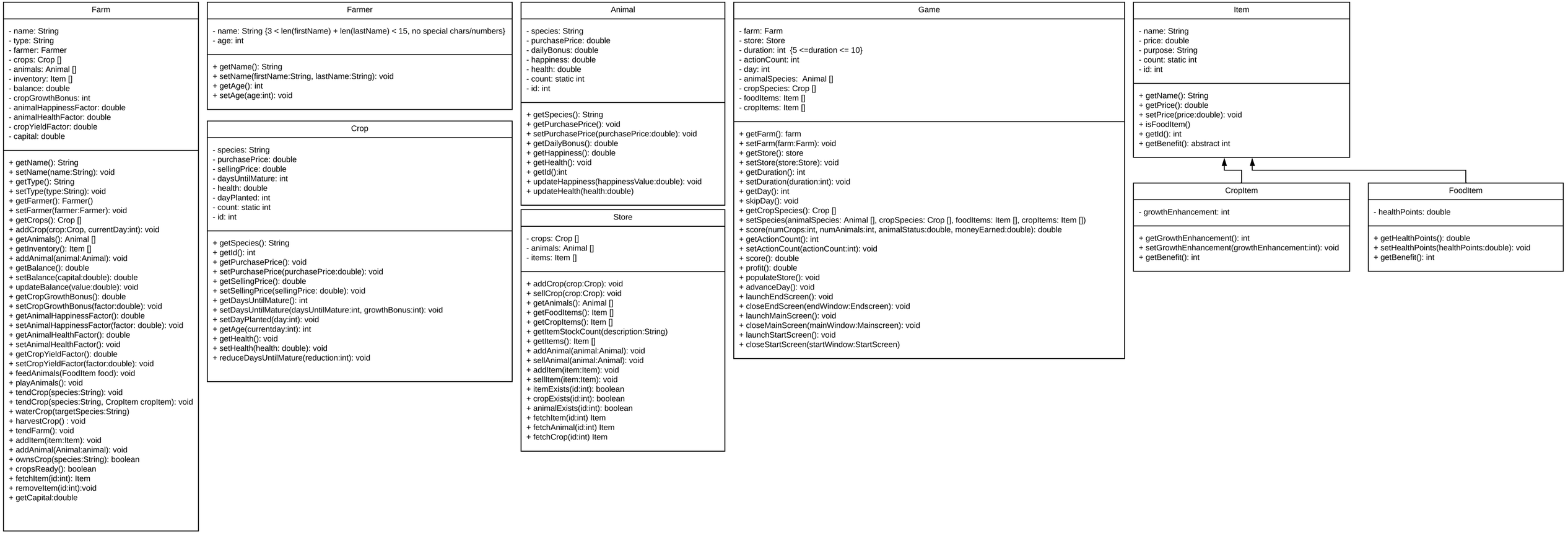
Farm
- name: String
- type: String
- farmer: Farmer
- crops: Crop []
- animals: Animal []
- inventory: Item []
- balance: double
- cropGrowthBonus: int
- animalHappinessFactor: double
- animalHealthFactor: double
- cropYieldFactor: double
- capital: double
+ getName(): String
+ setName(name:String): void
+ getType(): String
+ setType(type:String): void
+ getFarmer(): Farmer()
+ setFarmer(farmer:Farmer): void
+ getCrops(): Crop []
+ addCrop(crop:Crop, currentDay:int): void
+ getAnimals(): Animal []
+ getInventory(): Item []
+ addAnimal(animal:Animal): void
+ getBalance(): double
+ setBalance(capital:double): double
+ updateBalance(value:double): void
+ getCropGrowthBonus(): double
+ setCropGrowthBonus(factor:double): void
+ getAnimalHappinessFactor(): double
+ setAnimalHappinessFactor(factor: double): void
+ getAnimalHealthFactor(): double
+ setAnimalHealthFactor(): void
+ getCropYieldFactor(): double
+ setCropYieldFactor(factor:double): void
+ feedAnimals(FoodItem food): void
+ playAnimals(): void
+ tendCrop(species:String): void
+ tendCrop(species:String, CropItem cropItem): void
+ waterCrop(targetSpecies:String)
+ harvestCrop() : void
+ tendFarm(): void
+ addItem(item:Item): void
+ addAnimal(Animal:animal): void
+ ownsCrop(species:String): boolean
+ cropsReady(): boolean
+ fetchItem(id:int): Item
+ removeItem(id:int):void
+ getCapital:double
Farmer
- name: String {3 < len(firstName) + len(lastName) < 15, no special chars/numbers}
- age: int
+ getName(): String
+ setName(firstName:String, lastName:String): void
+ getAge(): int
+ setAge(age:int): void
Crop
- species: String
- purchasePrice: double
- sellingPrice: double
- daysUntilMature: int
- health: double
- dayPlanted: int
- count: static int
- id: int
+ getSpecies(): String
+ getId(): int
+ getPurchasePrice(): void
+ setPurchasePrice(purchasePrice:double): void
+ getSellingPrice(): double
+ setSellingPrice(sellingPrice: double): void
+ getDaysUntilMature(): int
+ setDaysUntilMature(daysUntilMature:int, growthBonus:int): void
+ setDayPlanted(day:int): void
+ getAge(currentday:int): int
+ getHealth(): void
+ setHealth(health: double): void
+ reduceDaysUntilMature(reduction:int): void
Animal
- species: String
- purchasePrice: double
- dailyBonus: double
