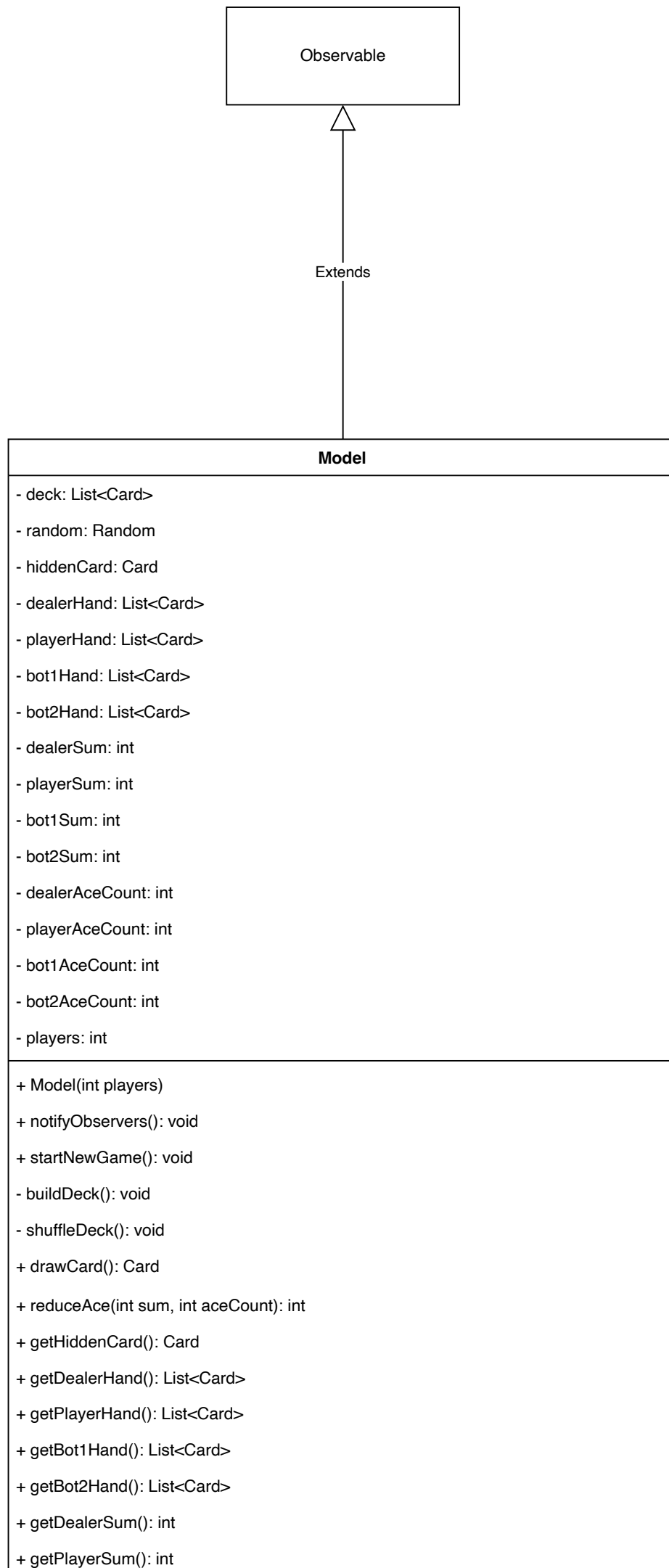


JBlackJack
<ul style="list-style-type: none"> <li>+ BOARD_WIDTH: int</li> <li>+ BOARD_HEIGHT: int</li> <li>+ BACKGROUND_COLOR: Color</li> <li>- start: Start</li> <li>- home: Home</li> <li>- onePlayer: OnePlayer</li> <li>- twoPlayers: TwoPlayers</li> <li>- threePlayers: ThreePlayers</li> <li>- betPanel: Bet</li> <li>- avatar: int</li> <li>- importo: int</li> <li>- winCount: int</li> <li>- defeatCount: int</li> <li>- drawCount: int</li> <li>- avatarBot1: int</li> <li>- winCountBot1: int</li> <li>- defeatCountBot1: int</li> <li>- drawCountBot1: