

**UNO Card Game**

Group 1

Muhammad Nabiel Rayhan Falaah

Darren Arqiarkaan Dyazfajri Teddy

Baskoro Bayu Baruno

Names :

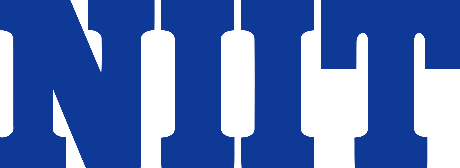
Class : 2SE1

Faculty : M. Riza Iqbal Latief, ST.

**Continuing Education Program Center for Computing and Information Technology**

**Faculty of Engineering University of Indonesia**

**2023**

  
  
PROJ

Developed by :

**Group 1**

1. **Muhammad Nabiel Rayhan Falaah**
2. **Darren Arqiarkaan Dyazfajri Teddy**
3. **Baskoro Bayu Baruno**

**PROJECT ON***UNO Card Game*

**UNO Card Game**

Batch Code : 2SE1

Start Date : March 21, 2023

End Date : April 10, 2023

Name of the Coordinator : M. Riza Iqbal Latief, ST.

Names of Developer:

* Muhammad Nabiel Rayhan Falaah
* Darren Arqiarkaan Dyazfajri Teddy
* Baskoro Bayu Baruno

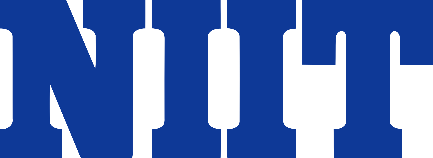
Date of Submission: April 10, 2023

**CERTIFICATE**

This is to certify that this report titled “UNO Card Game” embodies the original work done by Muhammad Nabiel Rayhan Falaah, Darren Arqiarkaan Dyazfajri Teddy, Baskoro Bayu Baruno. Project in partial fulfillment of their course requirement at NIIT.

Coordinator :

M. Riza Iqbal Latief, ST.



Depok, November 2017

Group 2

**ACKNOWLEDGMENT**

Authors would like to praise to Allah, Most Merciful bless and so authors can finish this project. Author also would like to thanks to M. Riza Iqbal Latief, ST. as lecturer who has given useful suggestion which are help author in writing this paper. This paper titled “UNO Card Game”. The writer wrote it to fulfill project NIIT.

Author hopes this paper can improve the knowledge and experience for the readers, and for the future the author can enrich the contents of the paper.

Due to our limited knowledge and experience, author believe there are still many shortcomings in this paper, therefore authors are looking to forward to constructive suggestions and criticism from readers for the perfection of this paper.

Depok, April 2023

Authors

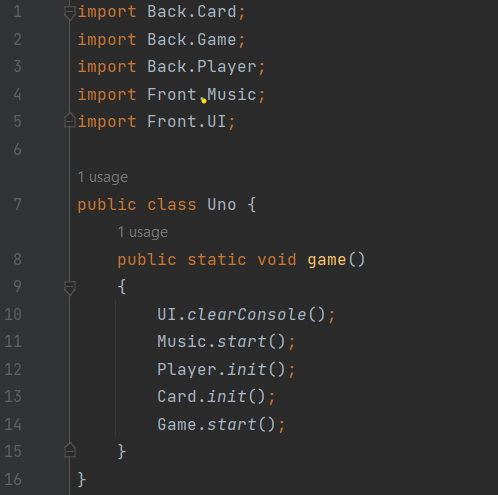
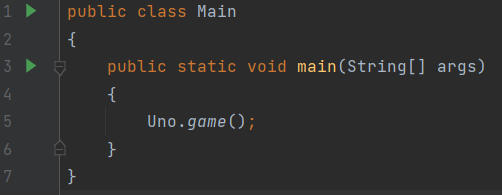
**System Summary :**

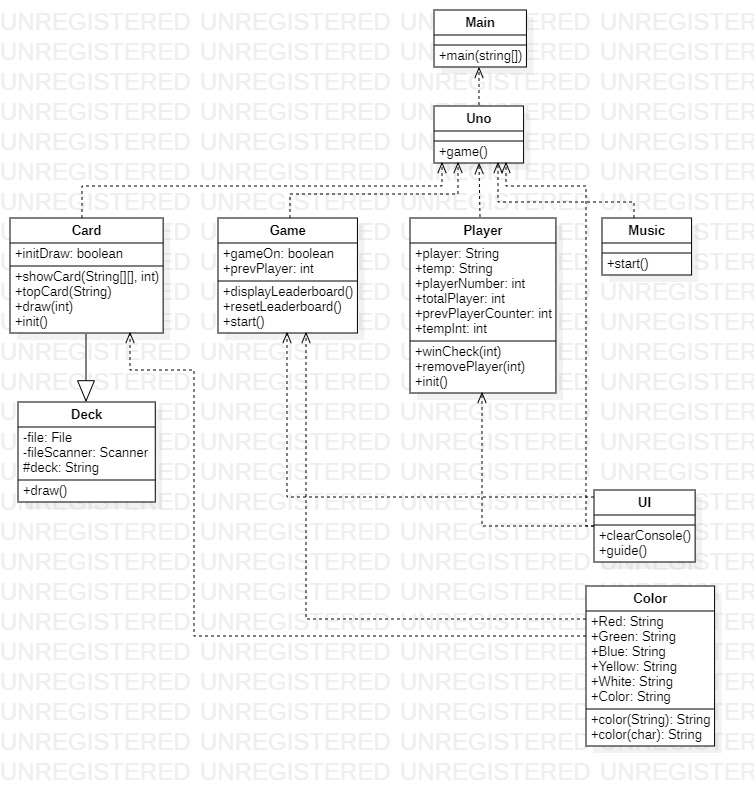
The UNO Card Game Project is a Java-based project that we created based on the creation of a Command Line Interface (CLI) application. This project is aimed at applying the object-oriented programming (OOP) principles we have learned.

The CLI interface allows players to interact with the game by entering commands such as “draw card”, “show card”, “next turn”, “withdraw”, “guide”, and “pick card”. The program then processes the input and updates the game state accordingly. The goal of this project is to create a functional and enjoyable UNO game experience using the CLI interface.

Using OOP principles, the UNO Card Game Project aims to make the development process easier and more organized. The project breaks games into smaller, more manageable components, making them easier to debug and maintain. Overall, the UNO Card Game Project demonstrates the power and versatility of OOP in creating complex and engaging applications.

