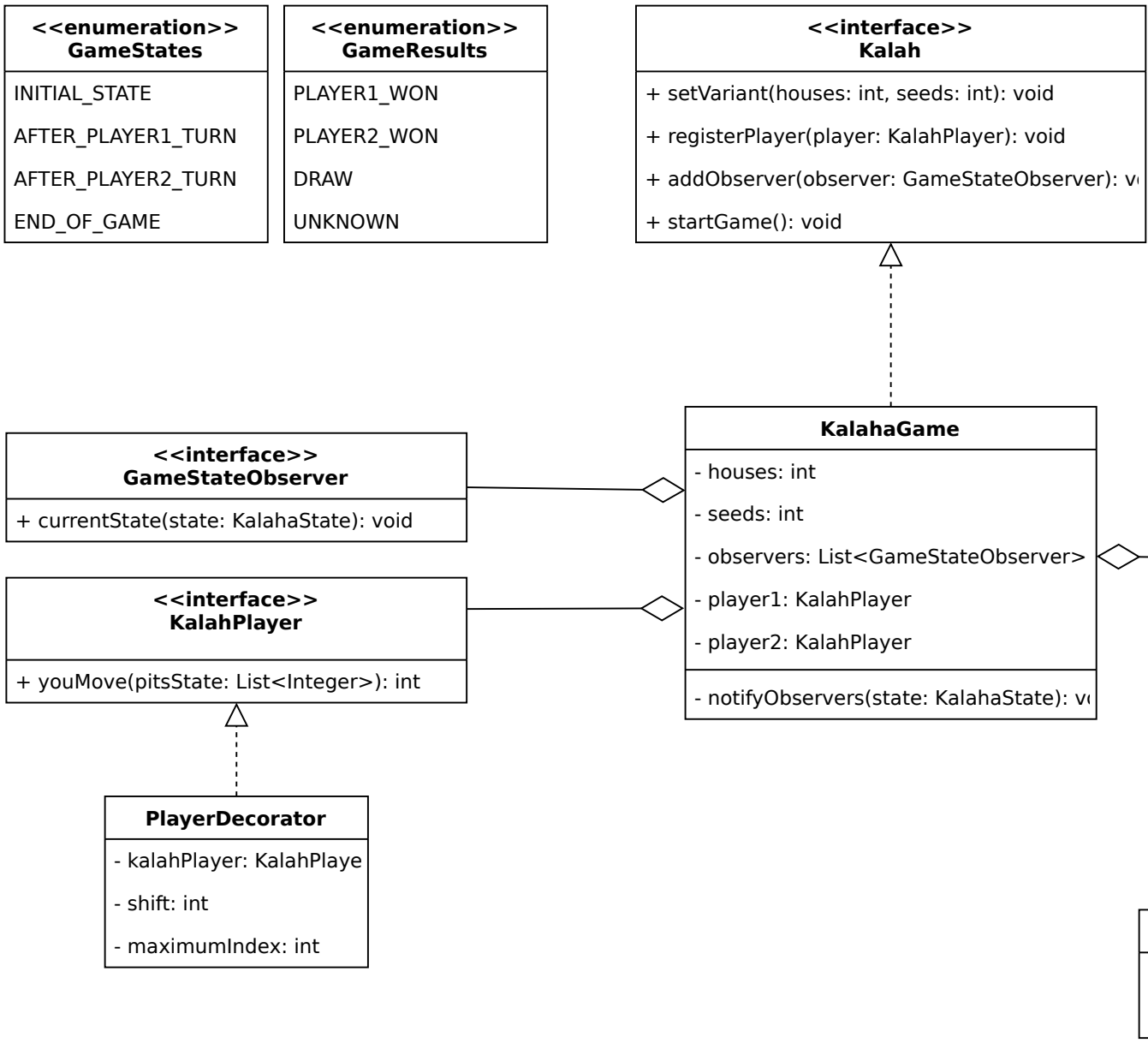
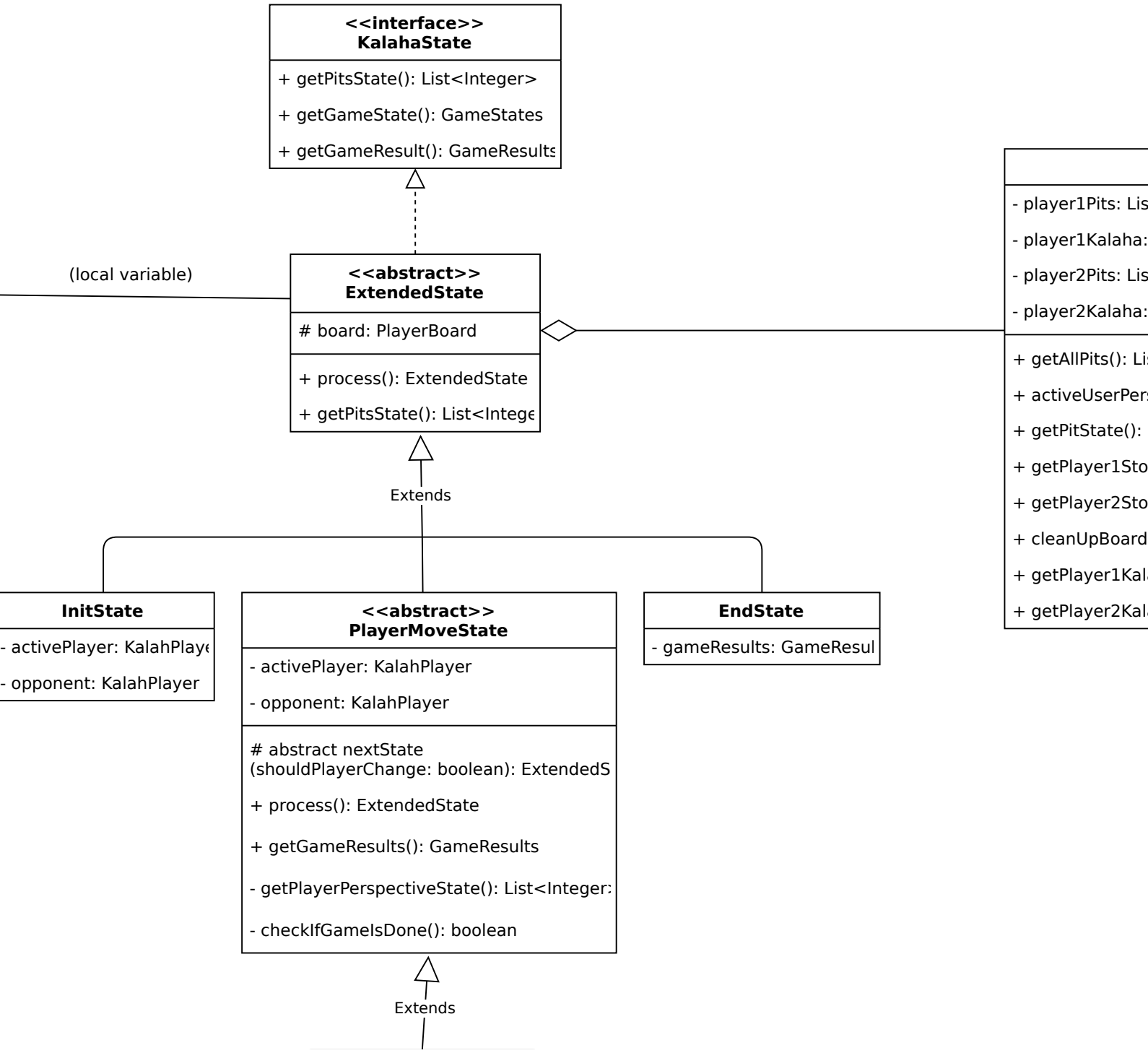
