Grid

- totalProduction: int
- totalConsumption: int
- height: int
- width: int

+ getWidth(): int
+ getHeight(): int
+ registerItem(int, int, AbstractItem): void
+ getItem(int, int): AbstractItem
+ getStockAt(int, int): int
+ emptyStockAt(int, int)
+ addToStockAt(int, int, int): void
+ reduceStockAt(int, int, int): void
+ setStockAt(int, int, int): void
+ processItems(Timestep): void
+ recordProduction(int): void
+ getTotalProduction(): int
+ recordConsumption(int): void
+ getTotalConsumption(): int
+ checkConsumer(int, int): boolean
+ checkProducer(int, int): boolean
+ checkTransporter(int, int): boolean
+ processProducers(Timestep): void
+ processConsumers(Timestep): void
+ processTransporters(Timestep): void

AbstractGrid

grid: AbstractItem[][]
stock: int[][]

+ *getWidth(): int*
+ *getHeight(): int*
+ *registerItem(int, int, AbstractItem): void*
+ *getItem(int, int): AbstractItem*
+ *getStockAt(int, int): int*
+ *emptyStockAt(int, int)*
+ *addToStockAt(int, int, int): void*
+ *reduceStockAt(int, int, int): void*
+ *setStockAt(int, int, int): void*
+ *processItems(Timestep): void*
+ *recordProduction(int): void*
+ *getTotalProduction(): int*
+ *recordConsumption(int): void*
+ *getTotalConsumption(): int*
+ *display(): void*
- *padCenter(String, int): String*

Rabbit

- capacity: int

+ *process(Timestep): void*
+ *getStock(): int*
+ *addToStock(int): void*
+ *reduceStock(int): void*
+ *toString():String*

Beaver

- capacity: int
- storarge: int

+ *process(Timestep): void*
+ *getStock(): int*
+ *addToStock(int): void*
+ *reduceStock(int): void*
+ *toString():String*

Game

- Grid: AbstractGrid

+ Game(AbstractGrid)
+ run(int): void

Farmer

AbstractItem

grid: AbstractGrid
xCoordinate: int
yCoordinate: int

+ *process(Timestep): void*
+ *getStock(): int*
+ *addToStock(int): void*
+ *reduceStock(int): void*

HorizontalTransporter

~ capacity:int
~ leftX: int
~ leftY: int
~ rightX: int
~ rightY: int
~ leftType: String
~ rightType: String

+ process(Timestep): void
+ getStock(): int
+ addToStock(int): void
+ reduceStock(int): void
+ toString():String
+ checkTransportStatus(): boolean
+ getLeft(): void
+ getRight(): void

TimeStep

- ticksFromStart: int

+ Timestep()
+ increment(): void
+ getValue(): int

CornFarmer

~ produce: int
~ nutritionPerProduce: int
~ produceTime: int

+ process(Timestep): void
+ getStock(): int
+ addToStock(int): void
+ reduceStock(int): void
+ toString(): String
~ checkLeft(): boolean
~ checkRight(): boolean
~ checkUp(): boolean
~ checkDown(): boolean
~ checkAll(): boolean

RadishFarmer

~ produce: int
~ nutritionPerProduce: int
~ produceTime: int

+ process(Timestep): void
+ getStock(): int
+ addToStock(int): void
+ reduceStock(int): void
+ toString(): String
~ checkLeft(): boolean
~ checkRight(): boolean
~ checkUp(): boolean
~ checkDown(): boolean
~ checkAll(): boolean

VerticalTransporter

~ capacity:int
~ topX: int
~ topY: int
~ bottomX: int
~ bottomY: int

+ process(Timestep): void
+ getStock(): int
+ addToStock(int): void
+ reduceStock(int): void
+ toString():String
+ checkTransportStatus(): boolean
+ getTop(): void
+ getBottom(): void