**SYSTEM ANALYSIS**





**CLASS DIAGRAM**

**System Process:**

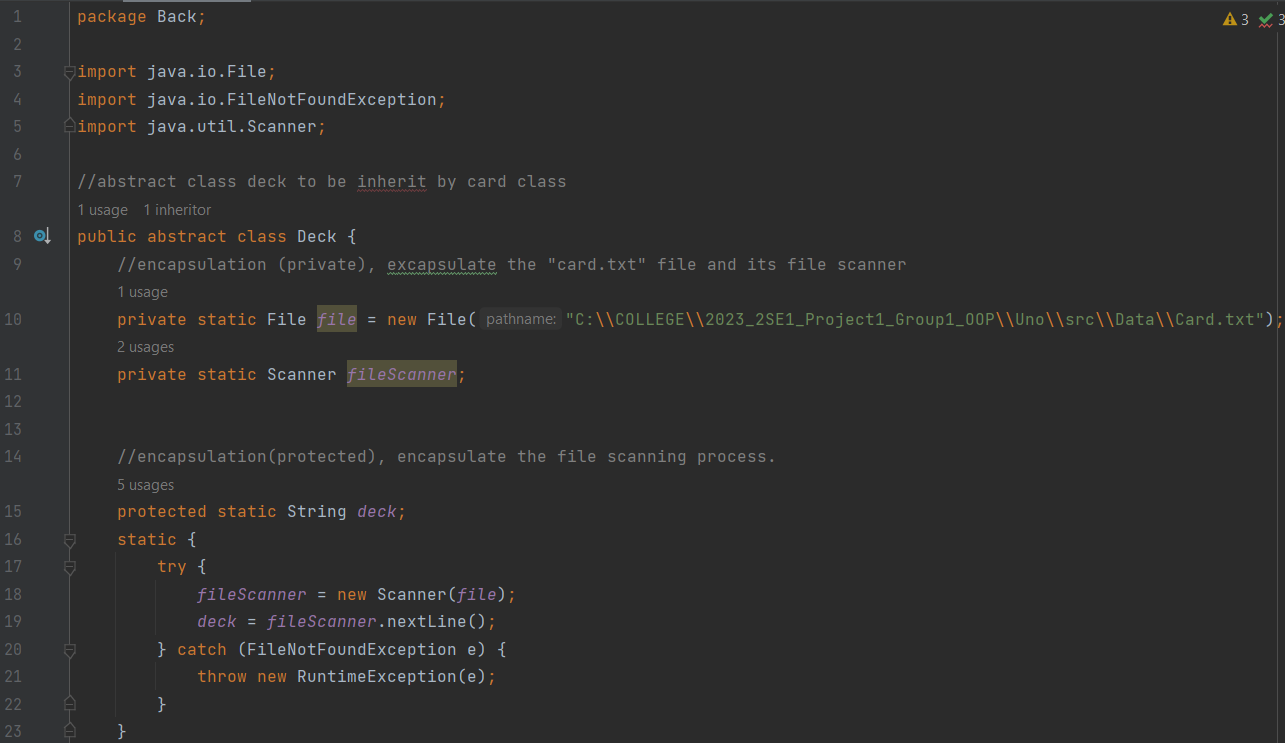
1. Main Class

The "Main" class only has the "main" method which is used to run the program. This method calls the "game" method of the "Uno" class to start the game.

**IMPLEMENTATION**

**IMPLEMENTATION**

1. Uno Class
2. Call the "clearConsole" method of the "UI" class to clear the console before the game starts.
3. Call the "start" method of the "Music" class to play background music during gameplay.
4. Call the "init" method of the "Player" class to initialize the cards for each player.
5. Call the "init" method of the "Card" class to initialize the card deck.
6. Call the "start" method of the "Game" class to start the game.



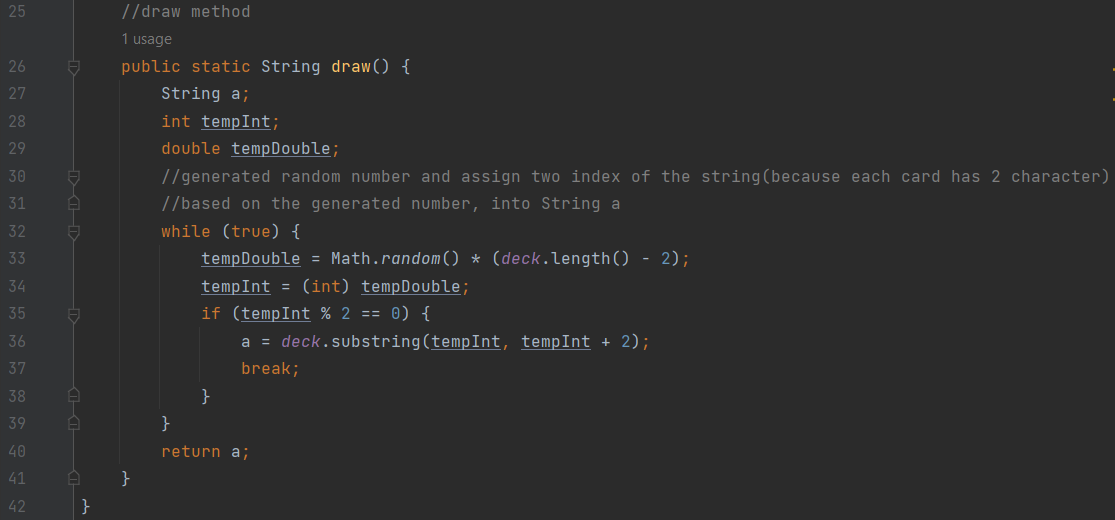
**System Process:**

1. Deck Class

Class "Deck" is an abstract class that is used as the parent class for class "Card". This class has several features such as:

1. Encapsulation (private) to encapsulate the "Card.txt" file and scan the file.
2. Encapsulation (protected) to encapsulate the file scanning process.
3. The static variable "deck" is used to store values from the "Card.txt" file.
4. The static block is used to read the "Card.txt" file and store its value in the "deck" variable.