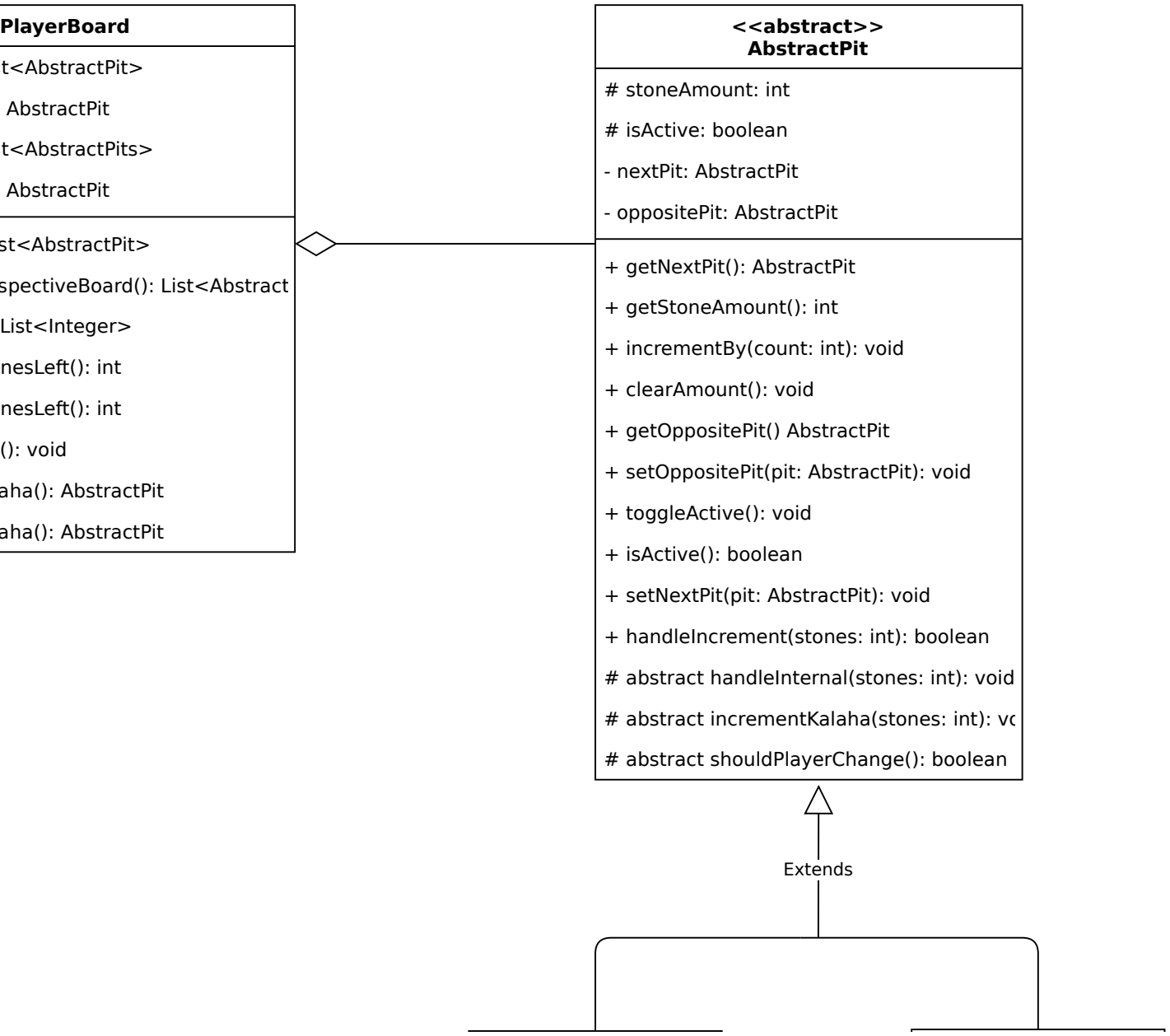
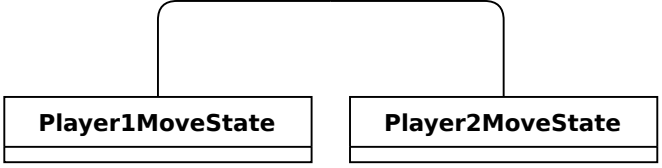

