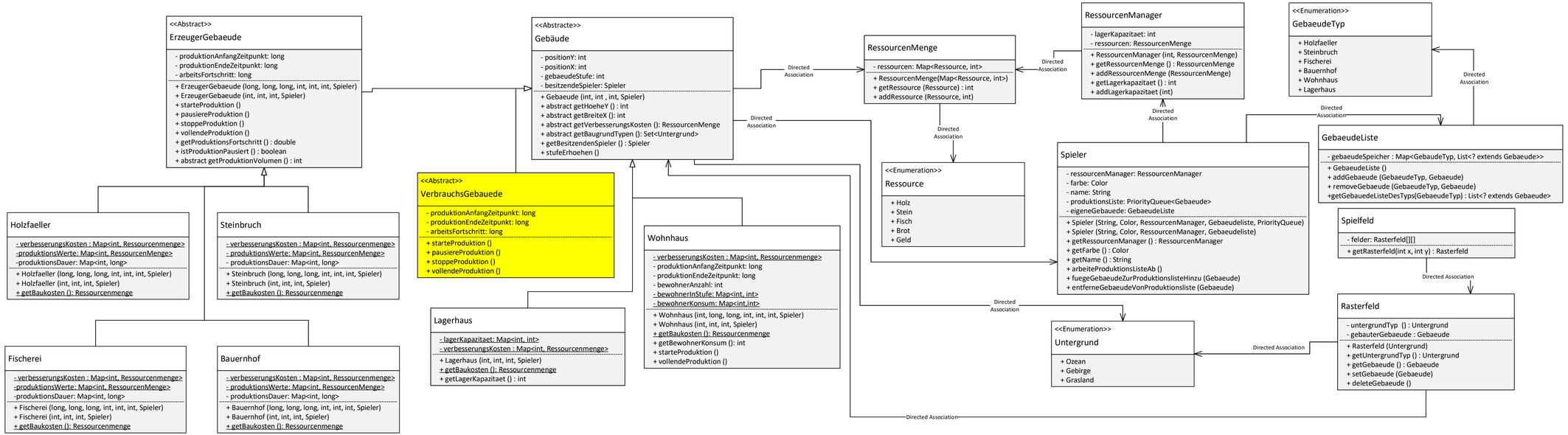
<<Abstract>>
ErzeugerGebaeude
- produktionAnfangZeitpunkt: long
- produktionEndeZeitpunkt: long
- arbeitsFortschritt: long
+ ErzeugerGebaeude (long, long, long, int, int, int, Spieler)
+ ErzeugerGebaeude (int, int, int, Spieler)
+ starteProduktion ()
+ pausiereProduktion ()
+ stoppeProduktion ()
+ vollendeProduktion ()
+ getProduktionsFortschritt () : double
+ istProduktionPausiert () : boolean
+ abstract getProduktionVolumen () : int
<<Abstracte>>
Gebäude
- positionY: int
- positionX: int
- gebaeudeStufe: int
- besitzendeSpieler: Spieler
+ Gebaeude (int, int , int, Spieler)
+ abstract getHoeheY () : int
+ abstract getBreiteX (): int
+ abstract getVerbesserungsKosten (): RessourcenMenge
+ abstract getBaugrundTypen (): Set<Untergrund>
+ getBesitzendenSpieler () : Spieler
+ stufeErhoehen ()
Directed Association
RessourcenMenge
- ressourcen: Map<Ressource, int>
+ RessourcenMenge(Map<Ressource, int>)
+ getRessource (Ressource) : int
+ addRessource (Ressource, int)
Directed Association
RessourcenManager
- lagerKapazitaet: int
- ressourcen: RessourcenMenge
+ RessourcenManager (int, RessourcenMenge)
+ getRessourcenMenge () : RessourcenMenge
+ addRessourcenMenge (RessourcenMenge)
+ getLagerkapazitaet () : int
+ addLagerkapazitaet (int)
<<Enumeration>>
GebaeudeTyp
+ Holzfaeller
+ Steinbruch
+ Fischerei
+ Bauernhof
+ Wohnhaus
+ Lagerhaus
Directed Association
Directed Association
Directed Association
Directed Association
Directed Association
<<Enumeration>>
Ressource
+ Holz
+ Stein
+ Fisch
+ Brot
+ Geld
Spieler
- ressourcenManager: RessourcenManager
- farbe: Color
- name: String
- produktionsListe: PriorityQueue<Gebaeude>
- eigeneGebaeude: GebaeudeListe
+ Spieler (String, Color, RessourcenManager, Gebaeudeliste, PriorityQueue)
+ Spieler (String, Color, RessourcenManager, Gebaeudeliste)
+ getRessourcenManager () : RessourcenManager
+ getFarbe () : Color
+ getName () : String
+ arbeiteProduktionsListeAb ()
+ fuegeGebaeudeZurProduktionslisteHinzu (Gebaeude)
+ entferneGebaeudeVonProduktionsliste (Gebaeude)
GebaeudeListe
- gebaeudeSpeicher : Map<GebaudeTyp, List<? extends Gebaeude>>
+ GebaeudeListe ()
+ addGebaeude (GebaeudeTyp, Gebaeude)
+ removeGebaeude (GebaeudeTyp, Gebaeude)
+getGebaeudeListeDesTyps(GebaeudeTyp) : List<? extends Gebaeude>
Spielfeld
