Ship
- position : Position
- hits : Position[]
- isSubmerged : boolean
+ attack(position: Position, laser: boolean) : ShipAttackResult
+ sonar(position: Position) : Position[]
+ turn() : boolean

Fleet
- sonarPulseAttacks : Int
- sunkSomeEnemyShip : boolean
+ attack(position: Position, laser: boolean) : FleetAttackResult
+ sonar(position: Position) : Position[]
+ turn() : boolean

ShipType
+ relativePositions : Position[]
+ captainsQuarters : Position
+ partOfShip(p : Position) : boolean

<< data type >>
FleetAttackResult

<< enumeration >>
SimpleFleetAttackResult
Hit
Miss
Surrender

<< enumeration >>
ShipAttackResult
Hit
Miss
Sunk

<< data type >>
Position
+ x: Int
+ y: Int

player1: Fleet
player2: Fleet

: Ship

: ShipType

alt

turn()

attack(position)

loop (for each ship)

attack(position)

the attack result

partOfShip(relativePos)

isCaptainsQuarters(relativePos)


formal type: Fleet
formal type: Fleet
formal type: Ship
formal type: ShipType

is winner

revealed positions

loop (for each ship)

sonar(position)

partOfShip(relativePos)

is winner

revealed positions

loop (for each ship)

sonar(position)

partOfShip(relativePos)