shared::Cell

~numLayers: int
~maxP: Point
-grid[][]: Cell
+LAYER = 0: int
+OIL = 1: int
+GAS = 2: int
+ROCK = 3: int
-fudgeValue = 10: int
-gridSize: int
-removePercentage = 70: double
-layers: LithologicType[]
-surfaceDepth: int
+owner: String
+layers: Integer[]
+gas: Integer[]
+oil: Integer[]
+xCoord: int
+yCoord: int
+rocktype: LithologicType[]
+isDrilled: boolean
+isOwned: boolean
+seismicExists: boolean
+layerExists: boolean[]

+Grid(Point max, int layerCount): ctor
+Grid(): ctor
~Cell(int layerCount, int x, int y): ctor
~Cell(int x, int y): ctor
+getGridSize(): int
+setLimits(Point max, int layerCount): void
+initializeAll(): void
-initializeCell(int x, int y): void
+importCSV(File file, int level, int type): void
-parseCSV(CSVReader reader, int type, int level): void
+setGridSize(int size): void
+setOwner(String owner, Point loc): void
+setLayers(Point loc, Integer[] layers): void
+setAllLayers(int[] layers): void
+getAllLayers(): int[]
+setLayersSocket(String in): void
+setOil(Point loc, Integer[] oil): void
+setOilSocket(String in): void
+setGas(Point p, int gas, int depth): void
+setOil(Point p, int oil, int depth): void
+setLayer(Point p, Integer layer, int depth): void
+setRock(Point p, LithologicType rock, int depth): void
+setRock(Point loc, LithologicType[] rock): void
+setRock(Point loc, int[] rocks): void
+setGas(Point loc, Integer[] gas): void
+setGasSocket(String in): void
+setRockSocket(String in): void
+setDrilled(Point p, boolean in): void
+setDrilledSocket(String in): void
-getArrayFromString(String array): Integer[]
-getLayerArrayFromString(String array): Integer[]
-getRockArrayFromString(String array): LithologicType[]
-getExistsArrayFromString(String string): boolean[]
+addRock(LithologicType rock, int layer): void
+getGas(Point p, int depth): int
+getGasArray(Point p): Integer[]
+getGasSocket(Point p): String
+getOil(Point p, int depth): int
+getOilArray(Point p): Integer[]
+getOilSocket(Point p): String
+getLayer(Point p, int depth): Integer
+getLayerArray(Point p): Integer[]
+getSurfaceElev(Point p): int
+getSurfaceLayer(Point p): int
+incSurfaceLayer(): void
+decSurfaceLayer(): void
+goToSurface(): void
+getHighestSurface(): Point
+getLowestSurface(): Point
+getHighestSurfaceElev(): int
+getLowestSurfaceElev(): int
+getElevationRange(): int
+getLayersSocket(Point p): String
+getRockSocket(Point p): String
+getRockArray(): LithologicType[]
+getRockArray(Point p): LithologicType[]
+getRock(Point p, int depth): LithologicType
+exists(Point location): boolean
+isOwned(Point location): boolean
+isDrilled(Point location): boolean
+seismicExists(Point p): boolean
+setSeismicExists(Point p, boolean exists): void
+getOwner(Point p): String
+getDimensions(): Point
+getNumLayers(): int
+getGridSocket(): String[]
+cellToSocket(Point p): String
+cellFromSocket(String s): void
+fudge(Integer layer[]): Integer[]
+getGasRate(Point p): int
+getOilRate(Point p): int
+removeEmptyFromArray(Point p, Object[] array): String
+getLayerExists(Point p): boolean[]
+setLayerExists(Point p, boolean[] none): void
+layerExists(Point p, int depth): boolean
+setFudgePercent(int fudge): void
+setRemovePrecent(int parseInt): void

shared::Operator

#opGrid: Grid
#teamName: String
#bidQueue: LinkedList<Bid>
#seismicQueue: LinkedList<SeismicRequest>
#drillQueue: LinkedList<Drill>
#bank: BankAccount
+socket: Messages

