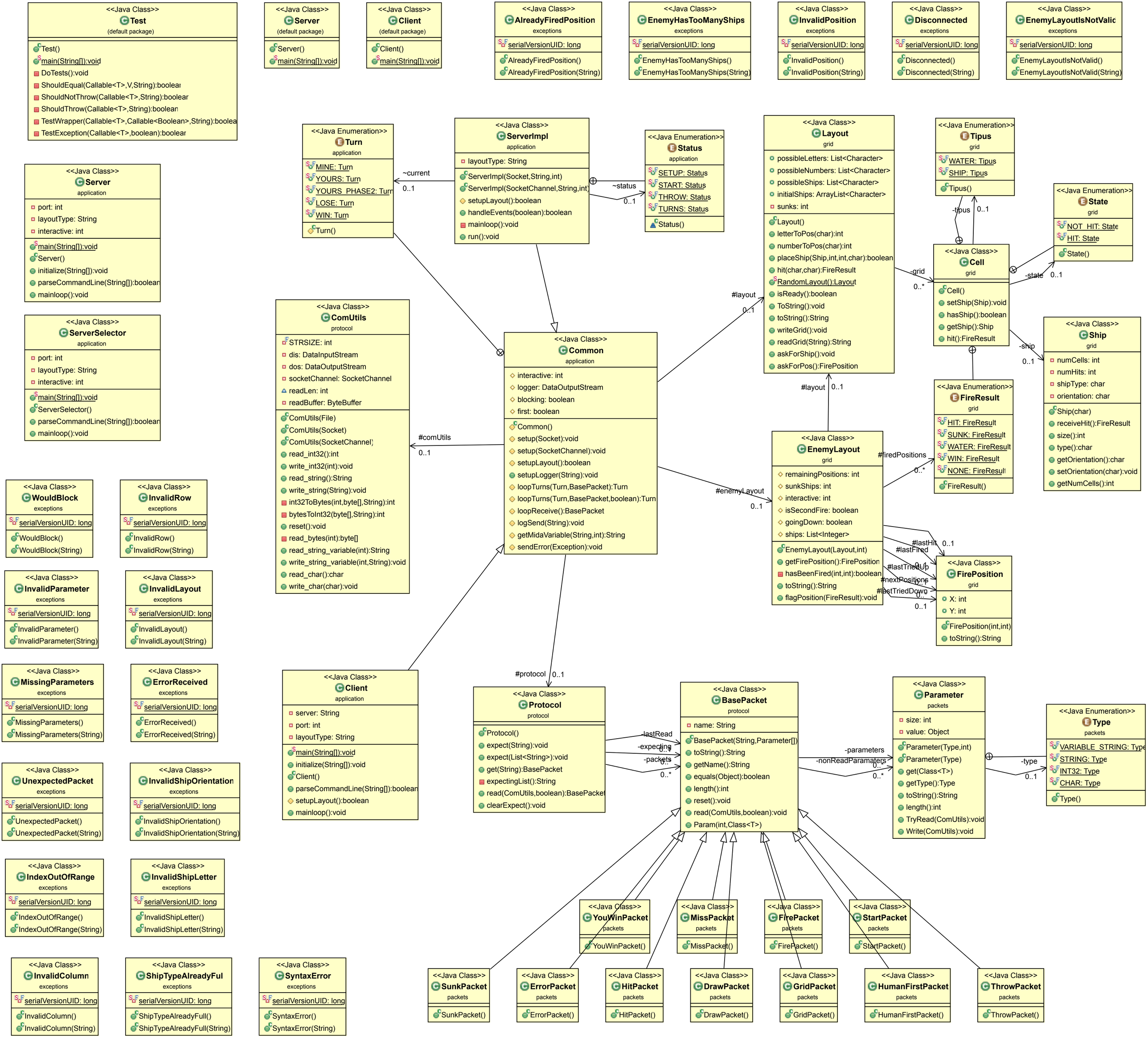
<<Java Class>> Test (default package)
Test()
main(String[]):void
DoTests():void
ShouldEqual(Callable<T>,V,String):boolean
ShouldNotThrow(Callable<T>,String):boolean
ShouldThrow(Callable<T>,String):boolean
TestWrapper(Callable<T>,Callable<Boolean>,String):boolean
TestException(Callable<T>,boolean):boolean
<<Java Class>> Server (default package)
Server()
main(String[]):void
<<Java Class>> Client (default package)
Client()
main(String[]):void
<<Java Class>> AlreadyFiredPosition exceptions
serialVersionUID: long
AlreadyFiredPosition()
AlreadyFiredPosition(String)
<<Java Class>> EnemyHasTooManyShips exceptions
serialVersionUID: long
EnemyHasTooManyShips()
EnemyHasTooManyShips(String)
<<Java Class>> InvalidPosition exceptions
serialVersionUID: long
InvalidPosition()
InvalidPosition(String)
<<Java Class>> Disconnected exceptions
serialVersionUID: long
Disconnected()
Disconnected(String)
<<Java Class>> EnemyLayoutIsNotValid exceptions
serialVersionUID: long
EnemyLayoutIsNotValid()
EnemyLayoutIsNotValid(String)
<<Java Enumeration>> Turn application
MINE: Turn
YOURS: Turn
YOURS_PHASE2: Turn
LOSE: Turn
WIN: Turn
Turn()
<<Java Class>> ServerImpl application
layoutType: String
ServerImpl(Socket,String,int)
ServerImpl(SocketChannel,String,int)
setupLayout():boolean
handleEvents(boolean):boolean
mainloop():void
run():void
<<Java Enumeration>> Status application
SETUP: Status
START: Status
THROW: Status
TURNS: Status
Status()
<<Java Class>> Layout grid
possibleLetters: List<Character>
possibleNumbers: List<Character>
possibleShips: List<Character>
initialShips: ArrayList<Character>
sunks: int
Layout()
letterToPos(char):int
numberToPos(char):int
placeShip(Ship,int,int,char):boolean
hit(char,char):FireResult
RandomLayout():Layout
isReady():boolean
ToString():void
toString():String
writeGrid():void
readGrid(String):String
askForShip():void
askForPos():FirePosition
<<Java Enumeration>> Tipus grid
WATER: Tipus
SHIP: Tipus
Tipus()
<<Java Enumeration>> State grid
NOT_HIT: State
HIT: State
State()
<<Java Class>> Cell grid
Cell()
setShip(Ship):void
hasShip():boolean
getShip():Ship
hit():FireResult
<<Java Class>> Server application
port: int
layoutType: String
interactive: int
main(String[]):void
Server()
initialize(String[]):void
parseCommandLine(String[]):boolean
mainloop():void
<<Java Class>> ServerSelector application
port: int
layoutType: String
interactive: int