int</li> <li>- avatarBot2: int</li> <li>- winCountBot2: int</li> <li>- defeatCountBot2: int</li> <li>- drawCountBot2: int</li> <li>- frame: JFrame</li> </ul>
<ul style="list-style-type: none"> <li>+ main(String[] args): void</li> <li>- setupGUI(): void</li> <li>- startGame(): void</li> <li>- generateBotAvatars(): void</li> <li>- switchAvatarLeft(): void</li> <li>- switchAvatarRight(): void</li> <li>- bet(String nickname, int avatar, int players): void</li> <li>- startGameWithPlayers(String nickname, int avatar, int players): void</li> <li>- onePlayerSelected(String nickname, int avatar): void</li> <li>- twoPlayersSelected(String nickname, int avatar): void</li> <li>- threePlayersSelected(String nickname, int avatar): void</li> <li>- backHome(String nickname, int avatar, int result, int bot1result, int bot2result): void</li> <li>- updateCounts(int result, int bot1result, int bot2result): void</li> <li>- updateBotCounts(int result, int botNumber): void</li> <li>- switchToPanel(JPanel panel): void</li> </ul>



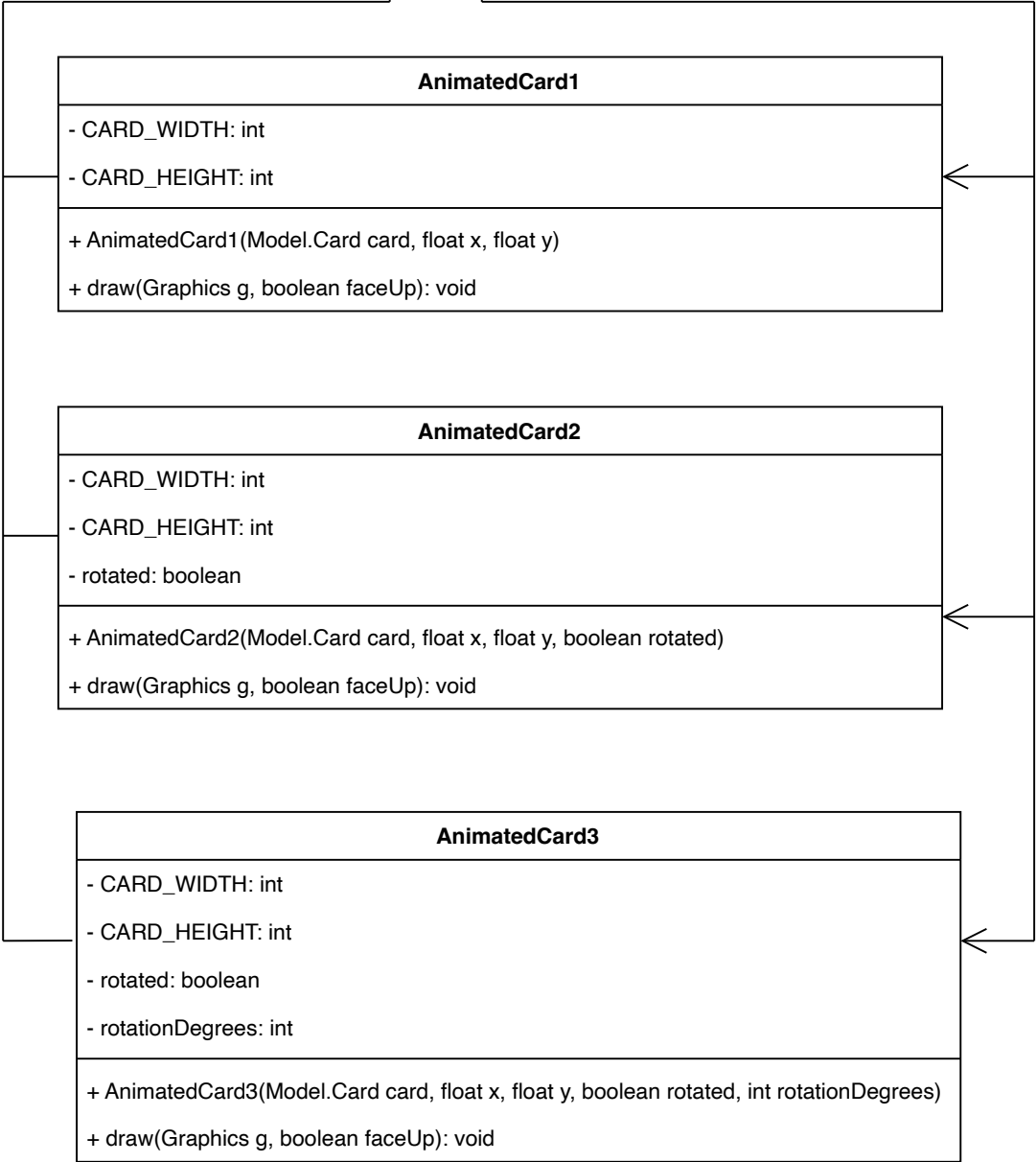
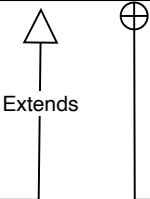
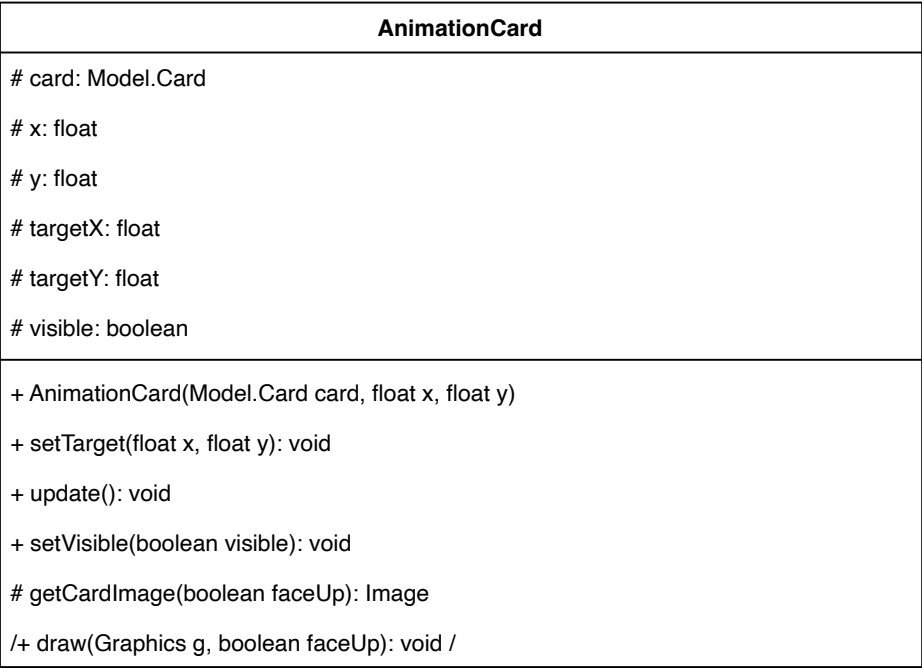
+ getBot1Sum(): int
+ getBot2Sum(): int
+ getDealerAceCount(): int
