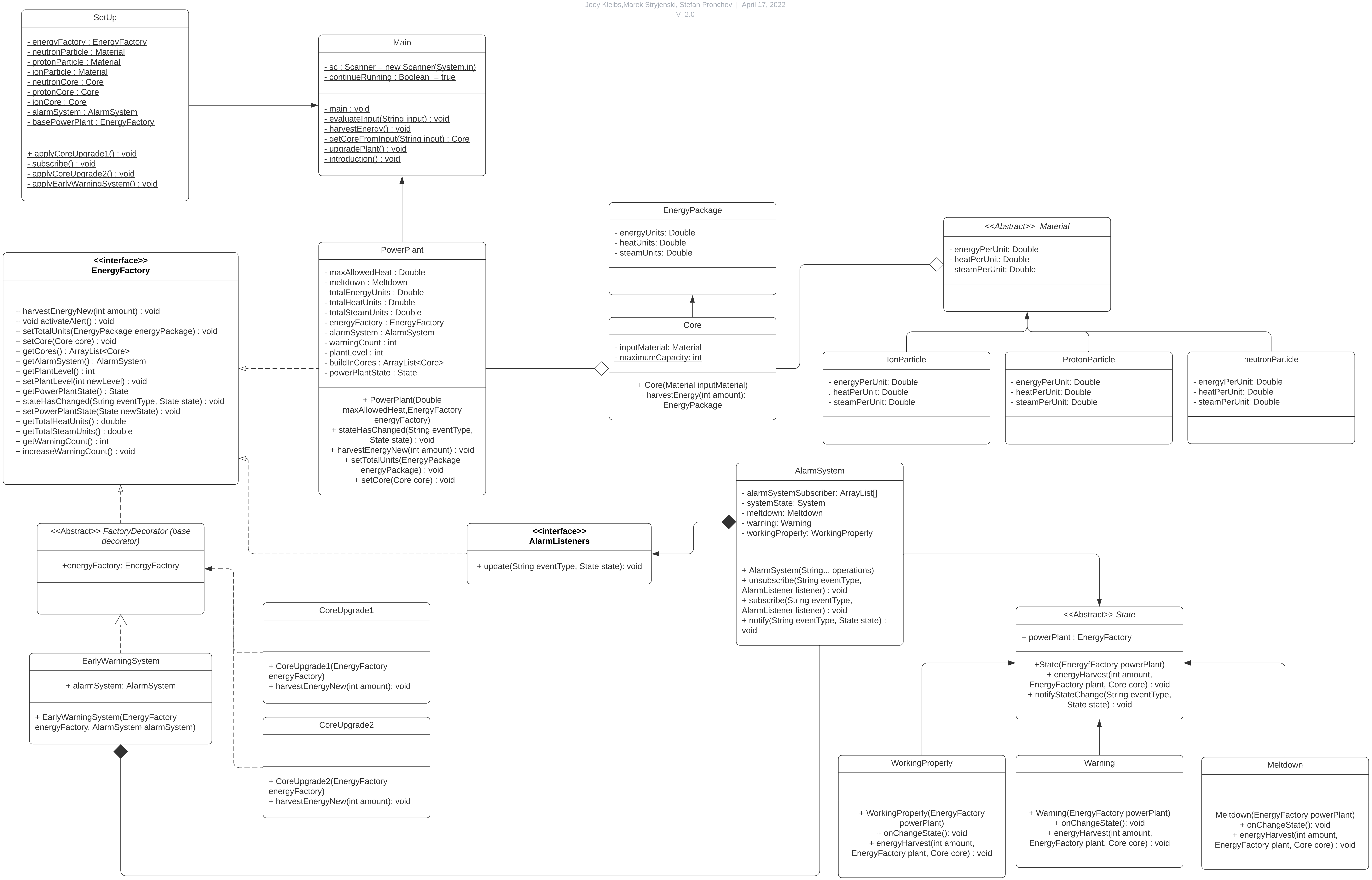
Power Plant design patterns UML

Joey Kleibs,Marek Stryjenski, Stefan Pronchev | April 17, 2022
V_2.0



SetUp

- energyFactory : EnergyFactory
- neutronParticle : Material
- protonParticle : Material
- ionParticle : Material
- neutronCore : Core
- protonCore : Core
- ionCore : Core
- alarmSystem : AlarmSystem
- basePowerPlant : EnergyFactory

