Opcionable
+ removerOpcionableExtra(): void
+ getOpcionables(): List<String>
+ ejecutar(): void
+ getOpcionableExtra(): String
<<interface>>
Propiedad
+ getGrupo(): String
+ venderPropiedad(Saldo): void
+ desHipotecar(Saldo): void
+ cambiarDisponibilidad(Disponibilidad): void
+ rematar(): void
+ hipotecar(Saldo): void
+ calcularPeaje(): double
OpcionableLibre
+ ejecutar(): void
VenderHotelUseCase
+ ejecutar(): void
VenderCasaUseCase
+ ejecutar(): void
DesHipotecarUseCase
+ ejecutar(): void
ConstruirHotelUseCase
+ ejecutar(): void
VenderPropiedadUseCase
+ ejecutar(): void
HipotecarUseCase
+ ejecutar(): void
NoVendible
ConstruirCasaUseCase
+ ejecutar(): void
<<interface>>
Disponibilidad
+ hipotecar(): void
+ desHipotecar(Saldo, double): void
+ interactuarCon(Jugador, Propiedad, double): void
+ venderPropiedad(): void
+ lePerteneceA(Jugador): boolean
+ construirCasa(): void
ComprarPropiedadUseCase
+ ejecutar(): void
NoDesHipotecable
NoSePuedeConstruir
NoHipotecable
Posicion
+ hashCode(): int
+ equals(Object): boolean
<<interface>>
Edificio
+ getPeaje(): double
DataCasilla
PagarPeajeUseCase
+ ejecutar(): void
Hipotecada
+ interactuarCon(Jugador, Propiedad, double): void
+ venderPropiedad(): void
+ hipotecar(): void
+ construirCasa(): void
+ desHipotecar(Saldo, double): void
+ lePerteneceA(Jugador): boolean
<<interface>>
Semaforo
+ construirCasa(Terreno, Saldo): void
+ venderCasa(Terreno, Saldo): void
EnVenta
+ lePerteneceA(Jugador): boolean
+ desHipotecar(Saldo, double): void
+ interactuarCon(Jugador, Propiedad, double): void
+ construirCasa(): void
+ hipotecar(): void
+ venderPropiedad(): void
Casa
+ getPeaje(): double
Hotel
+ getPeaje(): double
Casilla
+ getPosicion(): Posicion
+ getDataCasilla(): DataCasilla
+ interactuarCon(Jugador, Tablero): void
PagarMultaUseCase
+ ejecutar(): void
Rojo
+ venderCasa(Terreno, Saldo): void
+ construirCasa(Terreno, Saldo): void
Loteria
+ interactuarCon(Jugador, Tablero): void
Disponible
+ hipotecar(): void
+ desHipotecar(Saldo, double): void
+ construirCasa(): void
+ interactuarCon(Jugador, Propiedad, double): void
+ lePerteneceA(Jugador): boolean
+ venderPropiedad(): void
Verde
+ construirCasa(Terreno, Saldo): void
+ venderCasa(Terreno, Saldo): void
<<interface>>
JuegoObserver
+ notificarQuiebra(): void
<<interface>>
Estado
+ ejecutar(int): int
Transporte
+ cambiarDisponibilidad(Disponibilidad): void
+ venderPropiedad(Saldo): void
+ hipotecar(Saldo): void
+ calcularPeaje(): double
+ rematar(): void
+ equals(Object): boolean
+ desHipotecar(Saldo): void
+ getGrupo(): String
+ interactuarCon(Jugador, Tablero): void
Terreno
+ cambiarDisponibilidad(Disponibilidad): void
+ venderCasa(Saldo): void
+ getGrupo(): String
+ lePerteneceA(Jugador): boolean
+ venderPropiedad(Saldo): void
+ getEdificios(): List<Edificio>
+ desHipotecar(Saldo): void
+ construirHotel(Saldo): void
+ interactuarCon(Jugador, Tablero): void
+ rematar(): void
+ calcularPeaje(): double
+ construirCasa(Saldo): void
+ hipotecar(Saldo): void
+ venderHotel(Saldo): void
Multa
+ interactuarCon(Jugador, Tablero): void
LLegadaPartida
+ interactuarCon(Jugador, Tablero): void
DePaso
+ interactuarCon(Jugador, Tablero): void
SaldoNoDisponible
Saldo
+ invertir(double): void
+ pagar(double): void
+ agregarObserver(JuegoObserver): void
+ recibir(double): void
+ pagarSinObligacion(double): void
+ desInvertir(double): void
AdministradorDePropiedades
+ tieneTransporte(): boolean
+ calcularPeajePropiedades(Propiedad): double
+ agregarPropiedad(Propiedad, Jugador): void
+ hipotecar(Propiedad, Saldo): void
- obtenerYVerificarQuePropiedadesContieneA(Propiedad): List<Propiedad>
- obtenerListaQueContieneAlTipoDe(Propiedad): List<Propiedad>
+ desHiptecar(Propiedad, Saldo): void
+ venderCasa(Terreno, Saldo): void
+ venderPropiedad(Propiedad, Saldo): void
+ rematar(): void
- propiedadesContieneElTipoDe(Propiedad): boolean
+ venderHotel(Terreno, Saldo): void
+ construirCasaSobre(Terreno, Saldo): void
+ construirHotelSobre(Terreno, Saldo): void
Libre
+ ejecutar(int): int
Dado
+ tirar(): int
Jugador