+ getPlayerAceCount(): int
+ getBot1AceCount(): int
+ getBot2AceCount(): int
+ getPlayers(): int
+ setDealerSum(int sum): void
+ setPlayerrSum(int sum): void
+ setBot1Sum(int sum): void
+ setBot2Sum(int sum): void
+ setDealerAceCount(int count): void
+ setPlayerAceCount(int count): void
+ setBot1AceCount(int count): void
+ setBot2AceCount(int count): void

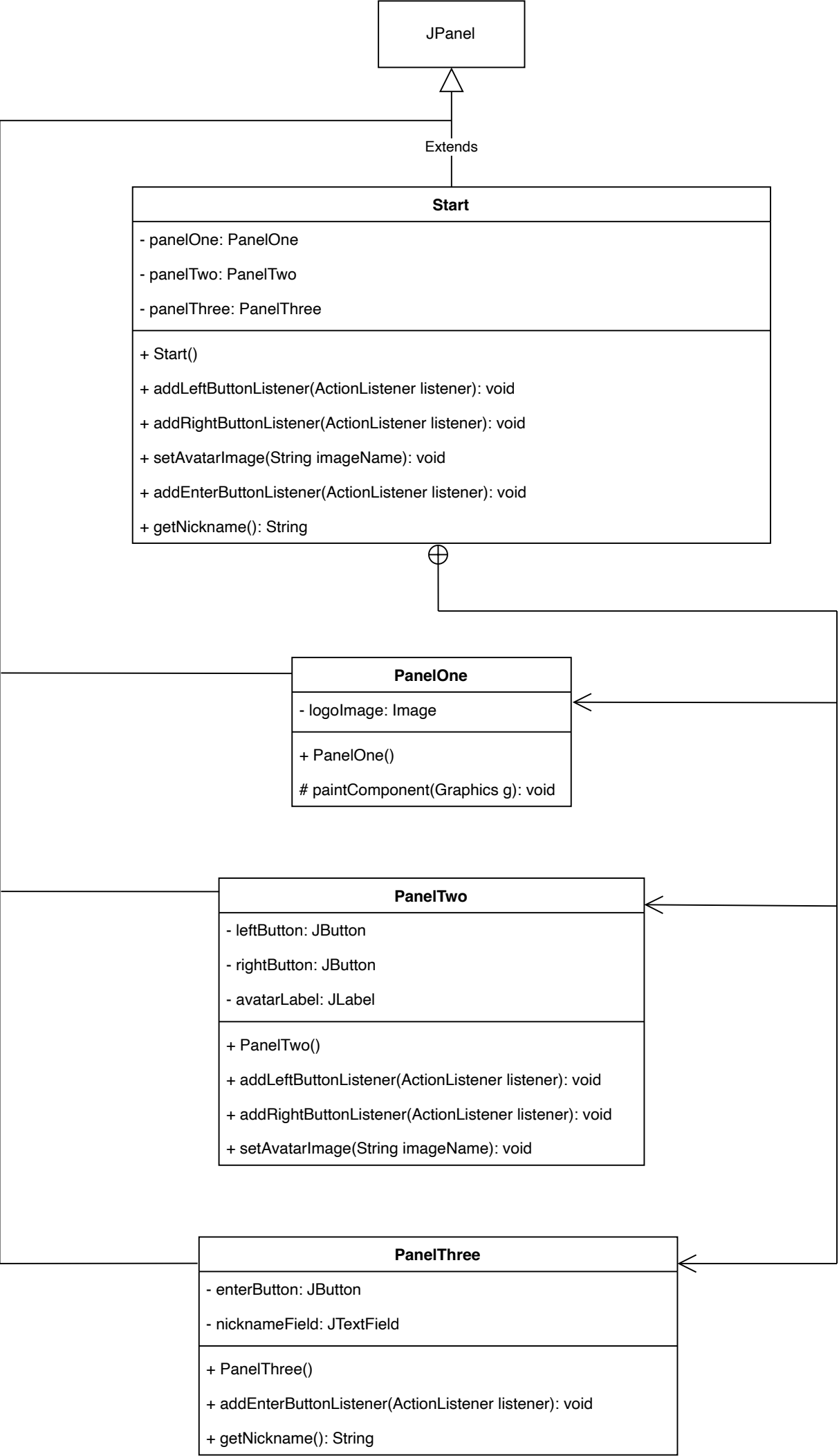


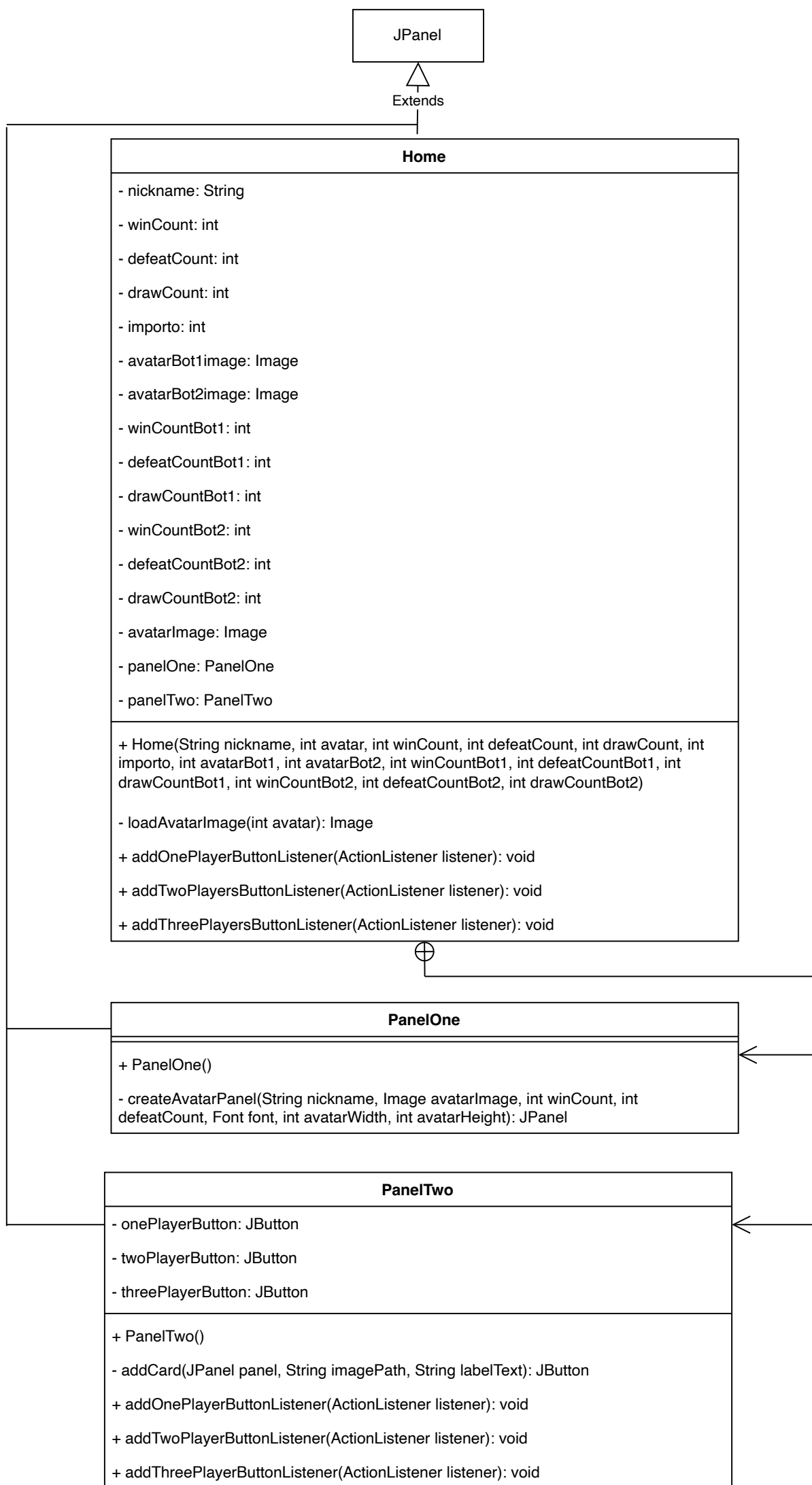
Card
- value: String
- type: String
+ Card(String value, String type)
+ getValue(): int
+ isAce(): boolean
+ getImagePath(): String
+ toString(): String