**IMPLEMENTATION**

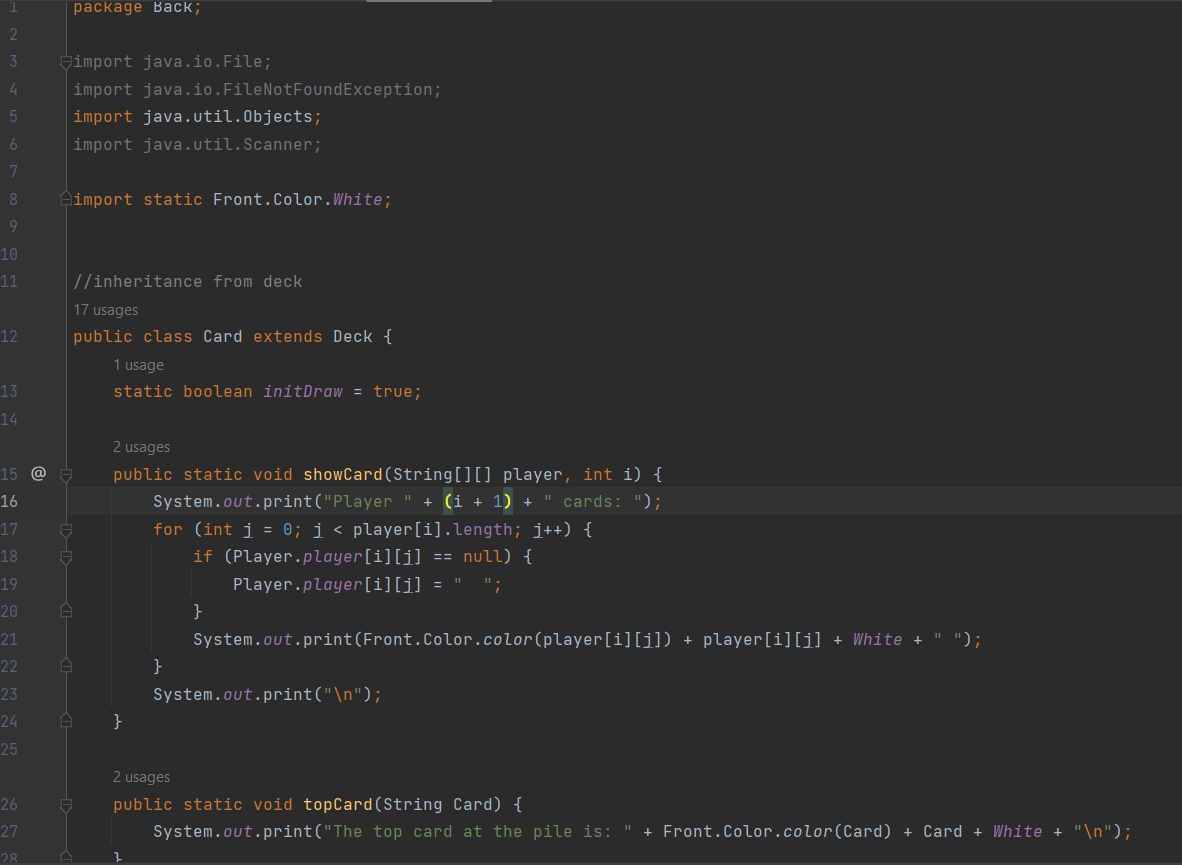


**IMPLEMENTATION**

1. Draw method is to draw a random card from the deck and return the value of that card .

The general function of the "draw" method in the "Deck" class is to draw a random card from the deck and return the value of that card. This method is used by derived classes to take cards from the deck and give them to the player. By using the "draw" method, each player will get a different card randomly from the deck. The "draw" method in class "Deck" is an abstract method that must be implemented by derived classes.

The code above is class "Deck" which is an abstract class used as the parent class for class "Card". The "Deck" class has several methods and variables that are used to arrange the card decks used in the Uno game.

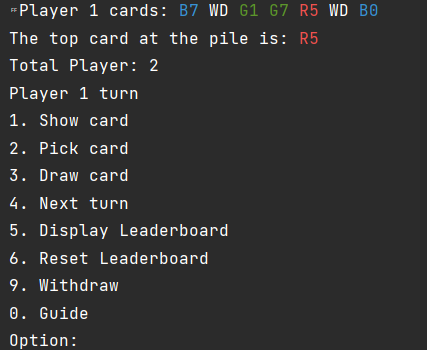
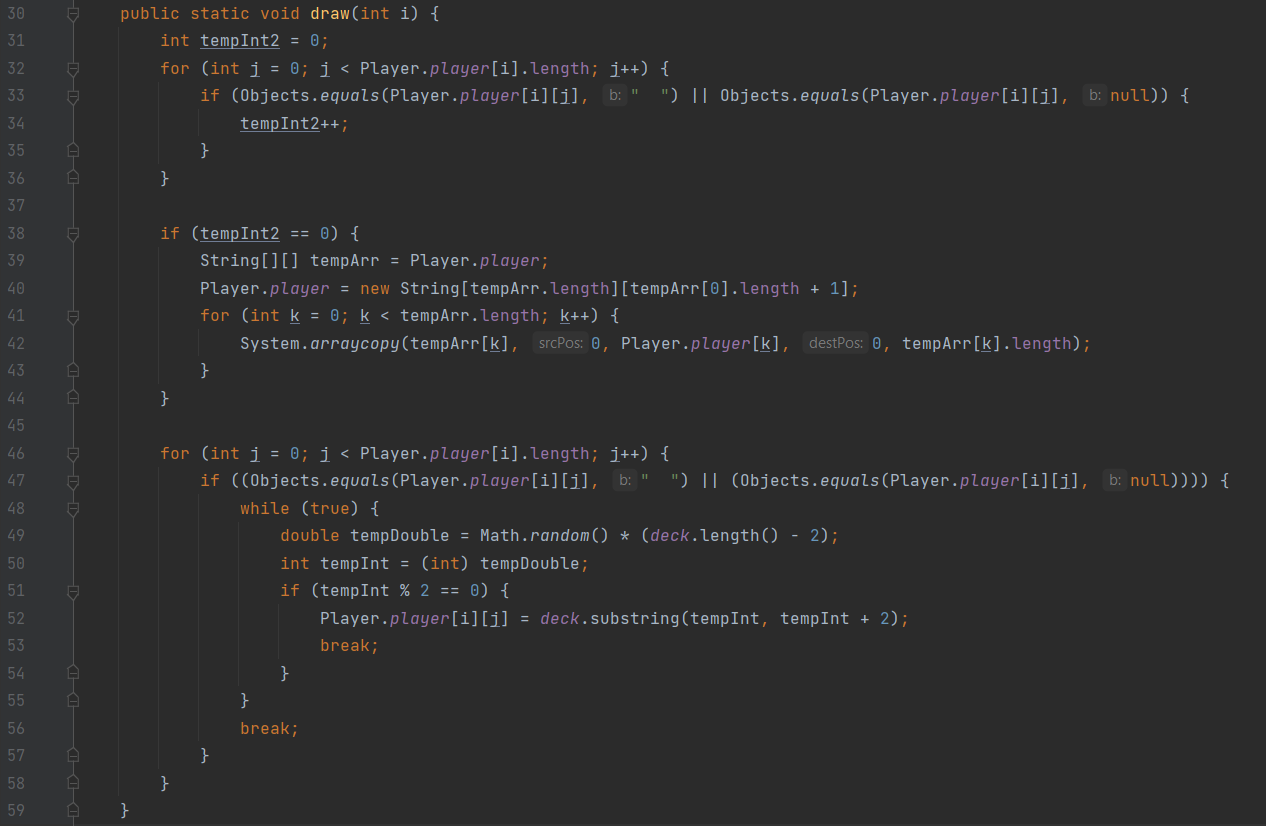


**System Process:**

1. Card Class

Class "Card" is a subclass of class "Deck". This class has several methods, namely:

1. The "showCard" method is used to display the cards owned by a player. This method accepts a parameter in the form of a 2-dimensional array "player" which represents the cards owned by each player, and the player number "i". This method will display the player's number and the cards it has on the screen. This showCard method later use to run the “Show Card” Menu in the program.
2. The "topCard" method is used to display the top card in the deck. This method accepts a parameter in the form of the string "Card" which represents the top card in the deck. This method will display the top card in the deck onto the screen with the corresponding color.