+Operator(String teamName, Messages socket): ctor
+Operator(String teamName): ctor
+getName(): String
+setSocket(Messages s): void
+closeSocket(): void
+newBid(String socketIn): void
+newSeismicRequest(String socketIn): void
+newDrillRequest(String socketIn): void
+removeBid(String socketIn): void
+removeSeismicRequest(String socketIn): void
+removeDrillRequest(String socketIn): void
+removeFromQueue(Action remove, LinkedList<? extends Action> queue): void
+getBidQueue(): LinkedList<Bid>
+getSeismicQueue(): LinkedList<SeismicRequest>
+getDrillQueue(): LinkedList<Drill>
+resetBidQueue(): void
+resetSeismicQueue(): void
+resetDrillQueue(): void
+getBank(): BankAccount
+getBankBalance(): int
+setBankBalance(int amount): void
+updateBankBalance(int amount): void
+bidResult(Bid bid): void
+seismicRequestResult(String in): void
+oilInfo(String in): void
+gasInfo(String in): void
+rockInfo(String in): void
+drillRequestResult(Point p, Integer[] gas, Integer[] oil, LithologicType[] rock): void
+drillRequestResult(String in): void
+getGrid(): Grid
+toString(): String

shared.action::Bid

-bidAmount: int
-landP: Point
+noOwner = "Not Owned.": String
+won = false: boolean

+Bid(Operator owner, int bidAmount, Point land): ctor
+Bid(Operator operator, String socketIn): ctor
+Bid(String socketIn): ctor
-parseSocket(String[] socket): void
+checkValid(int bankLimit, Point maxP): boolean
+getPoint(): Point
+compareTo(Bid in): int
+equalsLand(Bid other): boolean
+equals(Object other): boolean
+toSocket(): String
+toString(): String
+setWon(): void
+won(): boolean

shared.action::SeismicRequest

-startP: Point
-endP: Point

+SeismicRequest(Operator owner, Point startP, Point endP): ctor
+SeismicRequest(Operator owner, String in): ctor
+checkValid(Point maxP): boolean
+toString(): String
+toSocket(): String
+getNumCells(): int
+getStartP(): Point
+getEndP(): Point
-getSeismicCost(): int
+getPoints(): Point[]
+equals(Object other): boolean

shared.action::Drill

-landP: Point
-depth: int

+Drill(Operator owner, Point p): ctor
+Drill(Operator owner, String socketIn): ctor
-parseSocket(String[] socket): void
+checkValid(Point p): boolean
+getData(Grid grid): String
+toSocket(): String
+getPoint(): Point
+equals(Object other): boolean
+toString(): String

shared.action::Event

-description: String

+Event(Operator owner): ctor
+setEventLogin(): void
+setEventLogout(): void
+setEventNewRound(int curRound): void
+toString(): String

shared.action::Action

#seismicSetup: int
#seismicRate: int
+drillCost: int
+oilRate: int
+gasRate: int
#minBid = 0: int
#owner: Operator
#timeStamp: Date
#beginBalance: int
#cost: int

+Action(Operator owner): ctor
+Action(): ctor
+setOwner(Operator owner): void
+setSeismicSetup(int amount): void
+setSeismicRate(int amount): void
+setMinBid(int amount): void
+getMinBid(): int
+getCost(): int
+setDrillCost(int amount): void
+setOilRate(int amount): void
+setGasRate(int amount): void
+getOwner(): Operator
+setSeismicCosts(String in): void
+seismicCostsToSocket(): String
+equals(Object other): boolean
+toString(): String
+toSocket(): String

shared::Messages

+TESTMODE = false: boolean
#sock: Socket
#sockOut: PrintWriter
#sockIn: BufferedReader
#IP: String
#operator: Operator
#name: String
-child: Messages
~prefixMap: Map<String, Prefix>

