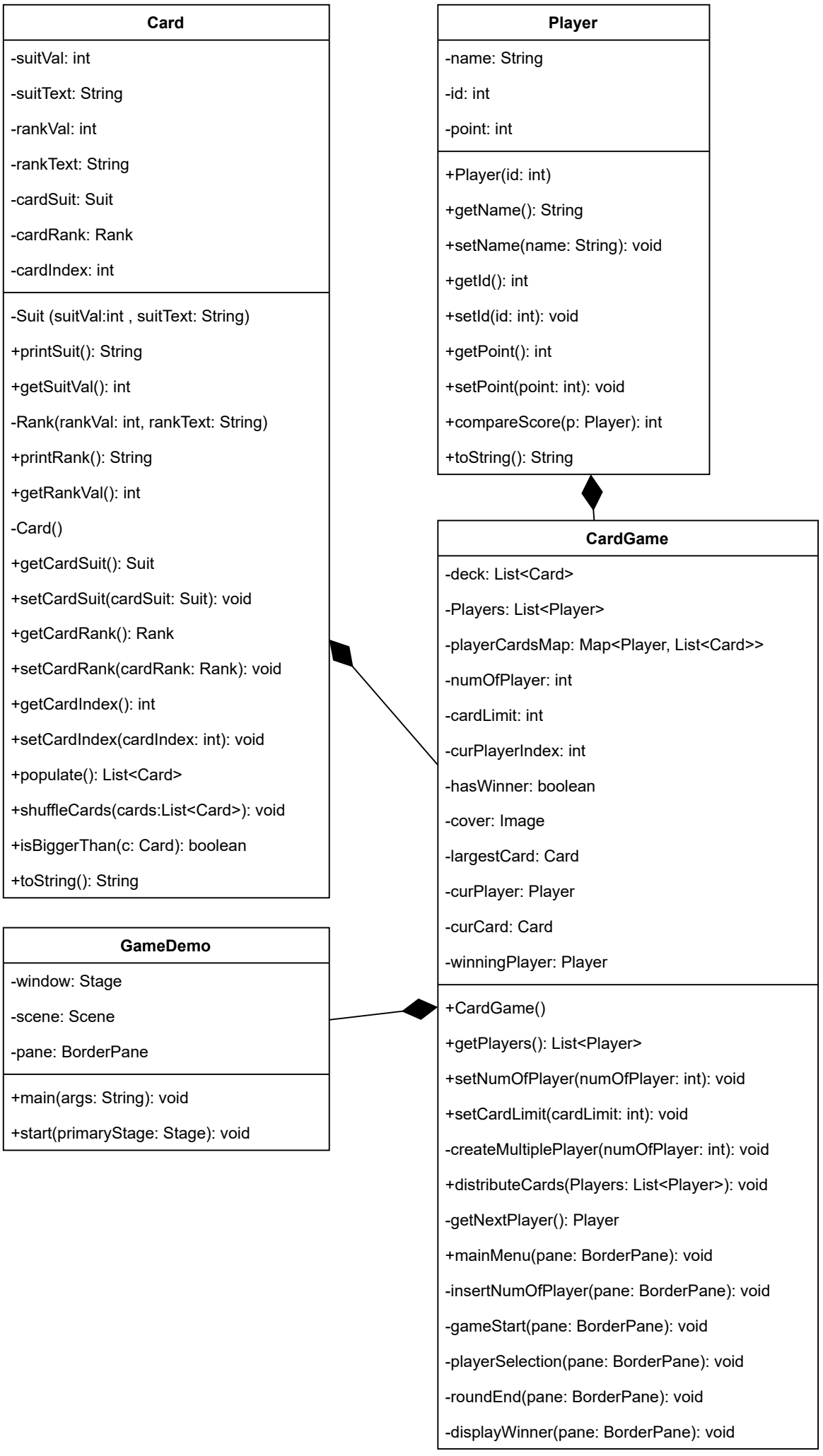
Card
-suitVal: int
-suitText: String
-rankVal: int
-rankText: String
-cardSuit: Suit
-cardRank: Rank
-cardIndex: int
-Suit (suitVal:int , suitText: String)
+printSuit(): String
+getSuitVal(): int
-Rank(rankVal: int, rankText: String)
+printRank(): String
+getRankVal(): int
-Card()
+getCardSuit(): Suit
+setCardSuit(cardSuit: Suit): void
+getCardRank(): Rank
+setCardRank(cardRank: Rank): void
+getCardIndex(): int
+setCardIndex(cardIndex: int): void
+populate(): List<Card>
+shuffleCards(cards:List<Card>): void
+isBiggerThan(c: Card): boolean
+toString(): String
Player
-name: String
-id: int
-point: int
+Player(id: int)
+getName(): String
+setName(name: String): void
+getId(): int
+setId(id: int): void
+getPoint(): int
+setPoint(point: int): void
+compareScore(p: Player): int
+toString(): String
CardGame
-deck: List<Card>
-Players: List<Player>
-playerCardsMap: Map<Player, List<Card>>
-numOfPlayer: int
-cardLimit: int
-curPlayerIndex: int
-hasWinner: boolean
-cover: Image
-largestCard: Card
-curPlayer: Player
-curCard: Card
-winningPlayer: Player
+CardGame()
+getPlayers(): List<Player>
+setNumOfPlayer(numOfPlayer: int): void
+setCardLimit(cardLimit: int): void
-createMultiplePlayer(numOfPlayer: int): void
+distributeCards(Players: List<Player>): void
-getNextPlayer(): Player
+mainMenu(pane: BorderPane): void
-insertNumOfPlayer(pane: BorderPane): void
-gameStart(pane: BorderPane): void
-playerSelection(pane: BorderPane): void
-roundEnd(pane: BorderPane): void
-displayWinner(pane: BorderPane): void
GameDemo
-window: Stage
-scene: Scene
-pane: BorderPane
+main(args: String): void
+start(primaryStage: Stage): void