

- resources

+ addToCh

+ subFrom

- resources - resources

+ getFromS

+ addResou + setTypeO

+ controlSto

+ addResou

+ defaultAd

+ addResou

+ discardRe

+ totResour

+ totalReso + subFromS

+ subFrom(

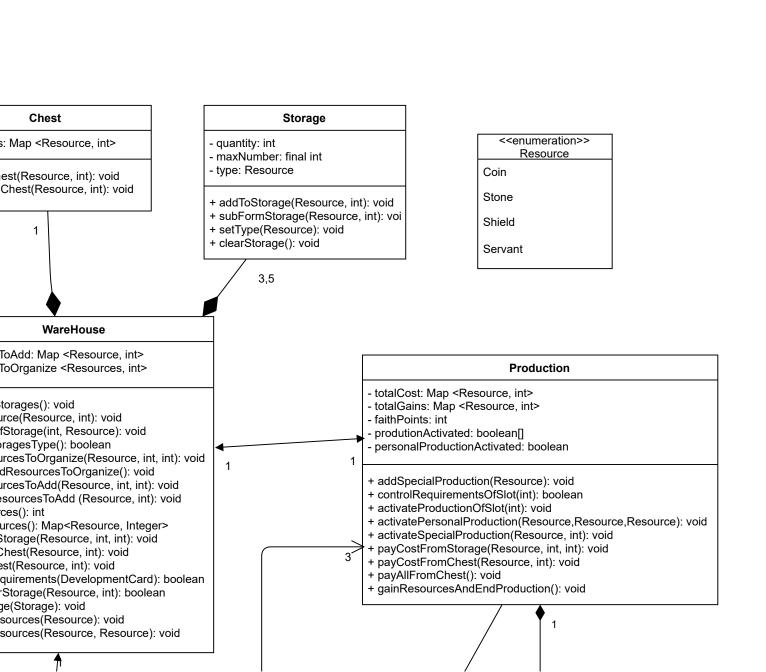
+ addToChe

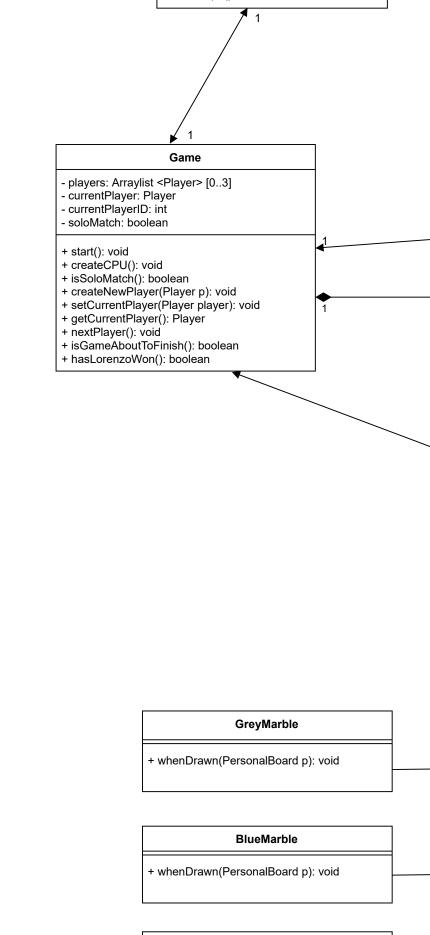
+ controlRe + controlFo

+ addStorag

+ addInitRe

+ addInitRe



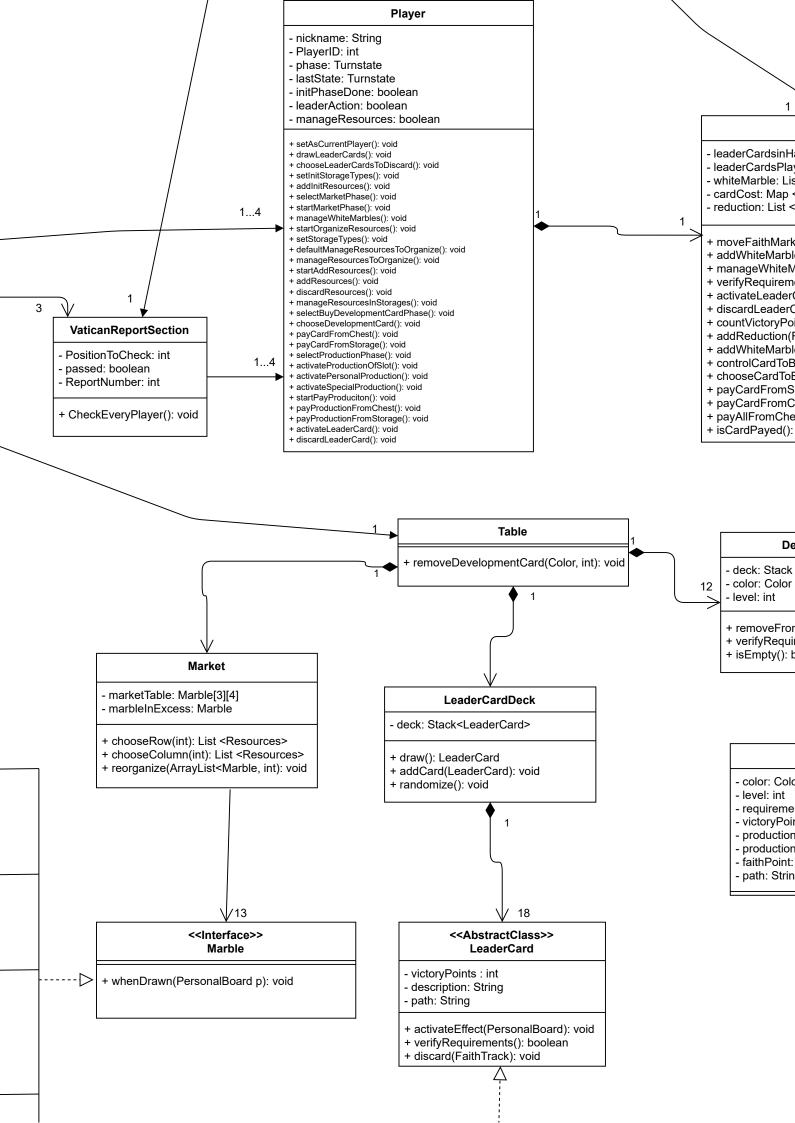


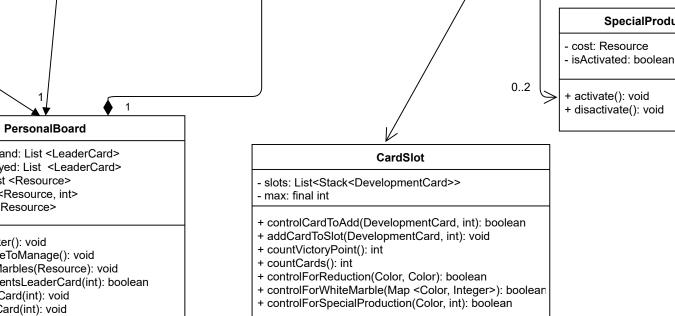
PurpleMarble

RedMarble

+ whenDrawn(PersonalBoard p): void

+ whenDrawn(PersonalBoard p): void





<<Enumeration>>

Color

Green

Yellow

Purple

Blu

velopmentCardDeck

ints(): int Resource): void e(Resource): void

boolean

<DevelopmentCards>

nTop(): void rement(): Map<Resource, Integer; oolean

uy(DevelopmentCard, int): boolean Buy(DevelopmentCard, int): void

torage(Resource, int, int): void

hest(Resource, int): void

st(Resource, int): void



DevelopmentCard

nt: Map <Resource, int> nt: int

g

Input: Map <Resource, int> Output: Map <Resource, int> int

48

<<Enumeration>> **TurnState**

SpecialProduction

BeforeStart Wait Preparation ChooseLeaderCards ChooseResources EndPreparation Start MarketPhase WhiteMarbles Choice OrganizeResources ManageResources AddResources BuyDevelopmentCardPhase PayDevelopmentCard ProductionPhase PayProductions

EndTurn

ViewOtherPlayers

+ whenDrawn(PersonalBoard p): void
WhiteMarble
+ whenDrawn(PersonalBoard p): void

YellowMarble

LeaderCardReduction

- color1: Color - color2: Color

- reduction: Resource

+ activateEffect(PersonalBoard): void + verifyRequirements(): boolean

LeaderCardStorage

- requirement: Resource - type: Resource

+ activateEffect(PersonalBoard): void + verifyRequirements(): boolean

LeaderCardWhiteMarble

requirement: Map <Color, int>resource: Resource

+ activateEffect(PersonalBoard): v + verifyRequirements(): boolean

${\bf Leader Card Special Production}$

- color: Color - cost: Resource

oid

+ activateEffect(PersonalBoard): void + verifyRequirements(): boolean