Poker Tournament Manager

Name: Thomas Archambault

Course: DataStructuresAndOOP

Deliverable 1 - Project Description

Scenario

This application simulates and manages poker tournaments. Players can register, participate in games, and get ranked based on chip count or winnings. Organizers (admins) can configure blind structures, seat players at tables, and track player progress as the tournament advances. It includes tracking of eliminations, prize pools, and rankings.

Design Paradigm

Key functionalities include:
- Player Registration and Tournament Setup
- Blind structure configuration
- Seating and table management
- Player elimination and ranking logic
- Prize pool calculation
- Text file reports for tournament results
- Custom sorting and stream filtering

Expected Output

Users can:
- Register players with name and buy-in amount
- Start a tournament with blind structure
- View player status and real-time ranking
- Export results to text file

Hierarchies

1. User -> Admin, Player
2. TournamentComponent -> BlindStructure, PrizePool

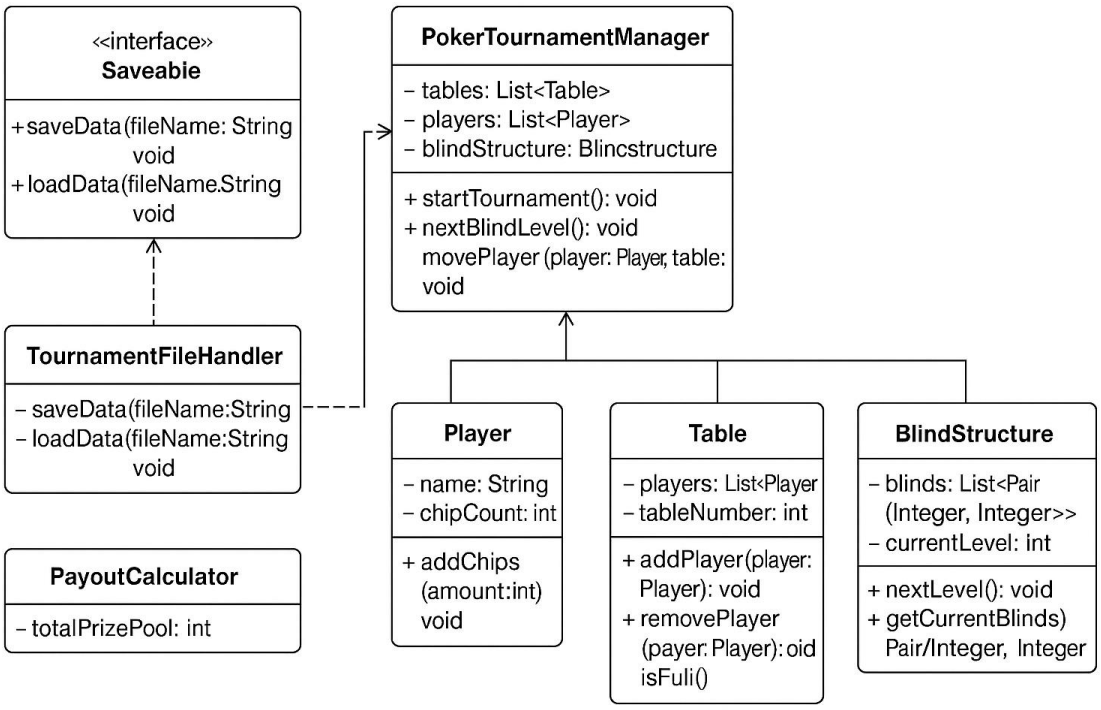
Interface
`Playable` interface is used for polymorphic execution of game rounds. Implemented in both `Player` and `Table` classes to simulate hands.

Polymorphism
Method `playHand()` is overridden in both `Player` and `Table`, allowing runtime polymorphism.

TextIO
`TournamentLogger` handles file writing/reading of player results and game history.

Comparable & Comparator
`Player` implements `Comparable` (by chip count). `PlayerNameComparator` implements `Comparator<Player>` to sort by name.

UML Diagram



Deliverable 2 Plan

- Implement classes: User, Admin, Player, Table, BlindStructure
- Interface: Playable
- Logger class and basic I/O
- JUnit test skeletons
- JavaDocs for all user-defined methods

Git Repository

- Maven project initialized
- .gitignore and README.md added
- `doc` folder includes this PDF