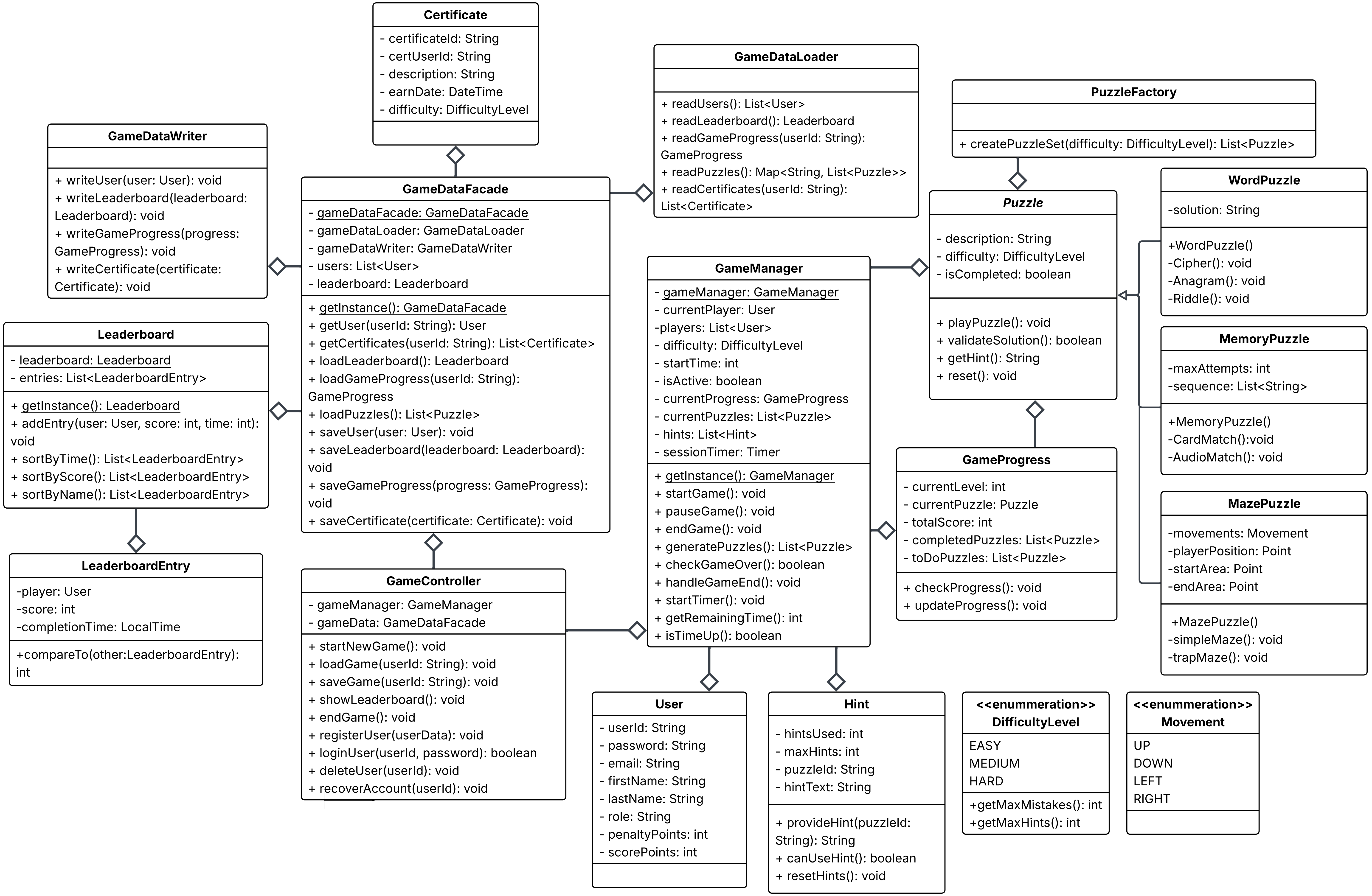
Certificate
- certificateId: String
- certUserId: String
- description: String
- earnDate: DateTime
- difficulty: DifficultyLevel
GameDataLoader
+ readUsers(): List<User>
+ readLeaderboard(): Leaderboard
+ readGameProgress(userId: String): GameProgress
+ readPuzzles(): Map<String, List<Puzzle>>
+ readCertificates(userId: String): List<Certificate>
PuzzleFactory
+ createPuzzleSet(difficulty: DifficultyLevel): List<Puzzle>
GameDataWriter
+ writeUser(user: User): void
+ writeLeaderboard(leaderboard: Leaderboard): void
+ writeGameProgress(progress: GameProgress): void
+ writeCertificate(certificate: Certificate): void
GameDataFacade
- gameDataFacade: GameDataFacade
- gameDataLoader: GameDataLoader
- gameDataWriter: GameDataWriter
- users: List<User>
- leaderboard: Leaderboard
+ getInstance(): GameDataFacade
+ getUser(userId: String): User
+ getCertificates(userId: String): List<Certificate>
+ loadLeaderboard(): Leaderboard
+ loadGameProgress(userId: String): GameProgress
+ loadPuzzles(): List<Puzzle>
+ saveUser(user: User): void
+ saveLeaderboard(leaderboard: Leaderboard): void
+ saveGameProgress(progress: GameProgress): void
+ saveCertificate(certificate: Certificate): void
Puzzle
- description: String
- difficulty: DifficultyLevel
- isCompleted: boolean
+ playPuzzle(): void
+ validateSolution(): boolean
+ getHint(): String
+ reset(): void
WordPuzzle
-solution: String
+WordPuzzle()
-Cipher(): void
-Anagram(): void
-Riddle(): void
MemoryPuzzle
-maxAttempts: int
-sequence: List<String>
+MemoryPuzzle()
-CardMatch():void
-AudioMatch(): void
MazePuzzle
-movements: Movement
-playerPosition: Point
-startArea: Point
-endArea: Point
+MazePuzzle()
-simpleMaze(): void
-trapMaze(): void
Leaderboard
- leaderboard: Leaderboard
