Explanation:

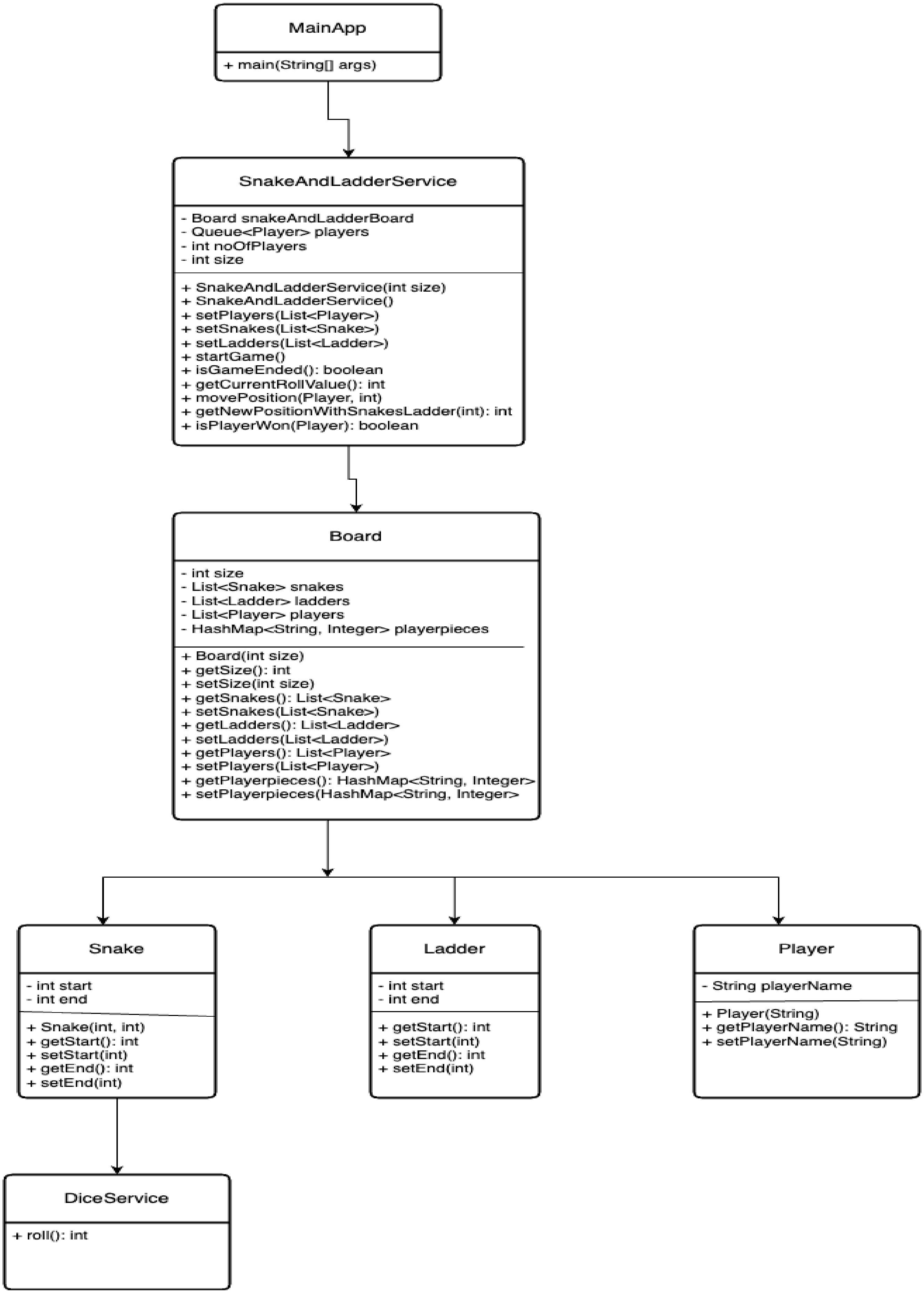
1. Driver Class:
   - The **MainApp** class contains the **main** method which acts as the entry point of the application. It reads input from the user and initializes the game by interacting with the **SnakeAndLadderService**. .
2. SnakeAndLadderService Class:
   - This class contains the core logic of the game. It manages the players, snakes, ladders, and the board. It provides methods to start the game, roll the dice, move players, and check the game status.
3. Board Class:
   - The **Board** class represents the game board. It stores the size of the board, lists of snakes, ladders, and players, and a hashmap to track each player's current position.
4. Snake Class:
   - Represents a snake on the board. It has a start (head) and an end (tail) position.
5. Ladder Class:
   - Represents a ladder on the board. It has a start (bottom) and an end (top) position.
6. Player Class:
   - Represents a player in the game. It has a name and methods to get and set the name.
7. DiceService Class:
   - Provides a method to simulate rolling a six-sided die.

Interaction Flow:

1. The **MainApp** class initializes the game by reading inputs and setting up the snakes, ladders, and players.
2. The **SnakeAndLadderService** class is used to start the game and manage the game flow.
3. The **Board** class maintains the state of the game, including player positions and the positions of snakes and ladders.
4. The **DiceService** class is used by **SnakeAndLadderService** to simulate dice rolls.

5. Players take turns to roll the dice and move their pieces on the board. The game continues until a player wins.

This architecture diagram helps visualize the structure and interactions within the Snake and Ladder game application.


MainApp
+ main(String[] args)
SnakeAndLadderService
- Board snakeAndLadderBoard
- Queue<Player> players
- int noOfPlayers
- int size
+ SnakeAndLadderService(int size)
+ SnakeAndLadderService()
+ setPlayers(List<Player>)
+ setSnakes(List<Snake>)
+ setLadders(List<Ladder>)
+ startGame()
+ isGameEnded(): boolean
+ getCurrentRollValue(): int
+ movePosition(Player, int)
+ getNewPositionWithSnakesLadder(int): int
+ isPlayerWon(Player): boolean
Board
- int size
- List<Snake> snakes
- List<Ladder> ladders
- List<Player> players
- HashMap<String, Integer> playerpieces
+ Board(int size)
+ getSize(): int
+ setSize(int size)
+ getSnakes(): List<Snake>
+ setSnakes(List<Snake>)
+ getLadders(): List<Ladder>
+ setLadders(List<Ladder>)
+ getPlayers(): List<Player>
+ setPlayers(List<Player>)
+ getPlayerpieces(): HashMap<String, Integer>
+ setPlayerpieces(HashMap<String, Integer>
Snake
- int start
- int end
+ Snake(int, int)
+ getStart(): int
+ setStart(int)
+ getEnd(): int
+ setEnd(int)
Ladder
- int start
- int end
+ getStart(): int
+ setStart(int)
+ getEnd(): int
+ setEnd(int)
Player
- String playerName
+ Player(String)
+ getPlayerName(): String
+ setPlayerName(String)
DiceService
+ roll(): int