+Messages(Socket sock): ctor
-initializeSocket(Socket sock): void
-createPrefixMap(): void
+addChild(Messages child): void
+run(): void
-parse(String in): void
+parse(String in, Prefix pr): void
+sendMessage(Prefix p, String message): void
+close(): boolean
+setOperator(Operator operator): void

shared::BankAccount

-startingBalance: int
-curBalance: int
#unusableFunds: int
-income: int

+BankAccount(): ctor
+setStart(int start): void
+getStart(): int
+getAdjustedBalance(): int
+getBalance(): int
+updateFunds(int amount): void
+setBalance(int amount): void
+adjustIncome(int inIncome): void
+getIncome(): int
+changeUnusable(int amount): void
+resetUnusable(): void
+applyIncome(): void

shared::LithologicType

-rockCode: int
-fullName, shortName, imagePath: String
-abbrNames = { "Grv or Cgl (1)", "Grv or Cgl (2)", "x-bd Grv or Cgl", "", "Brec (1)", "Brec (2)", "Mass Sd. or Sst", "Bd Sd. or Sst", "x-bd Sd. or Sst (1)",
"x-bd Sd. or Sst(2)", "Rpl-Bd Sd. or Sst", "arg or sh Sst",
"calc Sst", "dol Sst", "", "Slt, Sltst, or sh Slt", "calc Sltst", "dol Sltst", "sdy or slty Sh", "Cl or Cl Sh", "cht Sh", "dol Sh", "calc Sh or marl", "carb Sh", "o Sh", "Chk", "Ls",
"clas Ls", "foss clas Ls", "nod or irr Bd Ls", "Ls, irr (Bur?) fillings of sacc Dol",
"x-bd Ls", "cht x-bd Ls",
"cht and sdy x-bd clas Ls", "ooc Ls", "sdy Ls", "slty Ls", "arg or sh Ls", "cht Ls (1)", "cht Ls (2)", "dol Ls or lmy Dol", "Dolst or Dol", "x-bd Dolst or Dol", "ooc Dolst or Dol",
"sdy Dolst or Dol", "slty Dolst or Dol",
"arg or sh Dolst or Dol", "cht Dolst or Dol", "Bd Cht (1)", "Bd Cht (2)",
"foss Bd Cht", "foss Rk", "Diatomaceous Rk", "Sbgwke", "x-bd Sbgwke", "Rpl-Bd Sbgwke", "Peat", "C", "Bony C or impure C", "Uc", "Flint Cl", "Bent", "Blauc", "Lim", "Sid.", "phos-nod Rk", "Gyp",
"Sa", "intbd Sst and Sltst", "intbd Sst and Sh",
"intbd RplBd Sst and Sh", "intbd Sh and slty Ls (Sh dom)", "intbd Sh and Ls (Sh dom) (1)",
"intbd Sh and Ls (Sh dom) (2)", "intbd calc Sh and Ls (Sh dom)", "intbd slty Ls and Sh", "intbd Ls and Sh (1)", "intbd Ls and Sh (2)", "intbd Ls and Sh (Ls dom)", "intbd Ls and calc Sh", "Till or diamicton (1)",
"Till or diamicton (2)", "Till or diamicton (3)", "Loess (1)", "Loess (2)", "Loess (3)", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "Meta", "Qtzt.", "Sl.", "sch or gns Grt", "Sch.", "cntrt Sch.",
"Sch. and Gns", "Gns", "cntrt Gns", "Soapstone, talc, or serpentinite", "tf Rk", "Xl Tf", "devit Tf", "volc Brec and Tf", "volc Brec or Aglm", "Zeolitic Rk", "bas flows", "Grt (1)", "Grt (2)", "bnd Ig", "Ig (1)",
"Ig (2)", "Ig (3)", "Ig (4)", "Ig (5)", "Ig (6)", "Ig (7)", "Ig (8)", "Porphyritic Rk (1)", "Porphyritic Rk (2)", "Vit", "Qtz.", "Ore" }: String[]
-fullNames = { "Gravel or conglomerate (1st option)", "Gravel or conglomerate (2nd option)", "Crossbedded gravel or conglomerate", "", "Breccia (1st option)", "Breccia (2nd option)", "Massive sand or sandstone",
"Bedded sand or sandstone", "Crossbedded sand or sandstone (1st option)", "Crossbedded sand or sandstone(2nd option)", "Ripple-bedded sand or sandstone", "Argillaceous or shaly sandstone", "Calcareous sandstone",