- entries: List<LeaderboardEntry>
+ getInstance(): Leaderboard
+ addEntry(user: User, score: int, time: int): void
+ sortByTime(): List<LeaderboardEntry>
+ sortByScore(): List<LeaderboardEntry>
+ sortByName(): List<LeaderboardEntry>
LeaderboardEntry
-player: User
-score: int
-completionTime: LocalTime
+compareTo(other:LeaderboardEntry): int
GameManager
- gameManager: GameManager
- currentPlayer: User
-players: List<User>
- difficulty: DifficultyLevel
- startTime: int
- isActive: boolean
- currentProgress: GameProgress
- currentPuzzles: List<Puzzle>
- hints: List<Hint>
- sessionTimer: Timer
+ getInstance(): GameManager
+ startGame(): void
+ pauseGame(): void
+ endGame(): void
+ generatePuzzles(): List<Puzzle>
+ checkGameOver(): boolean
+ handleGameEnd(): void
+ startTimer(): void
+ getRemainingTime(): int
+ isTimeUp(): boolean
GameProgress
- currentLevel: int
- currentPuzzle: Puzzle
- totalScore: int
- completedPuzzles: List<Puzzle>
- toDoPuzzles: List<Puzzle>
+ checkProgress(): void
+ updateProgress(): void
GameController
- gameManager: GameManager
- gameData: GameDataFacade
+ startNewGame(): void
+ loadGame(userId: String): void
+ saveGame(userId: String): void
+ showLeaderboard(): void
+ endGame(): void
+ registerUser(userData): void
+ loginUser(userId, password): boolean
+ deleteUser(userId): void
+ recoverAccount(userId): void
User
- userId: String
- password: String
- email: String
- firstName: String
- lastName: String
- role: String
- penaltyPoints: int
- scorePoints: int
Hint
- hintsUsed: int
- maxHints: int
- puzzleId: String
- hintText: String
+ provideHint(puzzleId: String): String
+ canUseHint(): boolean
+ resetHints(): void
<<enummeration>>
DifficultyLevel
EASY
MEDIUM
HARD
+getMaxMistakes(): int
+getMaxHints(): int
<<enummeration>>
Movement
UP
DOWN
LEFT
RIGHT