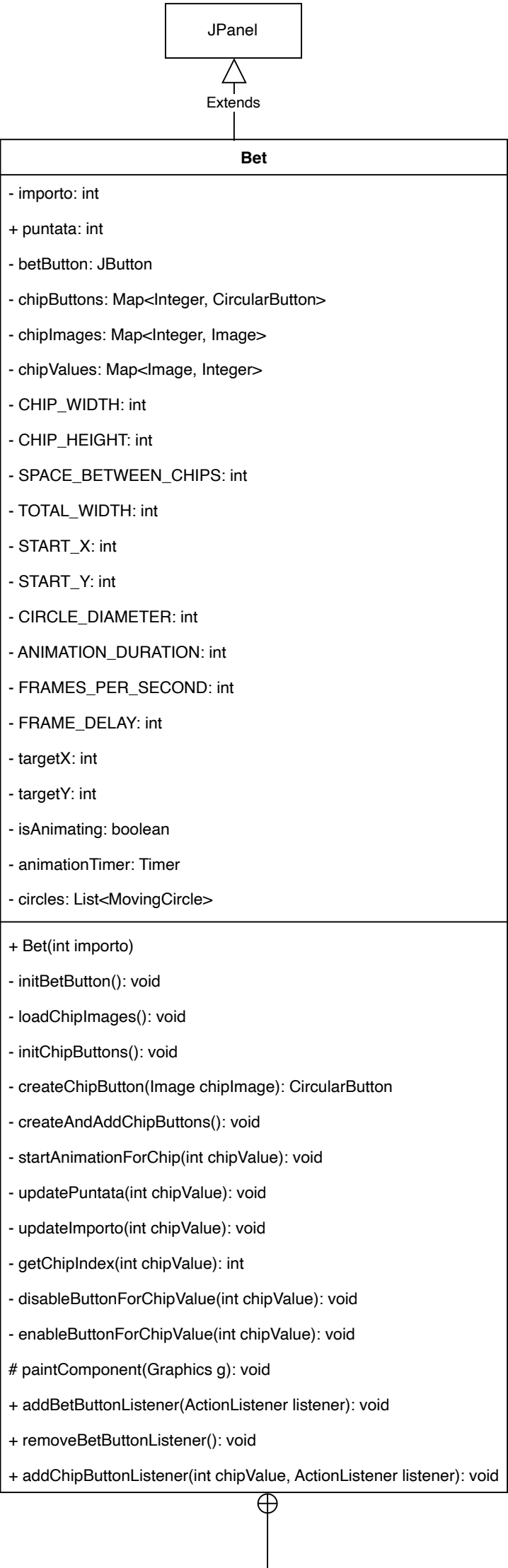
Controller
- model: Model
- view: GameView
+ Controller(Model model, GameView view)
- hitAction(): void
- stayAction(): void

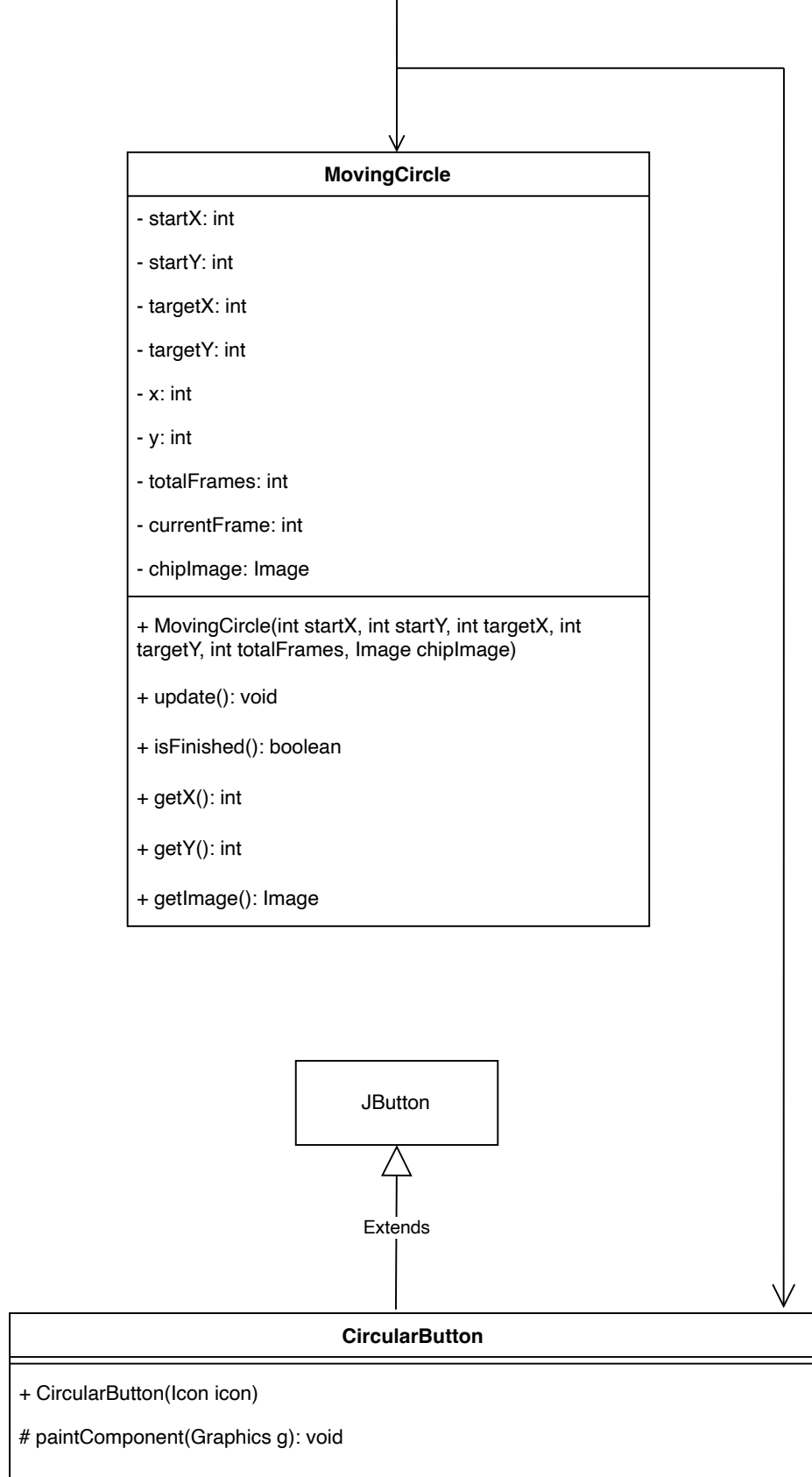
AudioManager
- instance: AudioManager
- clip: Clip
+ getInstance(): AudioManager
+ play(String filename): void



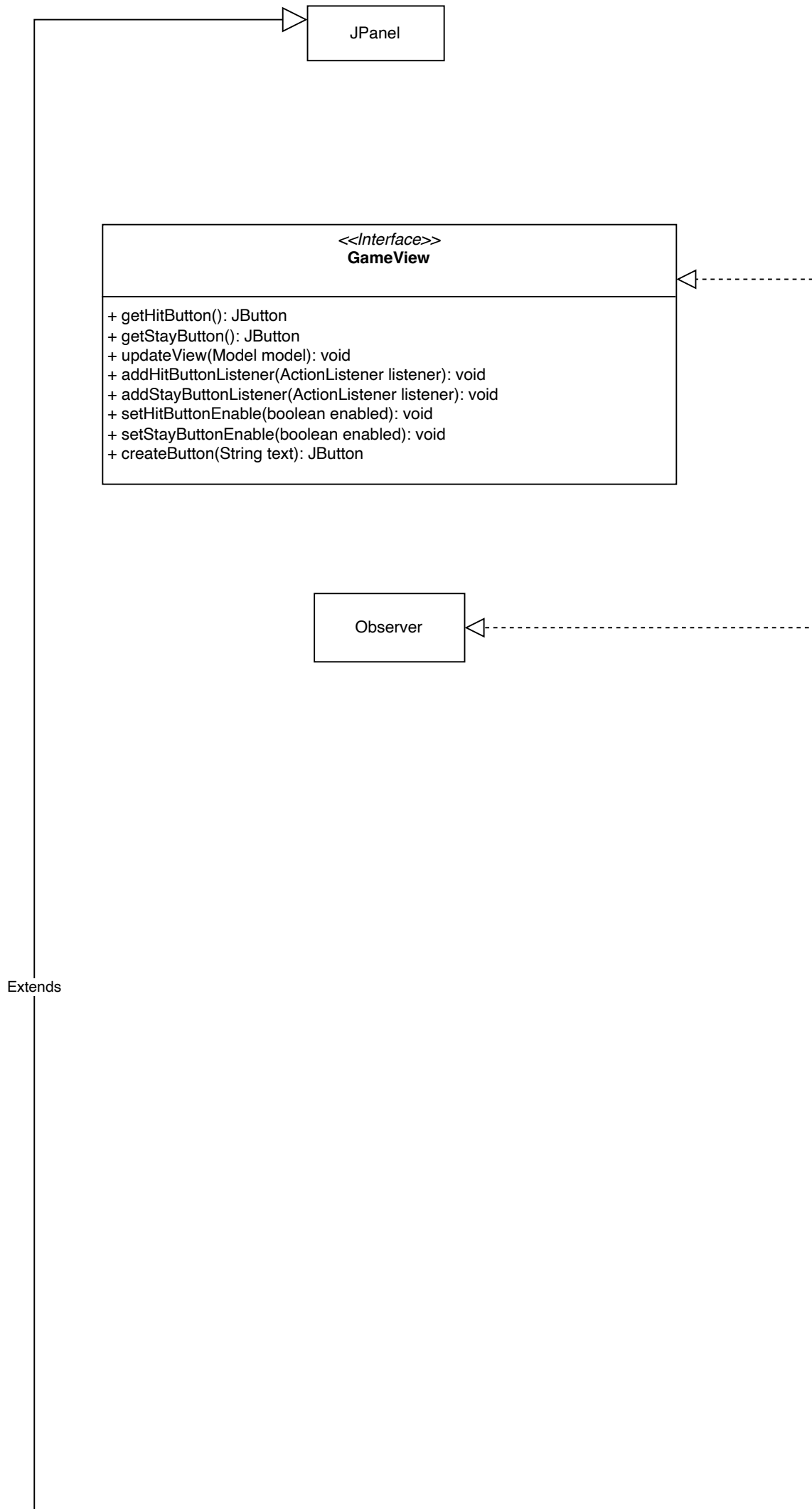


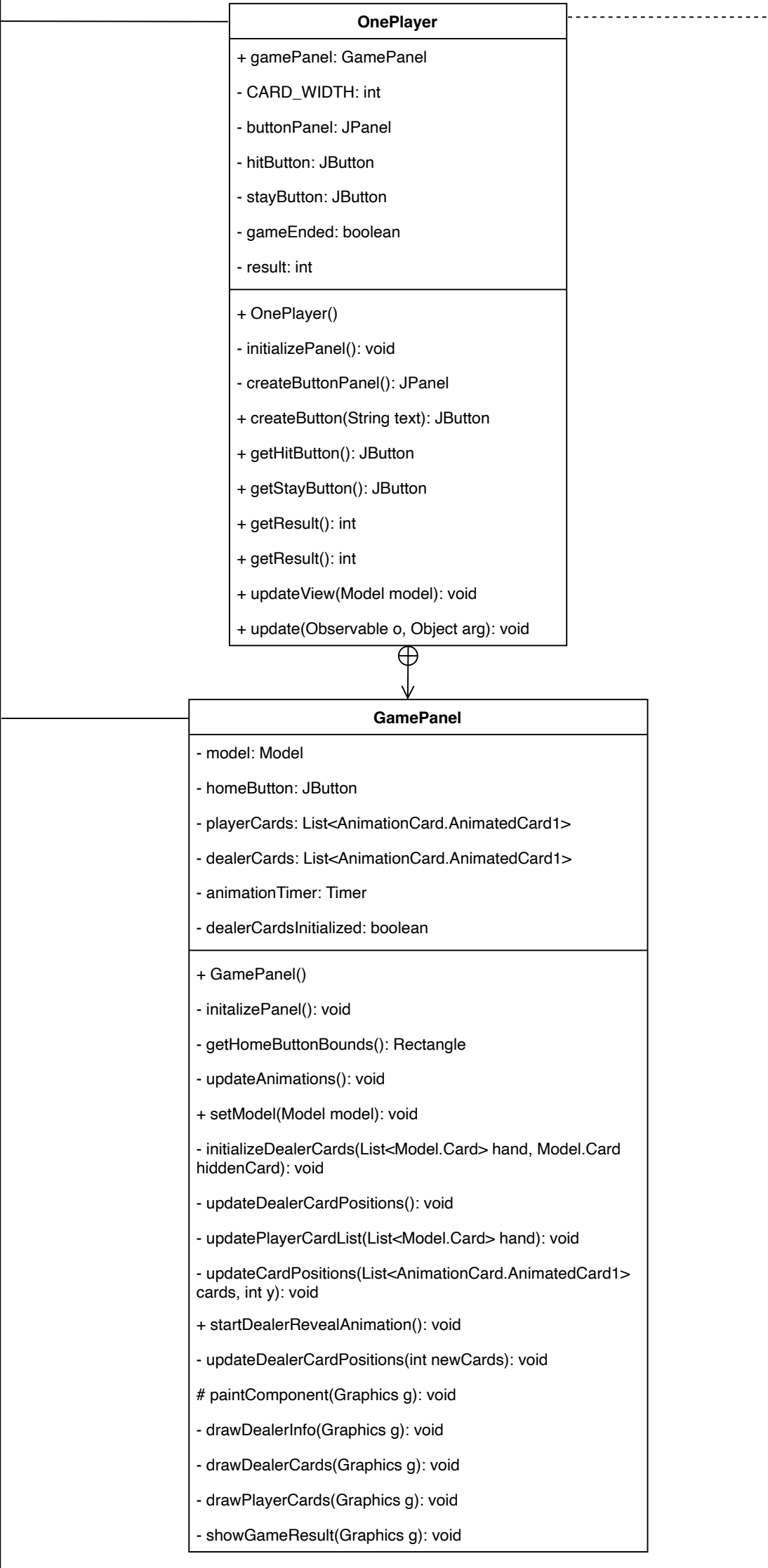


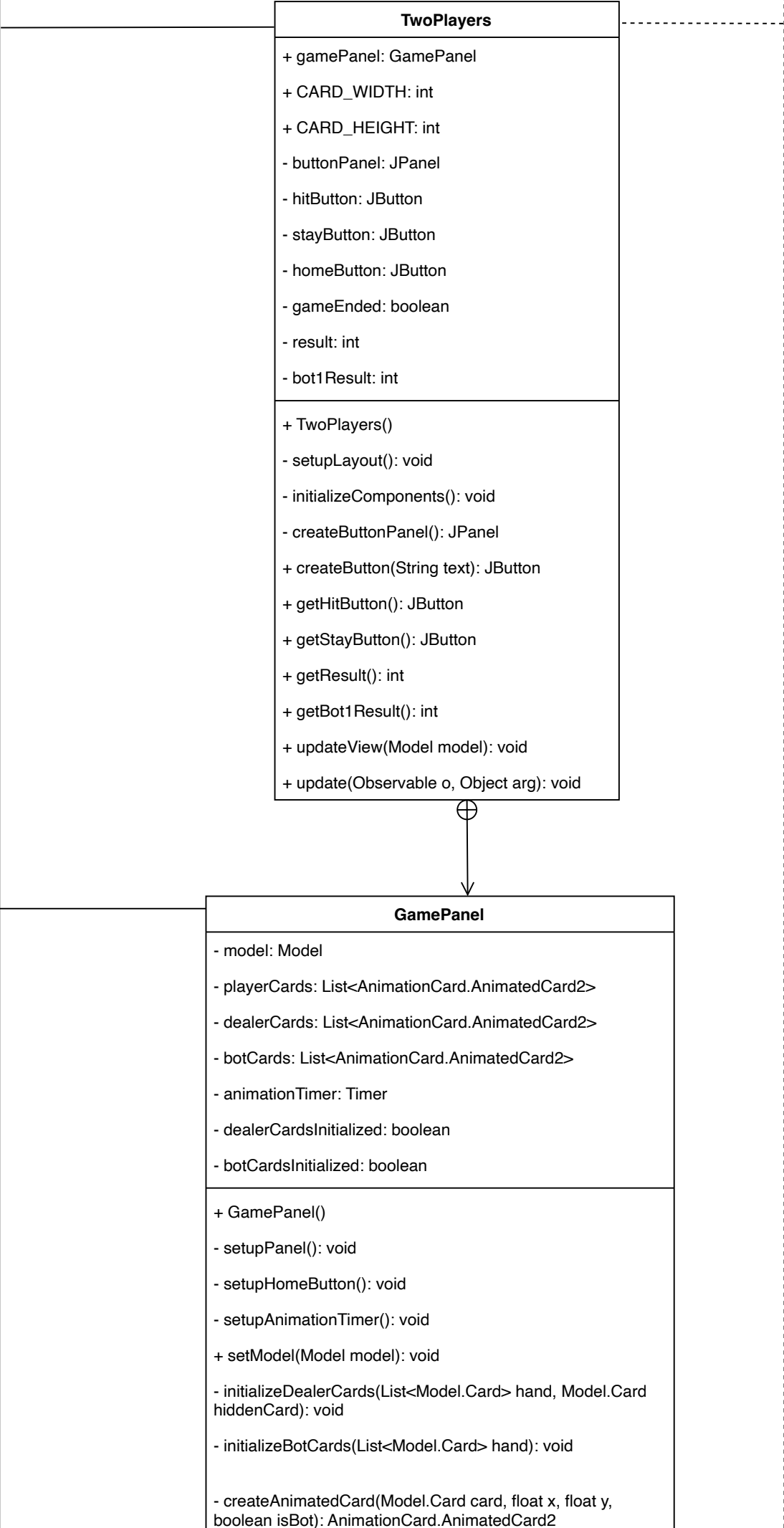












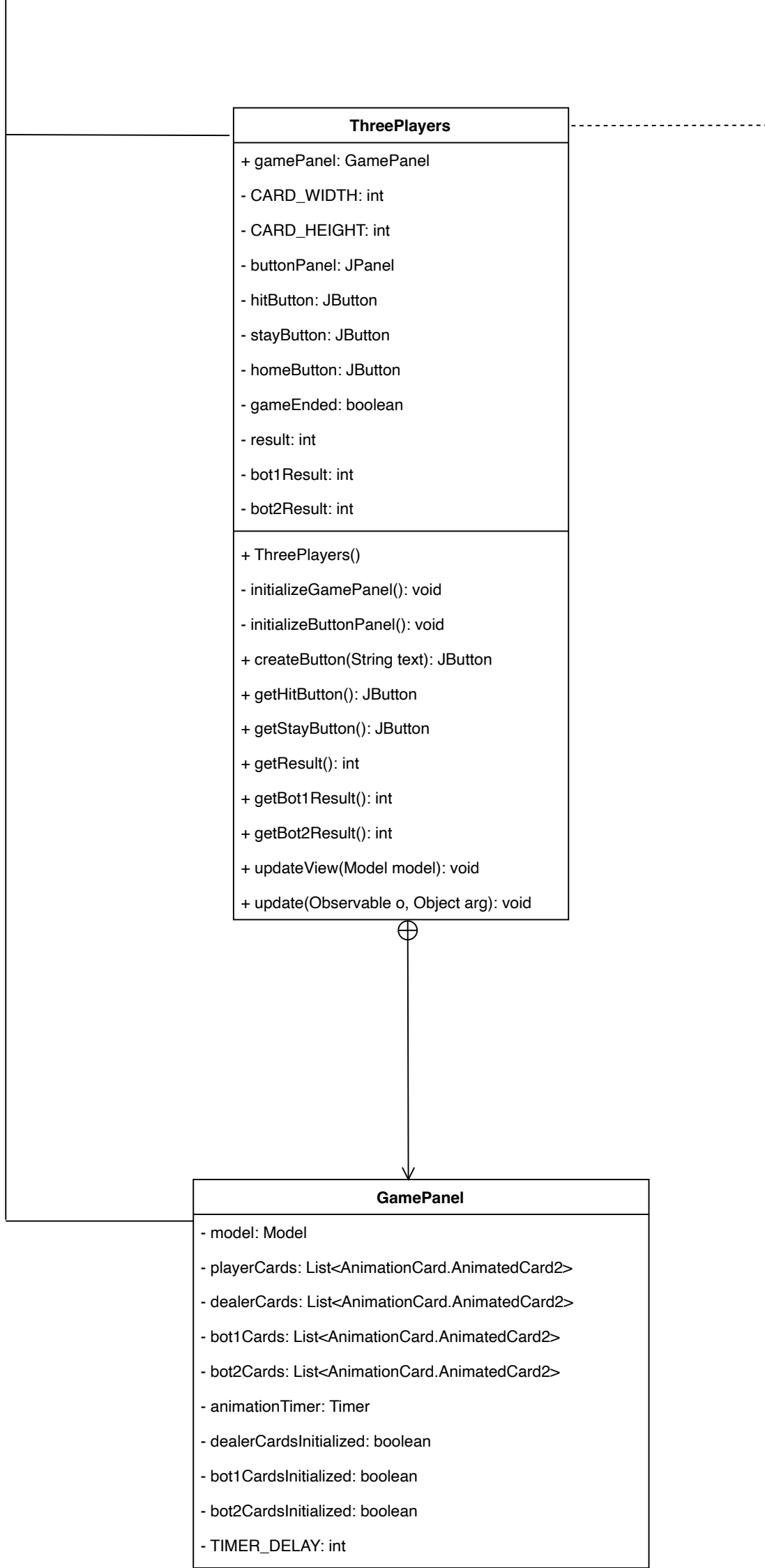
```
- updateDealerCardPositions(): void
- updateBotCardPositions(): void
- updatePlayerCardList(List<Model.Card> hand): void