main(String[]):void
ServerSelector()
parseCommandLine(String[]):boolean
mainloop():void
<<Java Class>> ComUtils protocol
STRSIZE: int
dis: DataInputStream
dos: DataOutputStream
socketChannel: SocketChannel
readLen: int
readBuffer: ByteBuffer
ComUtils(File)
ComUtils(Socket)
ComUtils(SocketChannel)
read_int32():int
write_int32(int):void
read_string():String
write_string(String):void
int32ToBytes(int,byte[],String):int
bytesToInt32(byte[],String):int
reset():void
read_bytes(int):byte[]
read_string_variable(int):String
write_string_variable(int,String):void
read_char():char
write_char(char):void
<<Java Class>> Common application
interactive: int
logger: DataOutputStream
blocking: boolean
first: boolean
Common()
setup(Socket):void
setup(SocketChannel):void
setupLayout():boolean
setupLogger(String):void
loopTurns(Turn,BasePacket):Turn
loopTurns(Turn,BasePacket,boolean):Turn
loopReceive():BasePacket
logSend(String):void
getMidaVariable(String,int):String
sendError(Exception):void
<<Java Class>> EnemyLayout grid
remainingPositions: int
sunkShips: int
interactive: int
isSecondFire: boolean
goingDown: boolean
ships: List<Integer>
EnemyLayout(Layout,int)
getFirePosition():FirePosition
hasBeenFired(int,int):boolean
toString():String
flagPosition(FireResult):void
<<Java Enumeration>> FireResult grid
HIT: FireResult
SUNK: FireResult
WATER: FireResult
WIN: FireResult
NONE: FireResult
FireResult()
<<Java Class>> Ship grid
numCells: int
numHits: int
shipType: char
orientation: char
Ship(char)
receiveHit():FireResult
size():int
type():char
getOrientation():char
setOrientation(char):void
getNumCells():int
<<Java Class>> FirePosition grid
X: int
Y: int
FirePosition(int,int)
toString():String
<<Java Class>> WouldBlock exceptions
serialVersionUID: long
WouldBlock()
WouldBlock(String)
<<Java Class>> InvalidRow exceptions
serialVersionUID: long
InvalidRow()
InvalidRow(String)
<<Java Class>> InvalidParameter exceptions
serialVersionUID: long
InvalidParameter()
InvalidParameter(String)
<<Java Class>> InvalidLayout exceptions
serialVersionUID: long
InvalidLayout()
InvalidLayout(String)
<<Java Class>> MissingParameters exceptions
serialVersionUID: long
MissingParameters()
MissingParameters(String)
<<Java Class>> ErrorReceived exceptions
serialVersionUID: long
ErrorReceived()
ErrorReceived(String)
<<Java Class>> UnexpectedPacket exceptions
serialVersionUID: long
UnexpectedPacket()
UnexpectedPacket(String)
