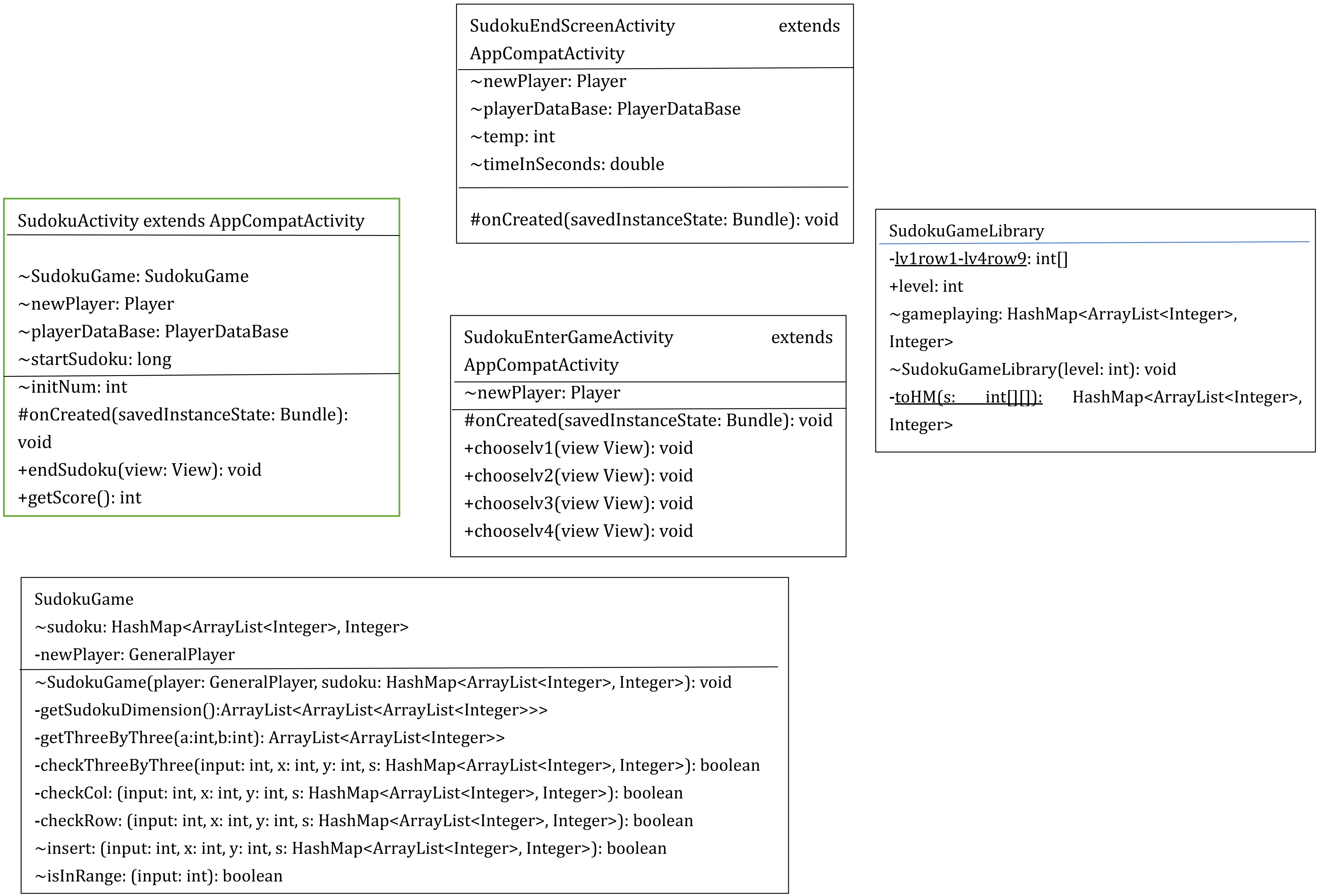

SudokuEndScreenActivity extends AppCompatActivity

~newPlayer: Player
~playerDataBase: PlayerDataBase
~temp: int
~timeInSeconds: double

#onCreated(savedInstanceState: Bundle): void

SudokuActivity extends AppCompatActivity

~SudokuGame: SudokuGame
~newPlayer: Player
~playerDataBase: PlayerDataBase
~startSudoku: long

~initNum: int
#onCreated(savedInstanceState: Bundle): void
+endSudoku(view: View): void
+getScore(): int

SudokuEnterGameActivity extends AppCompatActivity

~newPlayer: Player

#onCreated(savedInstanceState: Bundle): void
+chooselv1(view View): void
+chooselv2(view View): void
+chooselv3(view View): void
+chooselv4(view View): void

SudokuGameLibrary

-lv1row1-lv4row9: int[]
+level: int
~gameplaying: HashMap<ArrayList<Integer>, Integer>
~SudokuGameLibrary(level: int): void
-toHM(s: int[][]): HashMap<ArrayList<Integer>, Integer>

SudokuGame
~sudoku: HashMap<ArrayList<Integer>, Integer>
-newPlayer: GeneralPlayer

~SudokuGame(player: GeneralPlayer, sudoku: HashMap<ArrayList<Integer>, Integer>): void
-getSudokuDimension():ArrayList<ArrayList<ArrayList<Integer>>>
-getThreeByThree(a:int,b:int): ArrayList<ArrayList<Integer>>
-checkThreeByThree(input: int, x: int, y: int, s: HashMap<ArrayList<Integer>, Integer>): boolean
-checkCol: (input: int, x: int, y: int, s: HashMap<ArrayList<Integer>, Integer>): boolean
-checkRow: (input: int, x: int, y: int, s: HashMap<ArrayList<Integer>, Integer>): boolean
~insert: (input: int, x: int, y: int, s: HashMap<ArrayList<Integer>, Integer>): boolean
~isInRange: (input: int): boolean