- updatePlayerCardPosition(): void
- updateCardPositions(): void

+ startDealerRevealAnimation(): void
+ startBotRevealAnimation(): void
# paintComponent(Graphics g): void
- drawCards(Graphics g): void
- drawLabels(Graphics g): void
- drawDealerLabel(Graphics g, Fontmetrics fm): void
- drawPlayer2Label(Graphics g, Fontmetrics fm): void
- showGameResult(Graphics g): void
- determineResults(int playerFinalSum, int dealerFinalSum,
int bo1FinalSum): void
- calculateResult(int playerSum, int dealerSum): int

- getResultMessage(int playerSum, int dealerSum): String

- drawResultMessage(Graphics g, FontMetrics fm, String
message): void

+ addHomeButtonListener(ActionListener listener): void
```



```
+ GamePanel()

- initializeHomeButton(): void

- initializeAnimationTimer(): void

+ setModel(Model model): void

- initializeDealerCards(List<Model.Card> hand, Model.Card
hiddenCard): void

- initializeBot1Cards(List<Model.Card> hand): void

- initializeBot2Cards(List<Model.Card> hand): void

- updatePlayerCardList(List<Model.Card> hand): void

- updateCardPositions(): void

- updatePlayerCardPosition(): void

- updateDealerCardPositions(): void

- updateBot1CardPositions(): void

- updateBot2CardPositions(): void

+ startDealerRevealAnimation(): void

+ startBot1RevealAnimation(): void

+ startBot2RevealAnimation(): void

# paintComponent(Graphics g): void

- drawLabels(Graphics g): void

- drawLabel(Graphics g, String text, int x, int y, double angle):
void

- drawRotatedLabel(Graphics g, String text, int x, int y, double
angle): void

- drawCards(Graphics g): void

- showGameResult(Graphics g): void

- determineResult(int finalSum, int dealerFinalSum): void

- getResultMessage(int resultCode): String

+ addHomeButtonListener(ActionListener listener): void
```