+ applyCoreUpgrade1() : void
- subscribe() : void
- applyCoreUpgrade2() : void
- applyEarlyWarningSystem() : void

Main

- sc : Scanner = new Scanner(System.in)
- continueRunning : Boolean = true

- main : void
- evaluateInput(String input) : void
- harvestEnergy() : void
- getCoreFromInput(String input) : Core
- upgradePlant() : void
- introduction() : void

EnergyPackage

- energyUnits: Double
- heatUnits: Double
- steamUnits: Double

<<Abstract>> *Material*

- energyPerUnit: Double
- heatPerUnit: Double
- steamPerUnit: Double

<<interface>>
EnergyFactory

+ harvestEnergyNew(int amount) : void
+ void activateAlert() : void
+ setTotalUnits(EnergyPackage energyPackage) : void
+ setCore(Core core) : void
+ getCores() : ArrayList<Core>
+ getAlarmSystem() : AlarmSystem
+ getPlantLevel() : int
+ setPlantLevel(int newLevel) : void
+ getPowerPlantState() : State
+ stateHasChanged(String eventType, State state) : void
+ setPowerPlantState(State newState) : void
+ getTotalHeatUnits() : double
+ getTotalSteamUnits() : double
+ getWarningCount() : int
+ increaseWarningCount() : void

PowerPlant

- maxAllowedHeat : Double
- meltdown : Meltdown
- totalEnergyUnits : Double
- totalHeatUnits : Double
- totalSteamUnits : Double
- energyFactory : EnergyFactory
- alarmSystem : AlarmSystem
- warningCount : int
- plantLevel : int
- buildInCores : ArrayList<Core>
- powerPlantState : State

+ PowerPlant(Double maxAllowedHeat,EnergyFactory energyFactory)
+ stateHasChanged(String eventType, State state) : void
+ harvestEnergyNew(int amount) : void
+ setTotalUnits(EnergyPackage energyPackage) : void
+ setCore(Core core) : void

Core

- inputMaterial: Material
- maximumCapacity: int

+ Core(Material inputMaterial)
+ harvestEnergy(int amount): EnergyPackage

IonParticle

- energyPerUnit: Double
. heatPerUnit: Double
- steamPerUnit: Double

ProtonParticle

- energyPerUnit: Double
- heatPerUnit: Double
- steamPerUnit: Double

neutronParticle

- energyPerUnit: Double
- heatPerUnit: Double
- steamPerUnit: Double

AlarmSystem

- alarmSystemSubscriber: ArrayList[]
- systemState: System
- meltdown: Meltdown
- warning: Warning
- workingProperly: WorkingProperly

+ AlarmSystem(String... operations)
+ unsubscribe(String eventType, AlarmListener listener) : void
+ subscribe(String eventType, AlarmListener listener) : void
+ notify(String eventType, State state) : void

<<Abstract>> *FactoryDecorator (base decorator)*

+energyFactory: EnergyFactory

<<interface>>
AlarmListeners

+ update(String eventType, State state): void

CoreUpgrade1

+ CoreUpgrade1(EnergyFactory energyFactory)
+ harvestEnergyNew(int amount): void

<<Abstract>> *State*

+ powerPlant : EnergyFactory

+State(EnergyfFactory powerPlant)
+ energyHarvest(int amount, EnergyFactory plant, Core core) : void
+ notifyStateChange(String eventType, State state) : void

EarlyWarningSystem

+ alarmSystem: AlarmSystem

+ EarlyWarningSystem(EnergyFactory energyFactory, AlarmSystem alarmSystem)

CoreUpgrade2

+ CoreUpgrade2(EnergyFactory energyFactory)
+ harvestEnergyNew(int amount): void

WorkingProperly

+ WorkingProperly(EnergyFactory powerPlant)
+ onChangeState(): void
+ energyHarvest(int amount, EnergyFactory plant, Core core) : void

Warning

+ Warning(EnergyFactory powerPlant)
+ onChangeState(): void
+ energyHarvest(int amount, EnergyFactory plant, Core core) : void

Meltdown

Meltdown(EnergyFactory powerPlant)
+ onChangeState(): void
+ energyHarvest(int amount, EnergyFactory plant, Core core) : void