Both of these methods are used to display information about the cards owned by the player and the top card in the deck onto the screen



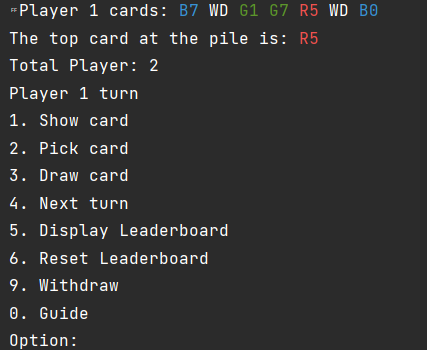
1. The “draw” used to randomly pick a card from the deck and give it to the player. This method accepts a parameter in the form of player number "i".
2. The "topCard" method is used to display the top card in the deck. This method accepts a parameter in the form of the string "Card" which represents the top card in the deck. This method will display the top card in the deck onto the screen with the corresponding color.

**IMPLEMENTATION**

In the "draw" method, first check whether the player still has an empty slot to accommodate a new card. If there are no free slots, then the "player" array will be enlarged by adding one new column.

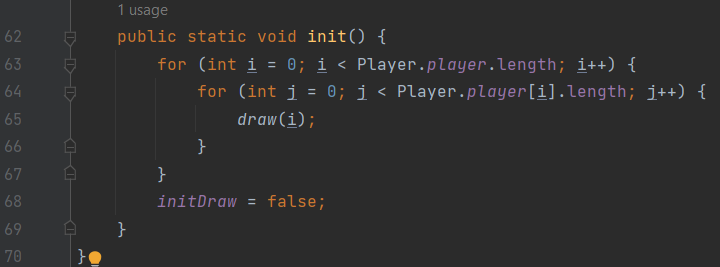
After that, it is repeated to check for empty slots in the "player" array. If an empty slot is found, a random number is generated using the "Math.random()" method. These numbers are then used to select two characters from the deck that represent the cards. The number must be even because each card has two characters. If the number is odd, it will be regenerated to produce an even number. After that, the two selected characters are combined into a string and stored in an empty slot in the "player" array.

The "draw" method in the "Card" class is used to draw a card from the deck and give it to the player. This method is used in games when players have to take cards from the deck because they don't have cards that can be played.



Later this draw method will function to run menu number 3 namely "Draw Card"

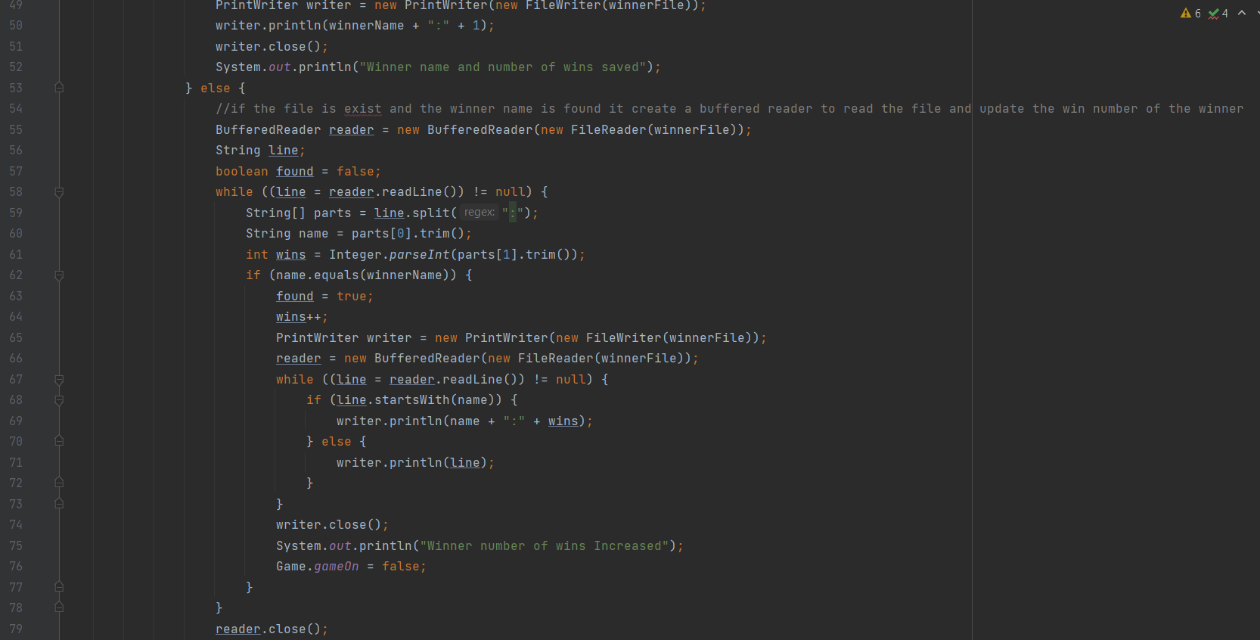
**IMPLEMENTATION**



1. The "init" method is used to initialize the cards for each player. This method calls the "draw" method for each player and each slot in the "player" array to draw a random card from the deck and add it to the player's hand.

Class "Card" is used to organize the cards in the Uno game. This class has several methods that are used to display information about the cards owned by the player and the top card in the deck, as well as take a card from the deck and give it to the player.