- happiness: double
- health: double
- count: static int
- id: int
+ getSpecies(): String
+ getPurchasePrice(): void
+ setPurchasePrice(purchasePrice:double): void
+ getDailyBonus(): double
+ getHappiness(): double
+ getHealth(): void
+ getId():int
+ updateHappiness(happinessValue:double): void
+ updateHealth(health:double)
Store
- crops: Crop []
- animals: Animal []
- items: Item []
+ addCrop(crop:Crop): void
+ sellCrop(crop:Crop): void
+ getAnimals(): Animal []
+ getFoodItems(): Item []
+ getCropItems(): Item []
+ getItemStockCount(description:String)
+ getItems(): Item []
+ addAnimal(animal:Animal): void
+ sellAnimal(animal:Animal): void
+ addItem(item:Item): void
+ sellItem(item:Item): void
+ itemExists(id:int): boolean
+ cropExists(id:int): boolean
+ animalExists(id:int): boolean
+ fetchItem(id:int) Item
+ fetchAnimal(id:int) Item
+ fetchCrop(id:int) Item
Game
- farm: Farm
- store: Store
- duration: int {5 <=duration <= 10}
- actionCount: int
- day: int
- animalSpecies: Animal []
- cropSpecies: Crop []
- foodItems: Item []
- cropItems: Item []
+ getFarm(): farm
+ setFarm(farm:Farm): void
+ getStore(): store
+ setStore(store:Store): void
+ getDuration(): int
+ setDuration(duration:int): void
+ getDay(): int
+ skipDay(): void
+ getCropSpecies(): Crop []
+ setSpecies(animalSpecies: Animal [], cropSpecies: Crop [], foodItems: Item [], cropItems: Item [])
+ score(numCrops:int, numAnimals:int, animalStatus:double, moneyEarned:double): double
+ getActionCount(): int
+ setActionCount(actionCount:int): void
+ score(): double
+ profit(): double
+ populateStore(): void
+ advanceDay(): void
+ launchEndScreen(): void
+ closeEndScreen(endWindow:Endscreen): void
+ launchMainScreen(): void
+ closeMainScreen(mainWindow:Mainscreen): void
+ launchStartScreen(): void
+ closeStartScreen(startWindow:StartScreen)
Item
- name: String
- price: double
- purpose: String
- count: static int
- id: int
+ getName(): String
+ getPrice(): double
+ setPrice(price:double): void
+ isFoodItem()
+ getId(): int
+ getBenefit(): abstract int
CropItem
- growthEnhancement: int
+ getGrowthEnhancement(): int
+ setGrowthEnhancement(growthEnhancement:int): void
+ getBenefit(): int
FoodItem
- healthPoints: double
+ getHealthPoints(): double
+ setHealthPoints(healthPoints:double): void
+ getBenefit(): int