



DELIVERABLE 2

CARD GAME: WAR

Table of Contents

Overview	2
1. Project Background and Description	2
2. Design Considerations	3
Associations:	4
Multiplicities:	4
Design Choices:	4
Game Rules:.....	5

SYST 17796 DELIVERABLE 2

DESIGN DOCUMENT TEMPLATE

OVERVIEW

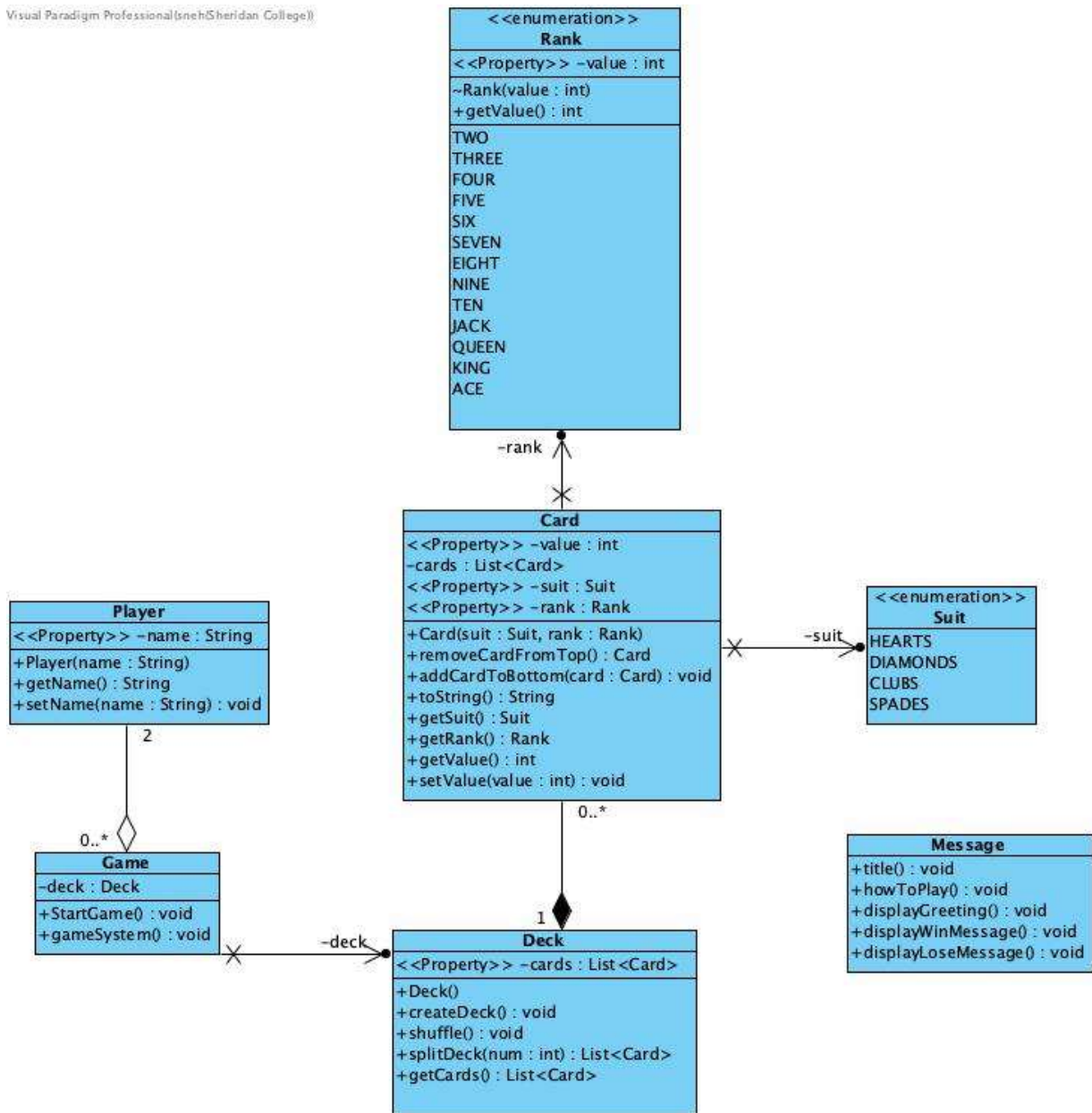
1. Project Background and Description

To recap, the card game that our team has chosen is called War. The interface of the game starts by displaying the title and then it proceeds to prompt the user whether they would like to play the game. If yes, the user is then prompted for a username. After the username input, the user is notified that the deck has been assigned to them, and the CPU, which insinuates that the deck was evenly split into two. Otherwise, the player is greeted with a goodbye message. Thus far, we can see the cards for both players and the game ends.

Analyzing the code that we currently have, a handful of the classes from the ProjectStarterCode folder were carried over and built on top of. Classes such as Deck, Rank, Suit, Main and Message are new additions. The mentioned classes allow us to create suitable methods and variables that do not interrupt other classes.

2. Design Considerations

Visual Paradigm Professional(snehlSheridan College)



Associations:

Game and Player:

- **Type of Association:** Aggregation
- **Description:** A Game involves two Player objects. The players can exist independently of the game.

Deck and Card:

- **Type of Association:** Composition
- **Description:** A Deck is composed of multiple Card objects. If the deck is destroyed, the cards within it are also destroyed.

Multiplicities:

- **From Game to Player:** 1 Game involves exactly 2 Players (2)
- **From Player to Game:** A Player can be involved in zero or more Games (0..*)
- **From Deck to Card:** 1 Deck contains zero or more Cards (0..*)
- **From Card to Deck:** Each Card belongs to exactly one Deck (1)

Design Choices:

- Card class has **low coupling** because it only depends on the Suit and Rank enums, and it does not rely on other classes. The class also demonstrates **high cohesion** because it encapsulates all the necessary variables and methods related to cards.
- Deck class illustrates **composition** by including a list of Card objects as an instance variable. In addition, the class includes methods that allow to perform actions such as shuffling, splitting the deck and retrieving a list of cards.
- Game class uses **aggregation** to create a relationship between the Game and Deck classes. In the gameSystem() method, an instance of the Deck class is created and assigned to the deck variable of the Game class therefore permitting access to methods and data of the Deck class.
- Main class focuses on a single responsibility that is initiating the Game class hiding logic and further implementation details.
- Message class **abstracts** the details of how the messages are displayed. The code only needs to call the appropriate method to display the desired message. There are a variety of messages that can be used and easily added in the future.

- Player class exhibits **encapsulation** too for it includes constructors, getters and setters for Player's name.
- Rank and Suit class are both **enumeration** classes that hold a fixed set of predefined constants. Understanding enums can enhance the **readability and maintainability** of codes in projects.
 - Rank.java – The following class includes the thirteen ranks with their associated integer value. It also has a constructor that sets the value for each rank and a get method to retrieve that value.
 - Suit.java – The following class includes the four standard card suits: HEARTS, DIAMONDS, CLUBS, SPADES.

Game Rules:

- Deck Splitting
At the start of the game, the deck is split between the two players. Each player receives half of the deck, ensuring both players start with an equal number of cards.
- Comparing Cards
 1. In each round, both players draw the top card from their deck.
 2. The player with the higher-ranked card wins the round and takes both cards, adding them to the bottom of their deck.
 3. If the drawn cards have the same rank, a "war" occurs:
 - Each player draws one additional card.
 - The higher-ranked card of these additional cards determines the winner of the round.
 - If the additional cards also have the same rank, the war continues until a player wins the round.
- Deciding Winner
The game continues until one player has all the cards. The player with the most cards at the end of the game is declared the winner.