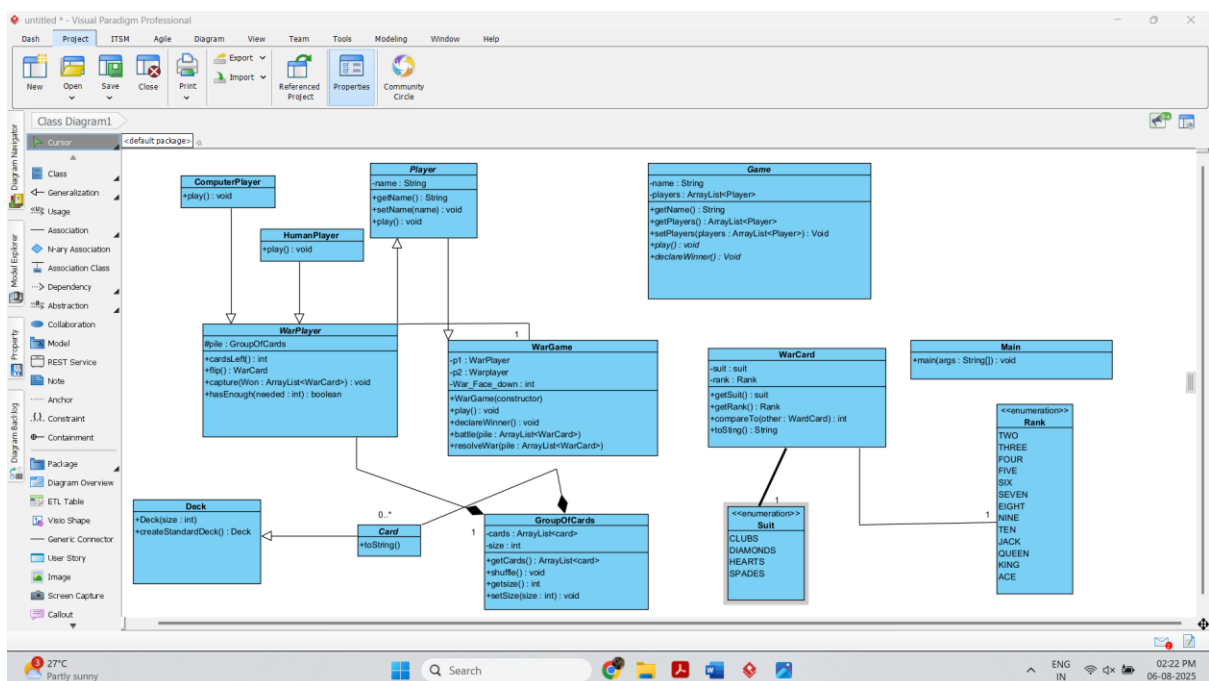


SYSTEM DELIVERABLE 3

1. Class Diagram

This updated class diagram reflects all feedback from Deliverable 2. All methods, attributes, associations, inheritance relationships, and enumerations are included. It is notationally correct and follows standard UML design conventions.



2. Source Code Implementation

The source code was implemented exactly according to the final class diagram. It is functionally correct and follows standard coding conventions including:

- Proper naming (CamelCase & camelCase)
- Meaningful method/class names
- Clear documentation/comments
- Proper indentation and formatting

Class	Description
Card	Base card with toString()
WarCard	Card with Rank & Suit
Deck	Creates and shuffles deck
GroupOfCards	Manages a collection of cards
Suit, Rank	Enumerations
Player	Abstract class for players
HumanPlayer	Concrete player
ComputerPlayer	Concrete player
WarPlayer	Extends Player, handles pile
Game	Generic interface
WarGame	Main game logic
Main	main() launcher

3. Source Code Design

Description:

The project follows Object-Oriented Design principles, as studied in the course:

- **Encapsulation:** Private fields with getters/setters.
- **Inheritance:** Player → WarPlayer → HumanPlayer / ComputerPlayer.
- **Composition:** WarPlayer uses GroupOfCards.
- **Delegation:** Methods in WarGame delegate actions to WarPlayer.
- **Cohesion & Coupling:** Classes are modular with high cohesion and low coupling.
- **Reusability & Flexibility:** Easily supports more players or game variations.

4. Game Playability:

The application is fully playable. A game is launched using the main() method and executes automatically, displaying:

- Player moves (card flips)
- War resolution when ties occur
- The winner when the game ends

Reference to Git Repository

The complete and runnable Java project, along with all source files, is available in the Git repository linked below:

https://github.com/tamurov1/War_Game-Project.git