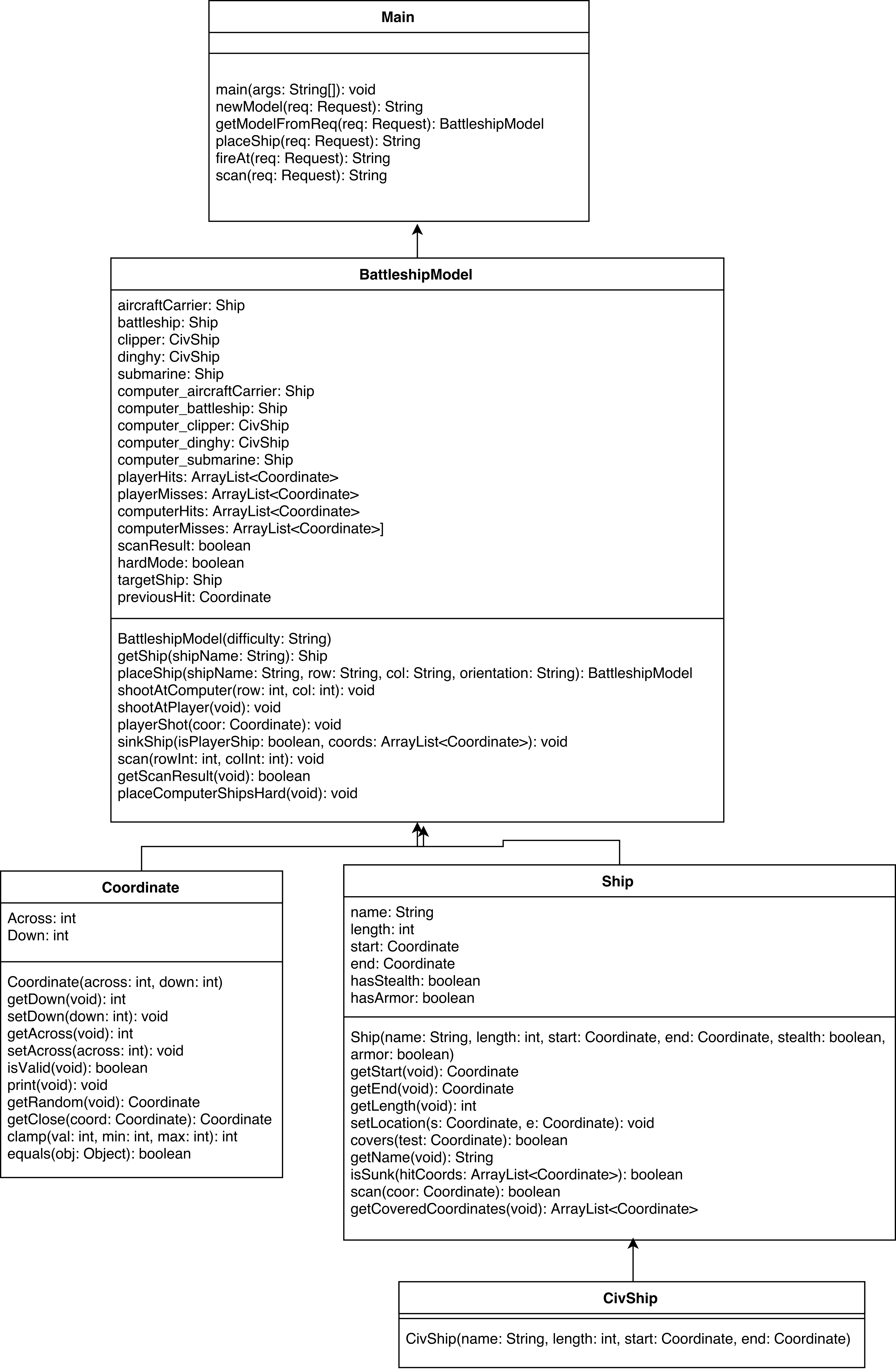
Main
main(args: String[]): void
newModel(req: Request): String
getModelFromReq(req: Request): BattleshipModel
placeShip(req: Request): String
fireAt(req: Request): String
scan(req: Request): String
BattleshipModel
aircraftCarrier: Ship
battleship: Ship
clipper: CivShip
dinghy: CivShip
submarine: Ship
computer_aircraftCarrier: Ship
computer_battleship: Ship
computer_clipper: CivShip
computer_dinghy: CivShip
computer_submarine: Ship
playerHits: ArrayList<Coordinate>
playerMisses: ArrayList<Coordinate>
computerHits: ArrayList<Coordinate>
computerMisses: ArrayList<Coordinate>]
scanResult: boolean
hardMode: boolean
targetShip: Ship
previousHit: Coordinate
BattleshipModel(difficulty: String)
getShip(shipName: String): Ship
placeShip(shipName: String, row: String, col: String, orientation: String): BattleshipModel
shootAtComputer(row: int, col: int): void
shootAtPlayer(void): void
playerShot(coor: Coordinate): void
sinkShip(isPlayerShip: boolean, coords: ArrayList<Coordinate>): void
scan(rowInt: int, colInt: int): void
getScanResult(void): boolean
placeComputerShipsHard(void): void
Coordinate
Across: int
Down: int
Coordinate(across: int, down: int)
getDown(void): int
setDown(down: int): void
getAcross(void): int
setAcross(across: int): void
isValid(void): boolean
print(void): void
getRandom(void): Coordinate
getClose(coord: Coordinate): Coordinate
clamp(val: int, min: int, max: int): int
equals(obj: Object): boolean
Ship
name: String
length: int
start: Coordinate
end: Coordinate
hasStealth: boolean
hasArmor: boolean
Ship(name: String, length: int, start: Coordinate, end: Coordinate, stealth: boolean, armor: boolean)
getStart(void): Coordinate
getEnd(void): Coordinate
getLength(void): int
setLocation(s: Coordinate, e: Coordinate): void
covers(test: Coordinate): boolean
getName(void): String
isSunk(hitCoords: ArrayList<Coordinate>): boolean
scan(coor: Coordinate): boolean
getCoveredCoordinates(void): ArrayList<Coordinate>
CivShip
CivShip(name: String, length: int, start: Coordinate, end: Coordinate)