+ comprarPropiedad(Propiedad, double): void
+ hipotecarPropiedad(Propiedad): void
+ recibirPago(double): void
+ ejecutarOpcionable(): void
+ setEstado(Estado): void
+ tieneTransporte(): boolean
+ getPosicion(): Posicion
+ construirCasaEn(Terreno): void
+ pagarFianza(double): void
+ agregarOpcionable(Opcionable): void
+ construirHotelEn(Terreno): void
+ tirarDado(): int
+ getOpcionable(): Opcionable
+ setPosicion(Posicion): void
+ getNombre(): String
+ venderCasa(Terreno): void
+ rematarPropiedades(): void
+ desHipotecar(Terreno): void
+ pagar(double): void
+ venderPropiedad(Propiedad): void
+ agregarJuegoObserver(JuegoObserver): void
+ venderHotel(Terreno): void
+ calcularPeajeTerrenos(Propiedad): double
Banco
+ pagarRondaCompletaA(Jugador): void
Carcel
+ equals(Object): boolean
+ interactuarCon(Jugador, Tablero): void
OpcionablePreso
+ ejecutar(): void
Quebrado
+ ejecutar(int): int
GestorDeTurno
+ turnoDe(): Jugador
+ eliminarJugador(Jugador): void
Tablero
+ mover(Jugador, int): void
- interactuarConCasilla(Jugador, int): void
+ moverACarcel(Jugador): void
+ getCasillasEn(Posicion): Casilla
+ getJugadores(): List<Jugador>
OpcionableQuebrado
+ ejecutar(): void
Preso
+ ejecutar(int): int
Juego
+ ejecutarOpcionableDelJugadorDeTurno(): void
+ hayGanador(): boolean
+ cambiarTurno(): void
+ iniciarPartida(): void
+ getTablero(): Tablero
+ ejecutarTurno(): void
+ eliminarJugadorQuebrado(): void
+ notificarQuiebra(): void
+ agregarJugadores(List<Jugador>): void
+ getJugadorEnTurno(): Jugador
EncarcelarUseCase
+ ejecutar(): void
Encarcelar
+ interactuarCon(Jugador, Tablero): void
CasillaView
+ insertarJugador(JugadorView): void
+ update(Jugador): void
+ removerJugador(JugadorView): void
AlertDialogo
JugadorView
+ getJugador(): Jugador
+ es(Jugador): boolean
+ getColor(): Color
LoteriaView
+ update(Jugador): void
EnCarcelarView
+ update(Jugador): void
TransporteView
+ update(Jugador): void
View
# configurarBackground(VBox): void
+ setDimensions(double, double): void
# configurarBoton(Button): void
# configurarTitulo(Label, String, int): void
TerrenoView
+ updateConstrucciones(Jugador): void
+ actualizarVistaPropiedad(JugadorView): void
+ mostrarBotonesElegibles(Jugador, String): boolean
+ update(Jugador): void
+ updateConstruccionesTerrenos(): void
+ quitarBotonCompra(): void
CarcelView
+ update(Jugador): void
DePasoView
+ update(Jugador): void
MultaView
+ update(Jugador): void
LlegadaPartidaView
+ update(Jugador): void
Controller
+ handle(ActionEvent): void
+ verificarGanador(): void
BotonTirarDadosHandler
+ handle(ActionEvent): void
ButtonTerminaTurno
+ handle(ActionEvent): void
TableroView
+ actualizarCasillas(Jugador): void
+ insertarTexto(): void
+ quitarOpcionable(): void
- mostrarOpcionableExtra(Jugador): void
+ mostrarAceptarCancelar(Jugador): void
+ crearCasilla(Casilla): CasillaView
+ eliminarJugadorView(Jugador): void
+ actualizarVista(Posicion, Jugador): void
- quitarBotonesCompra(): void
+ quitarOpcionableExtra(): void
+ mostrarCasillaElegible(Jugador, String): void
+ insertarTextoEnVBox(): void
+ getScene(): Scene
+ mostrarOpcionable(Jugador): void
- quitarBotonesElegibles(int, int): void
+ construir(): void
Configuracion
BotonIniciarTableroHandler
+ handle(ActionEvent): void
JugadoresView
- configurarTitulo(): void
- configurarBotonComenzar(): void
- configurarInputs(int): void
- configurarTextField(TextField): void
+ getScene(): Scene
BotonCantidadJugadorHandler
+ handle(ActionEvent): void
Logger
+ getInstance(): Logger
+ error(String): void
+ mostrarInfo(): String
+ info(String): void
SeleccionView
+ configurarBotones(HBox): void
+ getScene(): Scene
+ configurarTexto(): void
+ configurarContenido(): void
BotonIniciarSeleccionHandler
+ handle(ActionEvent): void
BotonSalirHandler
+ handle(ActionEvent): void
InicioView
- configurarBotones(): void
- configurarTituloPrincipal(): void
+ getScene(): Scene
App
+ main(String[]): void
- configurarVentana(Stage): void
+ start(Stage): void
PropiadNoPerteneceAJugador
Main
+ main(String[]): void
NoComprable
«create»