**IMPLEMENTATION**



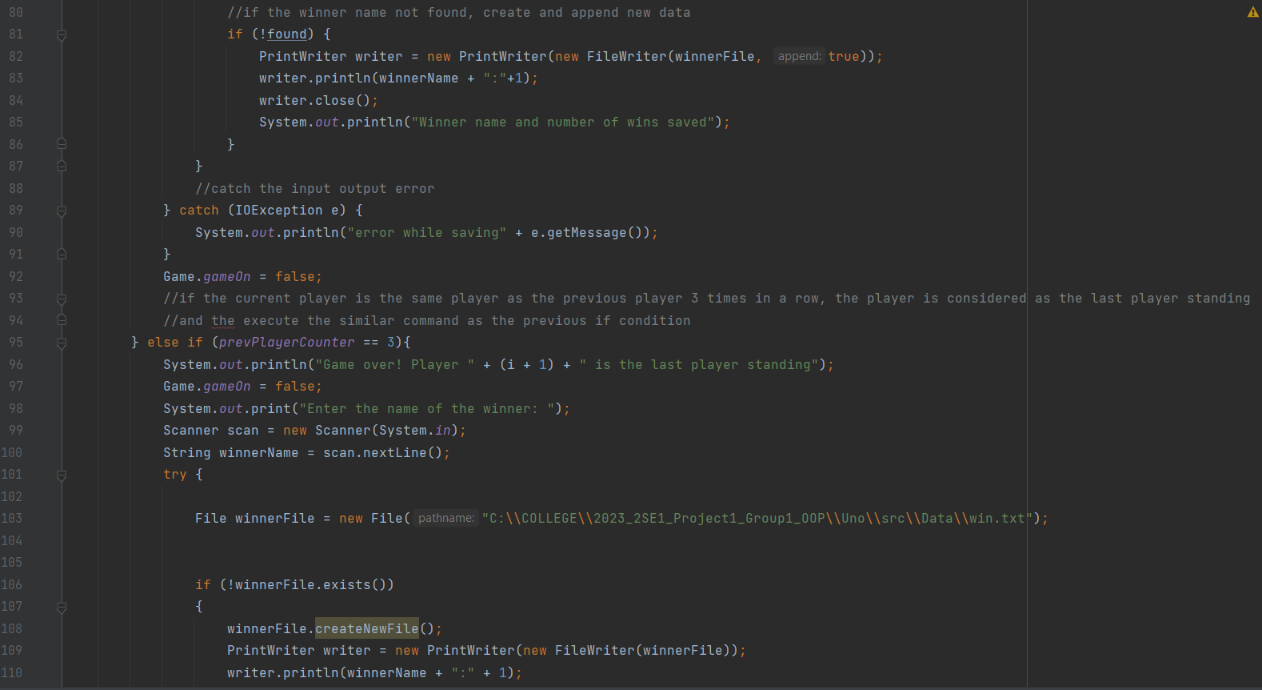
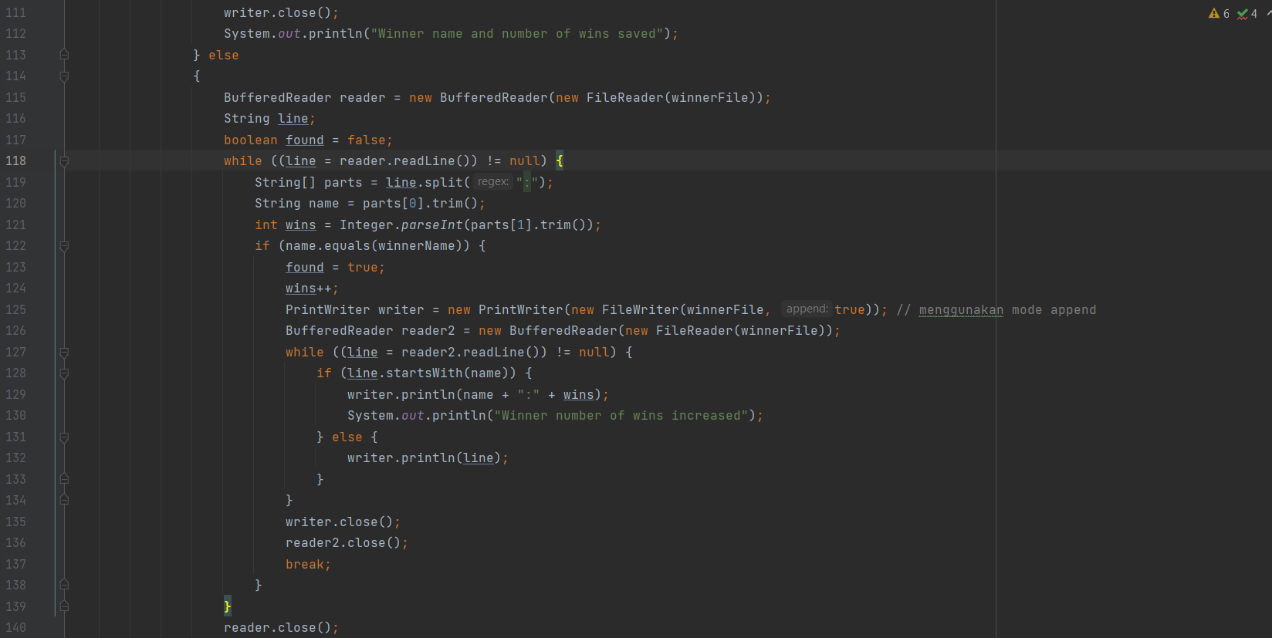
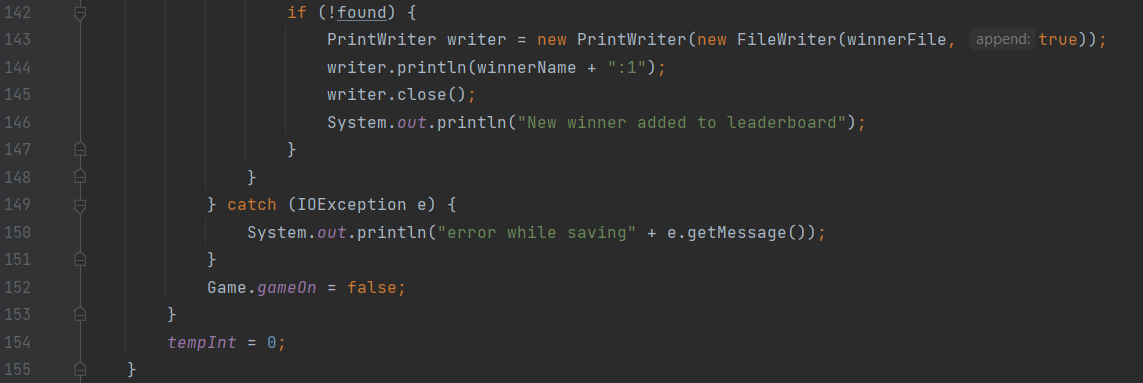
**IMPLEMENTATION**

**System Process:**

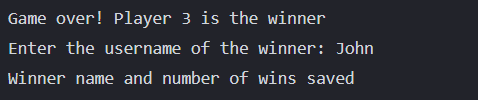
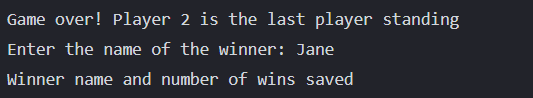
1. Player Class

The "Player" class is used to manage the players in the Uno game. This class has several methods, namely:

1. The "winCheck" method is used to check whether a player has won the game. This method accepts a parameter in the form of player number "i".