- felder: Rasterfeld[][]
+ getRasterfeld(int x, int y) : Rasterfeld
Directed Association
Rasterfeld
- untergrundTyp () : Untergrund
- gebauterGebaeude : Gebaeude
+ Rasterfeld (Untergrund)
+ getUntergrundTyp () : Untergrund
+ getGebaeude () : Gebaeude
+ setGebaeude (Gebaeude)
+ deleteGebaeude ()
Directed Association
<<Enumeration>>
Untergrund
+ Ozean
+ Gebirge
+ Grasland
Directed Association
Directed Association
<<Abstract>>
VerbrauchsGebauede
- produktionAnfangZeitpunkt: long
- produktionEndeZeitpunkt: long
- arbeitsFortschritt: long
+ starteProduktion ()
+ pausiereProduktion ()
+ stoppeProduktion ()
+ vollendeProduktion ()
Holzfaeller
- verbesserungsKosten : Map<int, Ressourcenmenge>
-produktionsWerte: Map<int, RessourcenMenge>
-produktionsDauer: Map<int, long>
+ Holzfaeller (long, long, long, int, int, int, Spieler)
+ Holzfaeller (int, int, int, Spieler)
+ getBaukosten (): Ressourcenmenge
Steinbruch
- verbesserungsKosten : Map<int, Ressourcenmenge>
- produktionsWerte: Map<int, RessourcenMenge>
- produktionsDauer: Map<int, long>
+ Steinbruch (long, long, long, int, int, int, Spieler)
+ Steinbruch (int, int, int, Spieler)
+ getBaukosten (): Ressourcenmenge
Wohnhaus
- verbesserungsKosten : Map<int, Ressourcenmenge>
- produktionAnfangZeitpunkt: long
- produktionEndeZeitpunkt: long
- bewohnerAnzahl: int
- bewohnerInStufe: Map<int, int>
- bewohnerKonsum: Map<int,int>
+ Wohnhaus (int, long, long, int, int, int, Spieler)
+ Wohnhaus (int, int, int, Spieler)
+ getBaukosten (): Ressourcenmenge
+ getBewohnerKonsum (): int
+ starteProduktion ()
+ vollendeProduktion ()
Lagerhaus
- lagerKapazitaet: Map<int, int>
- verbesserungsKosten : Map<int, Ressourcenmenge>
+ Lagerhaus (int, int, int, Spieler)
+ getBaukosten (): Ressourcenmenge
+ getLagerKapazitaet () : int
Fischerei
- verbesserungsKosten : Map<int, Ressourcenmenge>
-produktionsWerte: Map<int, RessourcenMenge>
-produktionsDauer: Map<int, long>
+ Fischerei (long, long, long, int, int, int, Spieler)
+ Fischerei (int, int, int, Spieler)
+ getBaukosten (): Ressourcenmenge
Bauernhof
- verbesserungsKosten : Map<int, Ressourcenmenge>
- produktionsWerte: Map<int, RessourcenMenge>
- produktionsDauer: Map<int, long>
+ Bauernhof (long, long, long, int, int, int, Spieler)
+ Bauernhof (int, int, int, Spieler)
+ getBaukosten (): Ressourcenmenge