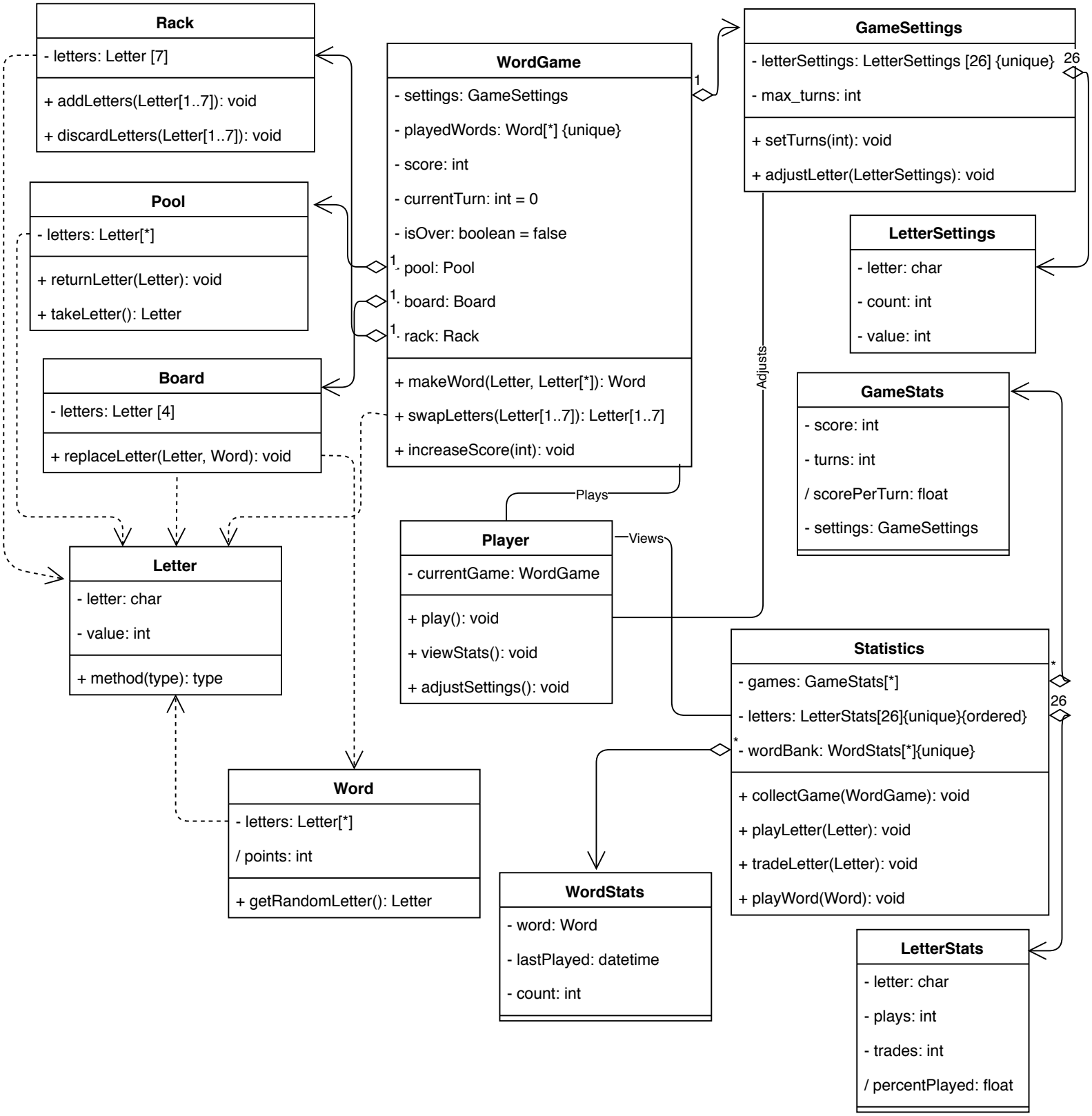
Rack
- letters: Letter [7]
+ addLetters(Letter[1..7]): void
+ discardLetters(Letter[1..7]): void
Pool
- letters: Letter[*]
+ returnLetter(Letter): void
+ takeLetter(): Letter
Board
- letters: Letter [4]
+ replaceLetter(Letter, Word): void
Letter
- letter: char
- value: int
+ method(type): type
Word
- letters: Letter[*]
/ points: int
+ getRandomLetter(): Letter
WordGame
- settings: GameSettings
- playedWords: Word[*] {unique}
- score: int
- currentTurn: int = 0
- isOver: boolean = false
1 - pool: Pool
1 - board: Board
1 - rack: Rack
+ makeWord(Letter, Letter[*]): Word
+ swapLetters(Letter[1..7]): Letter[1..7]
+ increaseScore(int): void
1
GameSettings
- letterSettings: LetterSettings [26] {unique}
- max_turns: int
+ setTurns(int): void
+ adjustLetter(LetterSettings): void
26
LetterSettings
- letter: char
- count: int
- value: int
Adjusts
GameStats
- score: int
- turns: int
/ scorePerTurn: float
- settings: GameSettings
Plays
Player
- currentGame: WordGame
+ play(): void
+ viewStats(): void
+ adjustSettings(): void
Views
Statistics
- games: GameStats[*]
- letters: LetterStats[26]{unique}{ordered}
* - wordBank: WordStats[*]{unique}
+ collectGame(WordGame): void
+ playLetter(Letter): void
+ tradeLetter(Letter): void
+ playWord(Word): void
*
26
WordStats
- word: Word
- lastPlayed: datetime
- count: int
LetterStats
- letter: char
- plays: int
- trades: int
/ percentPlayed: float