفاوت که به جای رندم انتخاب کردن کارت، از کاربر میخواهد تا یک کارت انتخاب کند و در صورتی که ورودی وارد شده معتبر بود و کارت جزو کارت های قابل بازی برای کاربر بود، آن کارت را بازی کرده و به عنوان کارت وسط set کرده و از لیست کارت های فرد حذف میکند.
در صورتی که کارت از نوع wild باشد ،متد playWildCard فراخوانی می شود.
اگر فرد یک کارت wildDraw انتخاب کند، آنگاه چک می شود که اگر هیچ کارت دیگری قابل بازی نباشد، آنگاه فرد می تواند این کارت را بازی کند.
در صورت وارد کردن ورودی اشتباه یا انتخاب کارت اشتباه، فرد نوبتش را از دست می دهد.

متد noPlayableCards:
این متد دقیقا عملکردی مشابه با سوپر کلاس دارد با این تفاوت که در ابتدا یک کارت از deck برای شخص بر میدارد، سپس اگر کارت قابل بازی بود از فرد می پرسد که آیا می خواهد این کارت را بازی کند یا نه.اگر خواست،آن کارت را برای آن فرد بازی میکند.اگر نخواست نوبتش را از دست می دهد.

متد های private کلاس:

متد playWildCard :
این متد مخصوص خود کلاس است و برای زمانی است که کاربر میخواهد یک کارت wild بازی کند و یک کارت wild و deck اصلی بازی را گرفته، ابتدا از کاربر می خواهد که یک رنگ برای کارت های بعدی انتخاب کرده و سپس رنگ کارت های بعدی را set کرده و سپس کارت انتخاب شده را به عنوان کارت وسط set کرده و از لیست کارت های فرد حذف می کند.

کلاس Main :

این کلاس ، کلاس اصلی بازی است. در ابتدا یک آرایه از بازیکن ها تشکیل داده و سپس از کاربر می خواهد که تعداد بازیکن های کامپیوتری، نام آنها، تعداد بازیکن های انسان و نام آنها را انتخاب کندو آنها را به لیست بازیکن ها اضافه می کند.سپس یک آبجکت از کلاس Deck به عنوان mainDeck بازی ایجاد کرده و سپس در یک حلقه ی بینهایت تا زمانی که متد isEnd در کلاس Deck ،false بر نگردانده، اسم فردی که نوبتش هست را چاپ کرده و متد playerTurn را نیز برای آن فرد صدا می زند.(این که نوبت چه کسی است را با استفاده از صدازدن getNextPlayer برای player قبلی متوجه می شود) با خارج شدن از حلقه ی while ، همه ی امتیاز ها چاپ می شود.

توضیحات کامل هر متد در جاواداک نوشته شده است.