Model::Upravljanje podacima o sadnicama

Plant
(from dto)

-plantId: Integer
-scientificName: String
-knownAs: String
-description: String
-image: Blob
-isConifer: Boolean
-owned: Boolean
-deleted: Boolean

« ⚙»Plant()
« ⚙»Plant(plantId: Integer, scientificName: String,
knownAs: String, description: String, image: Blob, isConifer:
Boolean, owned: Boolean, ratios: PriceHeightRatio[*], deleted:
Boolean)
+getPlantId(): Integer
+setPlantId(plantId: Integer): void
+getScientificName(): String
+setScientificName(scientificName: String): void
+getKnownAs(): String
+setKnownAs(knownAs: String): void
+getDescription(): String
+setDescription(description: String): void
+getImage(): Blob
+setImage(image: Blob): void
+getOwned(): Boolean
+setOwned(owned: Boolean): void
+getIsConifer(): Boolean
+setIsConifer(isConifer: Boolean): void
+getDeleted(): Boolean
+setDeleted(deleted: Boolean): void
+hashCode(): int
+equals(obj: Object): boolean
+toString(): String
+getRatios(): PriceHeightRatio[*]
+setRatios(ratios: PriceHeightRatio[*]): void
+getPrice(height: BigDecimal): BigDecimal
+getHeightMin(height: BigDecimal): BigDecimal

ReproductionCutting
(from dto)

-date: Date
-produces: Integer
-takeARoot: Integer
-basisId: Integer
-deleted: Boolean

« ⚙»ReproductionCutting(basis: Basis, date:
Date, produces: Integer, takeARoot: Integer, basisId:
Integer, deleted: Boolean)
« ⚙»ReproductionCutting()
+getBasisId(): Integer
+setBasisId(basisId: Integer): void
+getBasis(): Basis
+setBasis(basisId: Basis): void
+getDate(): Date
+setDate(date: Date): void
+getProduces(): Integer
+setProduces(produces: Integer): void
+getTakeARoot(): Integer
+setTakeARoot(takeARoot: Integer): void
+getDeleted(): Boolean
+setDeleted(deleted: Boolean): void
+hashCode(): int
+equals(obj: Object): boolean
+toString(): String

-cutting

ReproductionCuttingTableItem
(from dto)

-date: StringProperty
-produced: IntegerProperty
-takeARoot: IntegerProperty

« ⚙»ReproductionCuttingTableItem()
« ⚙»ReproductionCuttingTableItem(cutting: ReproductionCutting)
+getCutting(): ReproductionCutting
+setCutting(cutting: ReproductionCutting): void
+dateProperty(): StringProperty
+getDate(): Date
+setDate(date: Date): void
+producedProperty(): IntegerProperty
+getProduced(): int
+setProduced(produced: int): void
+takeARootProperty(): IntegerProperty
+getTakeARoot(): int
+setTakeARoot(takeARoot: int): void

Region
(from dto)

-regionId: Integer
-numberOfPlants: Integer
-basisId: Integer
-coords: Double[*]
-deleted: Boolean

« ⚙»Region()
« ⚙»Region(regionId: Integer, numberOfPlants: Integer,
basis: Basis, basisId: Integer, coords: Double[*], deleted: Boolean)
+getBasisId(): Integer
+setBasisId(basisId: Integer): void
+getRegionId(): Integer
+getDeleted(): Boolean
+setDeleted(deleted: Boolean): void
+setRegionId(regionId: Integer): void
+getNumberOfPlants(): Integer
+setNumberOfPlants(numberOfPlants: Integer): void
+getBasis(): Basis
+setBasis(basis: Basis): void
+hashCode(): int
+equals(obj: Object): boolean
+addPlants(num: int): void
+removePlants(num: int): boolean
+toString(): String
+getCoords(): Double[*]
+setCoords(coords: Double[*]): void

PlantDAOImpl
(from implementation)