**IMPLEMENTATION**



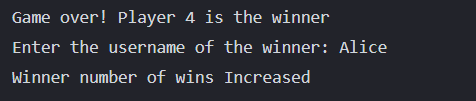
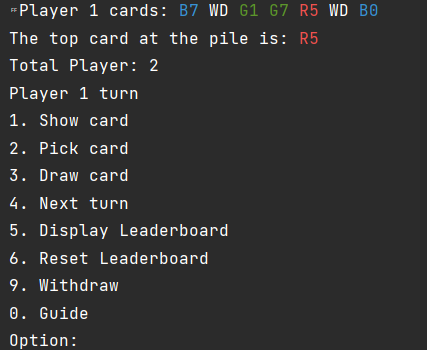
The "winCheck" method is used to check whether a player has won the game or is the last player standing. Below will explain the winCheck method in general with visualization of the points:

* At the start of the method, the "tempInt" variable is reset to 0.
* This method will iterate for every card owned by the player with index number "i".
* If the card is empty or null, the "tempInt" variable will be incremented by 1.
* After the loop is finished, the method will compare the value of the "tempInt" variable with the maximum number of cards that a player can have.
* If the value of the "tempInt" variable is equal to the maximum number of cards, then that player is considered the winner and the method will display the message "Game over! Player (i + 1) is the winner" followed by asking for the winner's name input and saving it into the "win" file. .txt".
* If the same player is the last player for 3 rounds, then that player is considered the last player standing and the method will display the message "Game over! Player (i + 1) is the last player standing" followed by prompting for the winner's name input and saving it to in the "win.txt" file.
* After that, the method will change the value of the "gameOn" variable in the "Game" class to false to stop the game.
* If no player meets the criteria to win the game, the method will return the "tempInt" variable to 0.

The following is an example of the output that the "winCheck" method produce:

1. Output when a player becomes the winner:
2. Output when a player becomes "last player standing":

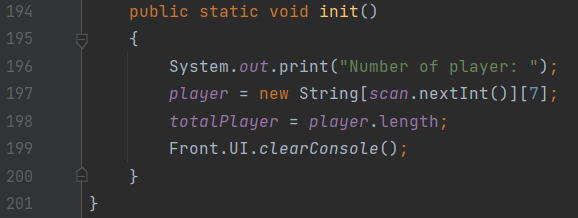
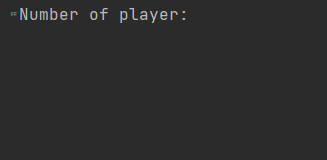
**IMPLEMENTATION**



1. Output when the winner's name is already in the "win.txt" file:
2. Output when an error occurred while saving data to the "win.txt" file:
3. The "removePlayer" method is used to remove a player from the game. This method accepts a parameter in the form of the index of the player you want to delete. This method will remove all cards owned by the player and reduce the value of the variable "totalPlayer" if the player has no cards in hand.

This method is used to run the "Withdraw" menu function in the program

**IMPLEMENTATION**



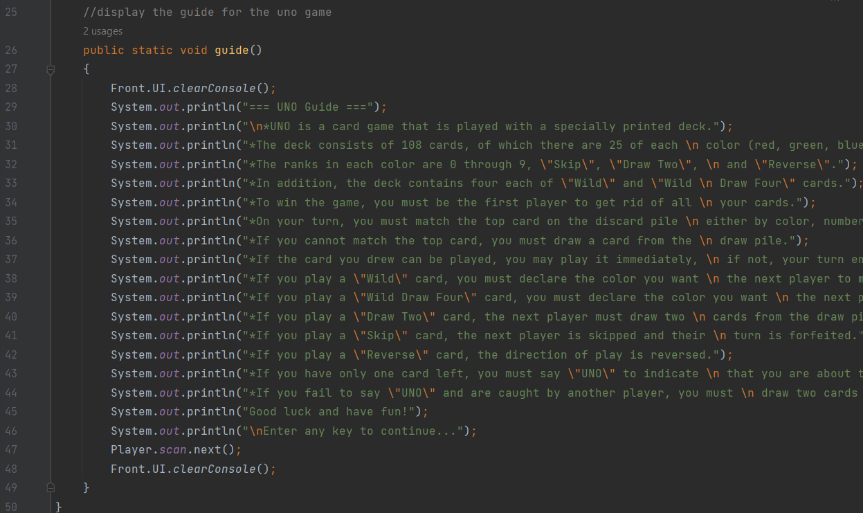
1. The "init" method is used to initialize the players in the game. This method will ask for the input of the number of players and create a 2-dimensional array "player" with the number of rows according to the number of players and the number of columns of 7. After that, the method will store the number of players in the variable "totalPlayer".

Previously there was also an init method for the card class, but the init method for the player class was different. The "init" method of the "Player" class will be executed first when the program starts. This is because the "init" method of the "Player" class is called first in the "main" method of the "Front" class. The "init" method in the "Player" class is used to initialize the number of players and the two-dimensional array that stores the cards owned by the players in the Uno game.

After the "init" method in the "Player" class has been executed, then the "init" method in the "Card" class is executed to initialize the cards owned by each player in the Uno game. The "init" method in the "Card" class will be called from within the "startGame" method in the "Front" class after the "init" method in the "Player" class has finished executing.

The "Player" class is used to manage the players in the Uno game. This class has several methods that are used to check whether a player has won a game, remove a player from the game, and initialize the players in the game.

**IMPLEMENTATION**

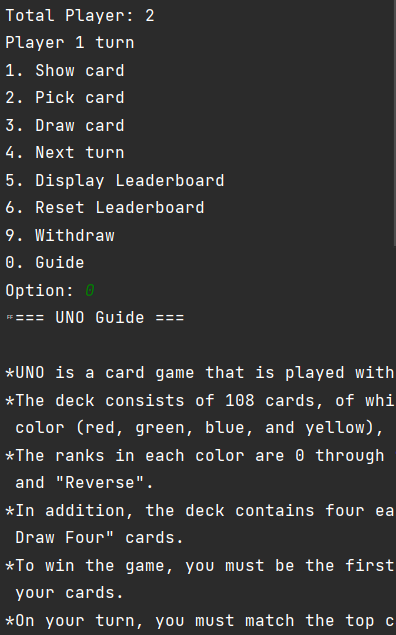
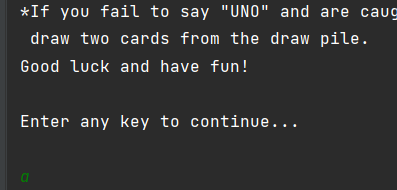