<<enumeration>>
GameStates
INITIAL_STATE
AFTER_PLAYER1_TURN
AFTER_PLAYER2_TURN
END_OF_GAME
<<enumeration>>
GameResults
PLAYER1_WON
PLAYER2_WON
DRAW
UNKNOWN
<<interface>>
Kalah
+ setVariant(houses: int, seeds: int): void
+ registerPlayer(player: KalahPlayer): void
+ addObserver(observer: GameStateObserver): v
+ startGame(): void
<<interface>>
GameStateObserver
+ currentState(state: KalahaState): void
<<interface>>
KalahPlayer
+ youMove(pitsState: List<Integer>): int
KalahaGame
- houses: int
- seeds: int
- observers: List<GameStateObserver>
- player1: KalahPlayer
- player2: KalahPlayer
- notifyObservers(state: KalahaState): v
PlayerDecorator
- kalahPlayer: KalahPlaye
- shift: int
- maximumIndex: int


<<interface>>
KalahaState
+ getPitsState(): List<Integer>
+ getGameState(): GameStates
+ getGameResult(): GameResults
(local variable)
<<abstract>>
ExtendedState
board: PlayerBoard
+ process(): ExtendedState
+ getPitsState(): List<Intege
- player1Pits: Lis
- player1Kalaha:
- player2Pits: Lis
- player2Kalaha:
+ getAllPits(): Li
+ activeUserPer
+ getPitState():
+ getPlayer1Sto
+ getPlayer2Sto
+ cleanUpBoard
+ getPlayer1Kal
+ getPlayer2Kal
Extends
InitState
- activePlayer: KalahPlaye
- opponent: KalahPlayer
<<abstract>>
PlayerMoveState
- activePlayer: KalahPlayer
- opponent: KalahPlayer
abstract nextState
(shouldPlayerChange: boolean): ExtendedS
+ process(): ExtendedState
+ getGameResults(): GameResults
- getPlayerPerspectiveState(): List<Integer:
- checkIfGameIsDone(): boolean
EndState
- gameResults: GameResul
Extends


PlayerBoard
t<AbstractPit>
AbstractPit
t<AbstractPits>
AbstractPit
st<AbstractPit>
spectiveBoard(): List<Abstract
List<Integer>
nesLeft(): int
nesLeft(): int
(): void
aha(): AbstractPit
aha(): AbstractPit
<<abstract>>
AbstractPit
stoneAmount: int
isActive: boolean
- nextPit: AbstractPit
- oppositePit: AbstractPit
+ getNextPit(): AbstractPit
+ getStoneAmount(): int
+ incrementBy(count: int): void
+ clearAmount(): void
+ getOppositePit() AbstractPit
+ setOppositePit(pit: AbstractPit): void
+ toggleActive(): void
+ isActive(): boolean
+ setNextPit(pit: AbstractPit): void
+ handleIncrement(stones: int): boolean
# abstract handleInternal(stones: int): void
# abstract incrementKalaha(stones: int): vo
# abstract shouldPlayerChange(): boolean
Extends


Player1MoveState
Player2MoveState

HousePit
KalahPit