<<Java Class>> InvalidShipOrientation exceptions
serialVersionUID: long
InvalidShipOrientation()
InvalidShipOrientation(String)
<<Java Class>> IndexOutOfRange exceptions
serialVersionUID: long
IndexOutOfRange()
IndexOutOfRange(String)
<<Java Class>> InvalidShipLetter exceptions
serialVersionUID: long
InvalidShipLetter()
InvalidShipLetter(String)
<<Java Class>> InvalidColumn exceptions
serialVersionUID: long
InvalidColumn()
InvalidColumn(String)
<<Java Class>> ShipTypeAlreadyFul exceptions
serialVersionUID: long
ShipTypeAlreadyFull()
ShipTypeAlreadyFull(String)
<<Java Class>> SyntaxError exceptions
serialVersionUID: long
SyntaxError()
SyntaxError(String)
<<Java Class>> Client application
server: String
port: int
layoutType: String
main(String[]):void
initialize(String[]):void
Client()
parseCommandLine(String[]):boolean
setupLayout():boolean
mainloop():void
<<Java Class>> Protocol protocol
Protocol()
expect(String):void
expect(List<String>):void
get(String):BasePacket
expectingList():String
read(ComUtils,boolean):BasePacket
clearExpect():void
<<Java Class>> BasePacket protocol
name: String
BasePacket(String,Parameter[])
toString():String
getName():String
equals(Object):boolean
length():int
reset():void
read(ComUtils,boolean):void
Param(int,Class<T>)
<<Java Class>> Parameter packets
size: int
value: Object
Parameter(Type,int)
Parameter(Type)
get(Class<T>)
getType():Type
toString():String
length():int
TryRead(ComUtils):void
Write(ComUtils):void
<<Java Enumeration>> Type packets
VARIABLE_STRING: Type
STRING: Type
INT32: Type
CHAR: Type
Type()
<<Java Class>> YouWinPacket packets
YouWinPacket()
<<Java Class>> MissPacket packets
MissPacket()
<<Java Class>> FirePacket packets
FirePacket()
<<Java Class>> StartPacket packets
StartPacket()
<<Java Class>> SunkPacket packets
SunkPacket()
<<Java Class>> ErrorPacket packets
ErrorPacket()
<<Java Class>> HitPacket packets
HitPacket()
<<Java Class>> DrawPacket packets
DrawPacket()
<<Java Class>> GridPacket packets
GridPacket()
<<Java Class>> HumanFirstPacket packets
HumanFirstPacket()
<<Java Class>> ThrowPacket packets
ThrowPacket()
~current
0..1
~status
0..1
#layout
0..1
-grid
0..*
-tipus
0..1
-state
0..1
-ship
0..1
#layout
0..1
#comUtils
0..1
#enemyLayout
0..1
#firedPositions
0..*
#lastHit
0..1
#lastFired
#lastTriedUp
#nextPositions
#lastTriedDown
0..1
#protocol
0..1
-lastRead
-expecting
-packets
0..*
-parameters
-nonReadParameters
0..*
-type
0..1