#pkColumns: String[*] {collection="List"}
#stdColumns: String[*] {collection="List"}
#allColumns: String[*] {collection="List"}
#tableName: String = "plant"
#allPlantOnPricelist: String = "select php.plant_id,
p.scientific_name, p.known_as, p.description, p.image,
p.is_conifer, p.owned, p.deleted from pricelist_has_plant php
inner join plant p on php.plant_id=p.plant_id and
php.pricelist_id=?"
#NUM_IN_REGIONS: String = "select sum(r.number_of_plants)
as total from basis b inner join region r on b.basis_id=r.basis_id
where b.plant_id=? and r.number_of_plants>0 and
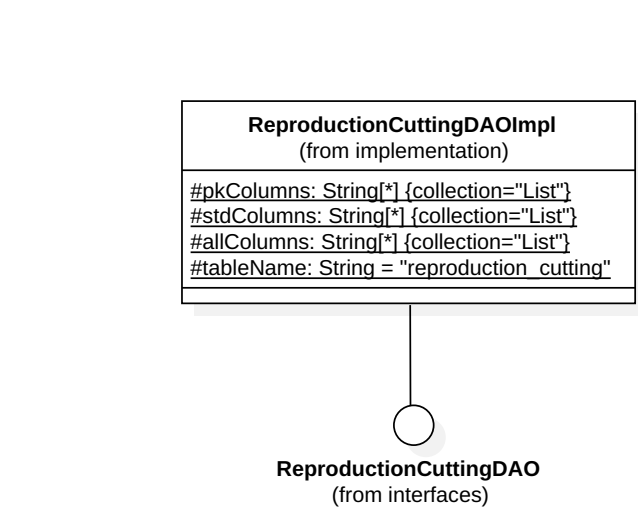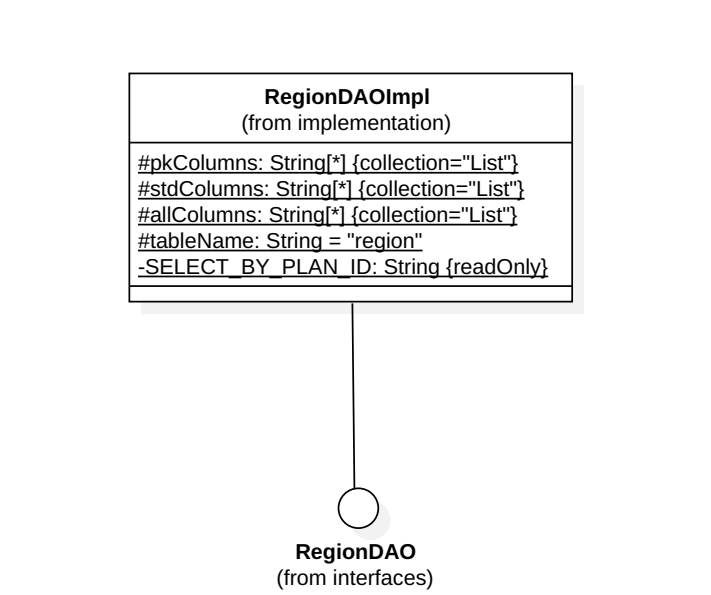r.deleted=false;"

PlantDAO
(from interfaces)

+getByPrimaryKey(plantId: Integer): Plant
+getAllByPrimaryKey(plantId: Integer): Plant
+selectAll(): Plant[*]
+select(whereStatement: String, bindVariables: Object[*]): Plant[*]
+selectCount(): long
+selectCount(whereStatement: String, bindVariables: Object[*]): long
+update(obj: Plant): int
+insert(obj: Plant): int
+delete(obj: Plant): int
+getByScientificName(scientificName: String): Plant[*]
+getPlantByPricelistId(pricelistId: Integer): Plant[*]
+getByKnownAs(knownAs: String): Plant[*]
+getByDescription(description: String): Plant[*]
+getByImage(image: Blob): Plant[*]
+getByIsConifer(isConifer: Boolean): Plant[*]
+getByOwned(owned: Boolean): Plant[*]
+getByDeleted(deleted: Boolean): Plant[*]
+getNumInRegions(plant: Plant): int

ReproductionCuttingDAOImpl
(from implementation)

#pkColumns: String[*] {collection="List"}
#stdColumns: String[*] {collection="List"}
#allColumns: String[*] {collection="List"}
#tableName: String = "reproduction_cutting"

ReproductionCuttingDAO
(from interfaces)

+getByPrimaryKey(idBasis: Integer, date: Date): ReproductionCutting
+selectAll(): ReproductionCutting[*]
+select(whereStatement: String, bindVariables: Object[*]): ReproductionCutting[*]
+selectCount(): Long
+selectCount(whereStatement: String, bindVariables: Object[*]): Long
+update(obj: ReproductionCutting): Integer
+insert(obj: ReproductionCutting): Integer
+delete(obj: ReproductionCutting): Integer
+getByBasisId(basisId: Integer): ReproductionCutting[*]
+getByDate(date: Date): ReproductionCutting[*]
+getByProduces(produces: Integer): ReproductionCutting[*]
+getByTakeARoot(takeARoot: Integer): ReproductionCutting[*]
+getByDeleted(deleted: Boolean): ReproductionCutting[*]

RegionDAOImpl
(from implementation)

#pkColumns: String[*] {collection="List"}
#stdColumns: String[*] {collection="List"}
#allColumns: String[*] {collection="List"}
#tableName: String = "region"
-SELECT_BY_PLAN_ID: String {readOnly}

RegionDAO
(from interfaces)

+getByPrimaryKey(regionId: Integer): Region
+selectAll(): Region[*]
+select(whereStatement: String, bindVariables: Object[*]): Region[*]
+selectCount(): Long
+selectCount(whereStatement: String, bindVariables: Object[*]): Long
+update(obj: Region): Integer
+insert(obj: Region): Integer
+delete(obj: Region): Integer
+getByNumberOfPlants(numberOfPlants: Integer): Region[*]
+getByBasis(basisId: Integer): Region[*]
+getByDeleted(deleted: Boolean): Region[*]
+getByPlanId(planId: Integer): Region[*]