<<enum>> **PlayerSpace** CardType +discardedDeck: List guard, priest, +hand: List baron, handmaid, +isSafe: boolean prince, king, +knownOpponentCard: CardType countess, +clone(out clone:PlayerSpace) princess +getTheOtherCardInHand(in firstCard:CardType, +getStrength() out secondCard:CardType) **RoundState** +deck: List +spaceOfFirstPlayer: PlayerSpace +spaceOfSecondPlayer: PlayerSpace <<enum>> +openDiscardedCards: List Winner +hiddenDiscardedCard: CardType none, +turnState: TurnState firstPlayer, +winByComparison: boolean secondPlayer +winner: Winner +rnd: Random +switchState() +drawCardForPlayerSpace(in space:PlayerSpace) +drawCardFromDeck(out card:CardType) +getProbabilityMap(out cardProbabilityMap:Map<CardType, Double>) +getHiddenCards(out hiddenCards:List) +discardAndDraw(in card:CardType,in space:PlayerSpace) <<enum>> **TurnState** notStarted, <<enum>> firstPlayer, DecisionType secondPlayer, guard_priest, guard_baron, guard_handmaid, guard_prince, guard_king, guard_countess, ended guard_princess, priestPlay, baronPlay, handmaidPlay, prince_onOpponent, prince_onMyself, kingPlay, countes_withKingOrPrince, countess_defaultPlay, princessPlay

+getDecisions(in hand:List,out decisions:List)