"Dolomitic sandstone", "Silt, siltstone, or shaly silt", "Calcareous siltstone", "Dolomitic siltstone", "", "Sandy or silty shale", "Clay or clay shale", "Cherty shale", "Dolomitic shale", "Calcareous shale or marl",
"Carbonaceous shale", "Oil shale", "Chalk", "Limestone", "Clastic limestone", "Fossiliferous clastic limestone", "Nodular or irregularly bedded limestone", "Limestone, irregular (burrow?) fillings of saccharoidal dolomite",
"Crossbedded limestone", "Cherty crossbedded limestone", "Cherty and sandy crossbedded clastic limestone", "Oolitic limestone", "Sandy limestone", "Silty limestone", "Argillaceous or shaly limestone", "Cherty limestone (1st option)",
"Cherty limestone (2nd option)", "Dolomitic limestone,limy dolostone, or limy dolomite", "Dolostone or dolomite", "Crossbedded dolostone ordolomite", "Oolitic dolostone or dolomite", "Sandy dolostone or dolomite", "Silty dolostone or dolomite",
"Argillaceous or shaly dolostone or dolomite", "Cherty dolostone or dolomite", "Bedded chert (1st option)", "Bedded chert (2nd option)", "Fossiliferous bedded chert", "Fossiliferous rock", "Diatomaceous rock", "Subgraywacke", "Crossbedded subgraywacke",
"Ripple-bedded subgraywacke", "Peat", "Coal", "Bony coal or impure coal", "Underclay", "Flint clay", "Bentonite", "Glauconite", "Limonite", "Siderite", "Phosphatic-nodular rock", "Gypsum", "Salt", "Interbedded sandstone and siltstone",
"Interbedded sandstone and shale", "Interbedded ripplebedded sandstone and shale", "Interbedded shale and silty limestone (shale dominant)", "Interbedded shale and limestone (shale dominant) (1st option)", "Interbedded shale and limestone (shale dominant) (2nd option)",
"Interbedded calcareous shale and limestone (shale dominant)", "Interbedded silty limestone and shale", "Interbeddedlimestone and shale (1st option)", "Interbedded limestone and shale (2nd option)", "Interbedded limestone and shale (limestone dominant)", "Interbedded limestone and calcareous shale",
"Till or diamicton (1st option)", "Till or diamicton (2nd option)", "Till or diamicton (3rd option)", "Loess (1st option)", "Loess (2nd option)", "Loess (3rd option)", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "Metamorphism", "Quartzite", "Slate", "Schistose or gneissoid granite",
"Schist", "Contorted schist", "Schist and gneiss", "Gneiss", "Contorted gneiss", "Soapstone, talc, or serpentinite", "Tuffaceous rock", "Crystal tuff", "Devitrified tuff", "Volcanic breccia and tuff", "Volcanic breccia or agglomerate", "Zeolitic rock", "Basaltic flows", "Granite (1st option)",
"Granite (2nd option)", "Banded igneous rock", "Igneous rock (1st option)", "Igneous rock (2nd option)", "Igneous rock (3rd option)", "Igneous rock (4th option)", "Igneous rock (5th option)", "Igneous rock (6th option)", "Igneous rock (7th option)", "Igneous rock (8th option)", "Porphyritic rock (1st option)",
"Porphyritic rock (2nd option)", "Vitrophyre", "Quartz", "Ore" }: String[]

+LithologicType(int rockCode): ctor
+getRockCode(): int
+getFullName(): String
+getShortName(): String
+imagePath(): String
+toString(): String
+toArrayString(LithologicType[] lithRocks): String[]