**System Process:**

1. UI Class

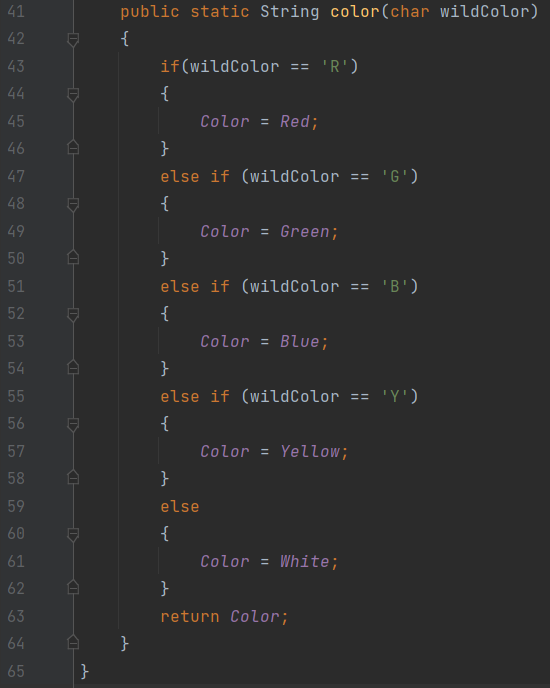
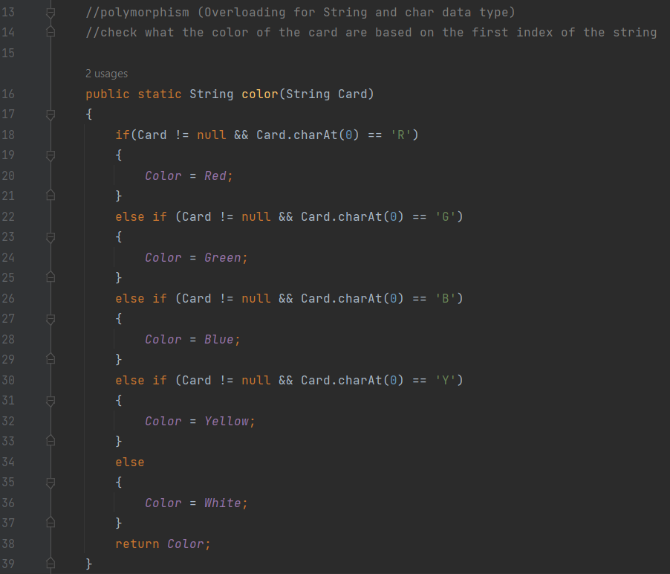
The code above is the "UI" class that is used to display the game's appearance and perform input from the user. Class "UI" has two methods namely "clearConsole" and "guide".

**IMPLEMENTATION**



1. The "clearConsole" method is used to clear the console display by running the "cls" command on Windows or "clear" on other operating systems. This method uses the "ProcessBuilder" class to execute commands on the console
2. The "guide" method is used to display the Uno game guide. This method will display information about card decks, how to win the game, game rules, and so on. After displaying the wizard, this method will wait for input from the user before clearing the console view.

**IMPLEMENTATION**

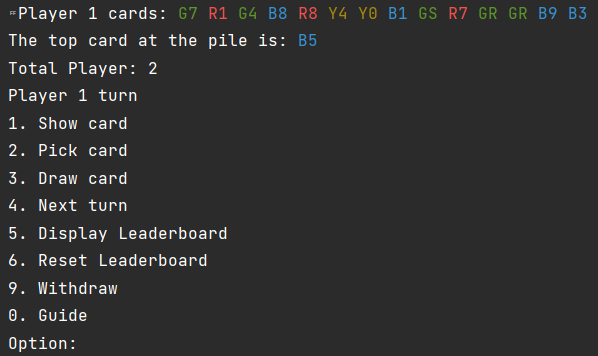


**System Process:**

1. Color Class

The code below is the "Color" class which is used to set the color of the text on the game display. Class "Color" has two methods namely "color" which is overloaded to accept parameters in the form of strings or characters.

**IMPLEMENTATION**



The following is a brief description of each method in the "Color" class:

1. The "Red" variable is used to store the red color code.
2. The "Green" variable is used to store the green color code.
3. The "Blue" variable is used to store the blue color code.
4. The "Yellow" variable is used to store the yellow color code.
5. The "White" variable is used to store the white color code.
6. The "Color" variable is used to store the color code to be used in the text.

The "color" method which accepts a string parameter is used to determine the color code to be used for text based on the first character in the string. If the first character is 'R', the color code red will be used. If the first character is 'G', the green color code will be used. If the first character is 'B', then the blue color code will be used. If the first character is 'Y' then the yellow color code will be used. If no characters match, the white color code is used. This method will return the appropriate color code.

**IMPLEMENTATION**



**System Process:**

1. Music Class:

The code above is the "Music" class which is used to play background music in the Uno game. The "Music" class has one method which is "start".

The following is a brief description of the "start" method in the "Music" class:

1. The "file" variable is used to store the music file to be played.
2. The "audioInputStream" variable is used to read the music file to be played.
3. The "clip" variable is used to play the read music file.
4. The try-catch block is used to read music files and play them using the "AudioSystem" and "Clip" classes. After the music file is played, this method will play the music continuously using the "loop" method with the parameter "Clip.LOOP\_CONTINUOUSLY". If an error occurs while reading or playing a music file, a stack trace of the error will be displayed.

**IMPLEMENTATION**

**IMPLEMENTATION**

**System Process:**

1. Game Class:

The "Game" class is used to determine the rules of the uno card game. The "Game" class has several methods namely:

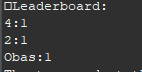
1. The “displayLeaderboard” method used to display the leaderboard of the Uno game, which is stored in the win.txt file.



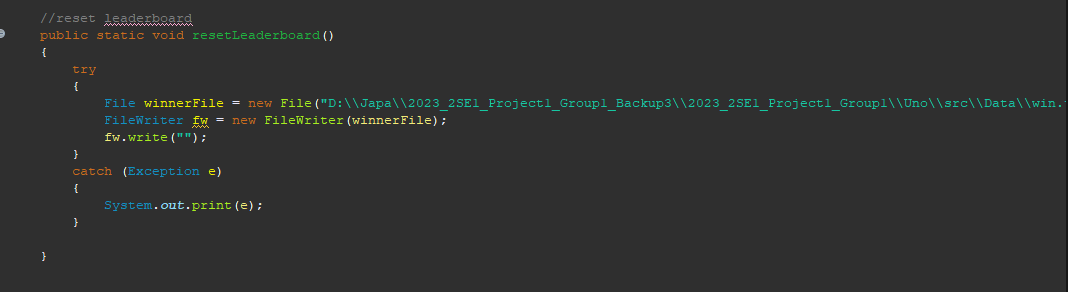
Below will explain the displayLeaderboard method in general with visualization of the points:

* At the beginning of the method, we check whether the win.txt file exists or not.
* If the file does not exist, the method will print "Leaderboard is empty"
* if the file exists, then the method will read each line in the file using BufferedReader, then insert each line into the leaderboard list.
* The method will check whether the leaderboard list is empty or not.
* If empty, the method will print "Leaderboard is empty".
* if it is not empty, then the leaderboard list will be sorted based on the number of wins for each player.
* After the leaderboard list is sorted, the method prints "Leaderboard:" followed by each line in the leaderboard list, which contains the player's name and the number of wins player has won.
* If an error occurs while reading the file, the method will print "error while reading".

