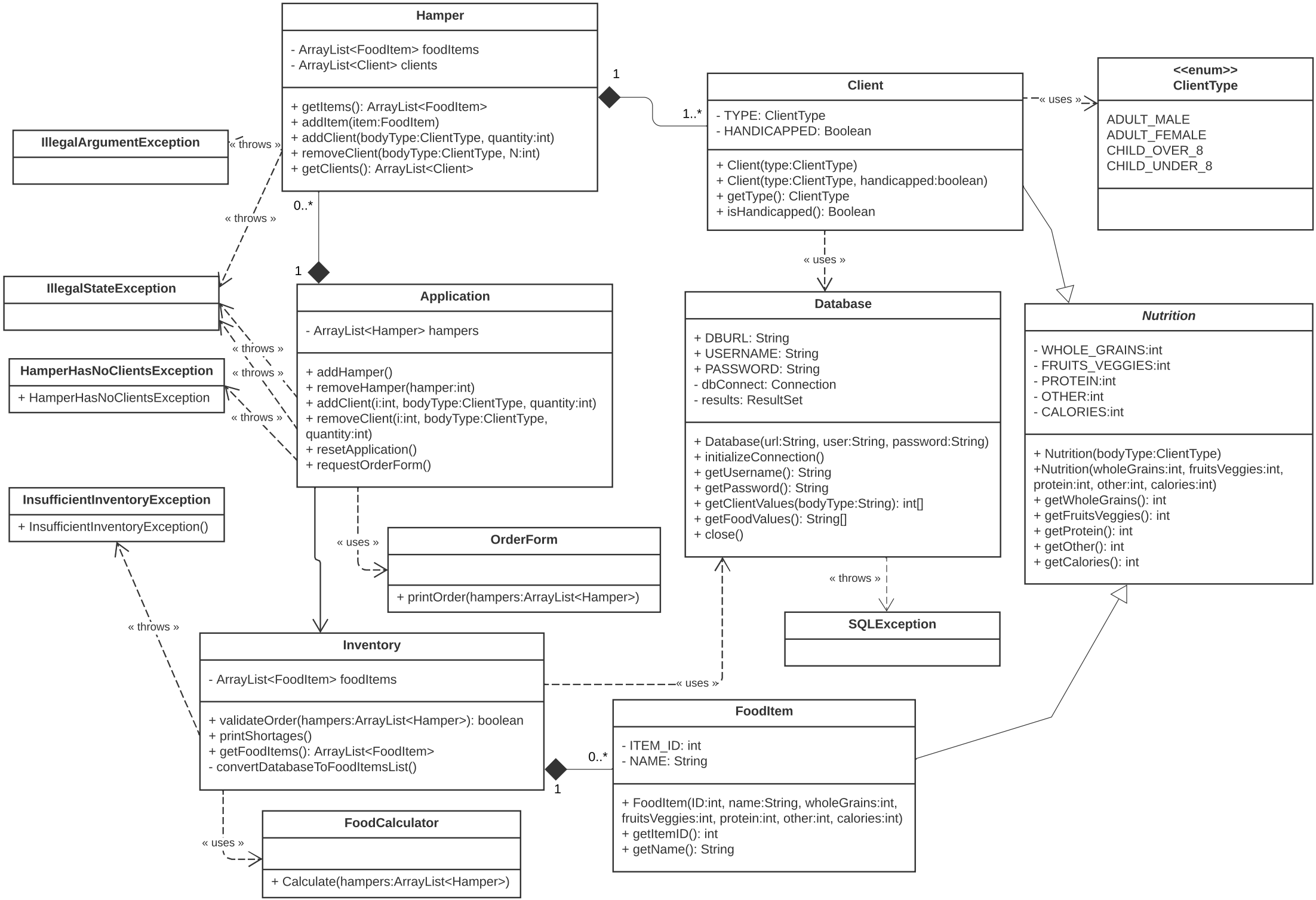
Hamper
- ArrayList<FoodItem> foodItems
- ArrayList<Client> clients
+ getItems(): ArrayList<FoodItem>
+ addItem(item:FoodItem)
+ addClient(bodyType:ClientType, quantity:int)
+ removeClient(bodyType:ClientType, N:int)
+ getClients(): ArrayList<Client>
1
1..*
Client
- TYPE: ClientType
- HANDICAPPED: Boolean
+ Client(type:ClientType)
+ Client(type:ClientType, handicapped:boolean)
+ getType(): ClientType
+ isHandicapped(): Boolean
« uses »
<<enum>>
ClientType
ADULT_MALE
ADULT_FEMALE
CHILD_OVER_8
CHILD_UNDER_8
IllegalArgumentException
« throws »
« throws »
0..*
1
IllegalStateException
Application
- ArrayList<Hamper> hampers
+ addHamper()
+ removeHamper(hamper:int)
+ addClient(i:int, bodyType:ClientType, quantity:int)
+ removeClient(i:int, bodyType:ClientType, quantity:int)
+ resetApplication()
+ requestOrderForm()
« throws »
« throws »
« throws »
HamperHasNoClientsException
+ HamperHasNoClientsException
« uses »
Database
+ DBURL: String
+ USERNAME: String
+ PASSWORD: String
- dbConnect: Connection
- results: ResultSet
+ Database(url:String, user:String, password:String)
+ initializeConnection()
+ getUsername(): String
+ getPassword(): String
+ getClientValues(bodyType:String): int[]
+ getFoodValues(): String[]
+ close()
Nutrition
- WHOLE_GRAINS:int
- FRUITS_VEGGIES:int
- PROTEIN:int
- OTHER:int
- CALORIES:int
+ Nutrition(bodyType:ClientType)
+Nutrition(wholeGrains:int, fruitsVeggies:int, protein:int, other:int, calories:int)
+ getWholeGrains(): int
+ getFruitsVeggies(): int
+ getProtein(): int
+ getOther(): int
+ getCalories(): int
InsufficientInventoryException
+ InsufficientInventoryException()
« uses »
OrderForm
+ printOrder(hampers:ArrayList<Hamper>)
« throws »
SQLException
« throws »
Inventory
- ArrayList<FoodItem> foodItems
+ validateOrder(hampers:ArrayList<Hamper>): boolean
+ printShortages()
+ getFoodItems(): ArrayList<FoodItem>
- convertDatabaseToFoodItemsList()
« uses »
0..*
1
FoodItem
- ITEM_ID: int
- NAME: String
+ FoodItem(ID:int, name:String, wholeGrains:int, fruitsVeggies:int, protein:int, other:int, calories:int)
+ getItemID(): int
+ getName(): String
« uses »
FoodCalculator
+ Calculate(hampers:ArrayList<Hamper>)