The following is an example of the output that the "displayLeaderboard" method produce:



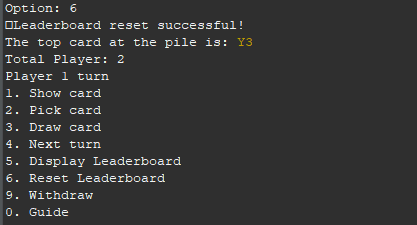
1. The “resetLeaderboard” method used to reset the leaderboard of the Uno game, which is stored in the win.txt file.



Below will explain the resetLeaderboard method in general with visualization of the points:

* At the beginning of the method, the win.txt file is retrieved.
* After that, the method will create a FileWriter object with the winnerFile file parameter.
* By writing an empty string, the entire contents of the win.txt file will be deleted.

The following is an example of the output that the "resetLeaderboard" method produce:

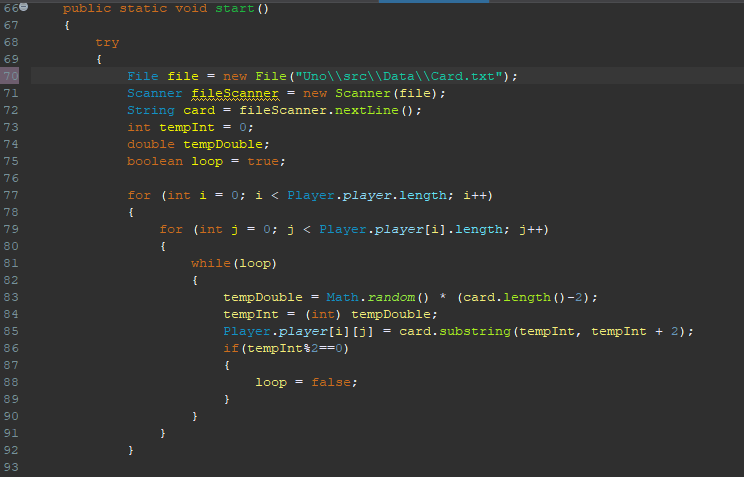


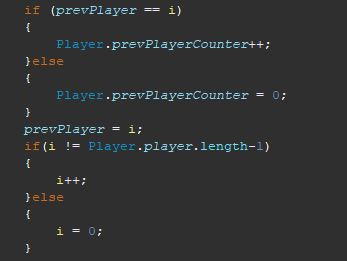
**IMPLEMENTATION**

**IMPLEMENTATION**

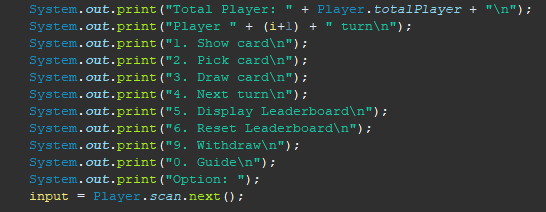
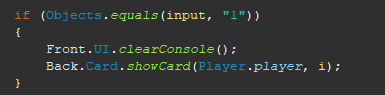
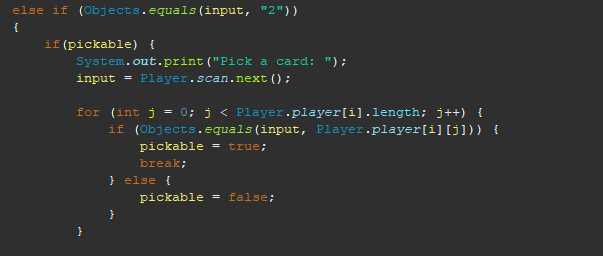
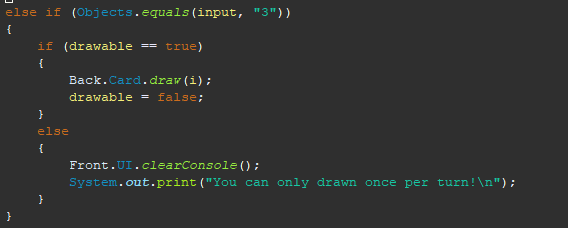
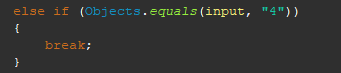
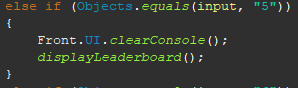
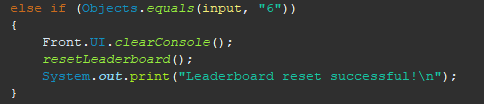
1. The “Start” method is used to run several rules and function in the uno card game. the rule and function consists of :

* Function for start game and distribute cards from the Card.txt file



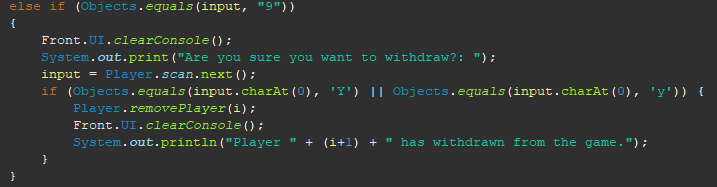
* Function to determine the next player / next turn

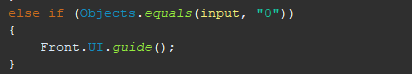
**IMPLEMENTATION**

* Function to display a menu that can be selected by the player
* Function to display the player's card if the player chooses the show card menu
* Function to pick the player's card if the player chooses the Pick card menu
* Function to draw the player's card if the player chooses the Draw card menu****
* Function to next turn when the player chooses the Next turn menu****
* Function to display the game leaderboard by calling “displayLeaderboard” method if the player chooses the display leaderboard menu****
* Function to reset the game leaderboard by calling “resetLeaderboard” method if the player chooses the reset leaderboard menu****

**IMPLEMENTATION**

* Function to withdraw the player if the player chooses the Withdraw menu

****

* Function to display the guide of the game by calling guide method from UI class if the player chooses the guide menu****

**IMPLEMENTATION**

**Hardware :** Intel® Core™ i5-10300H, 8GB RAM/AMD Ryzen 5 4600H 8GB RAM

**Operating System :** Windows 11

**Software :** Eclipse

|  |  |  |
| --- | --- | --- |
| **PROJECT FILE DETAILS** | | |
| **No** | **File Name** | **Remarks** |
| 1. | 2023\_2SE1\_Project1\_Group 1.zip | Zip file containing packages of the UNO Card Game project(Java, Class, txt, mp3, and WAV) |

**CONFIGURATION**