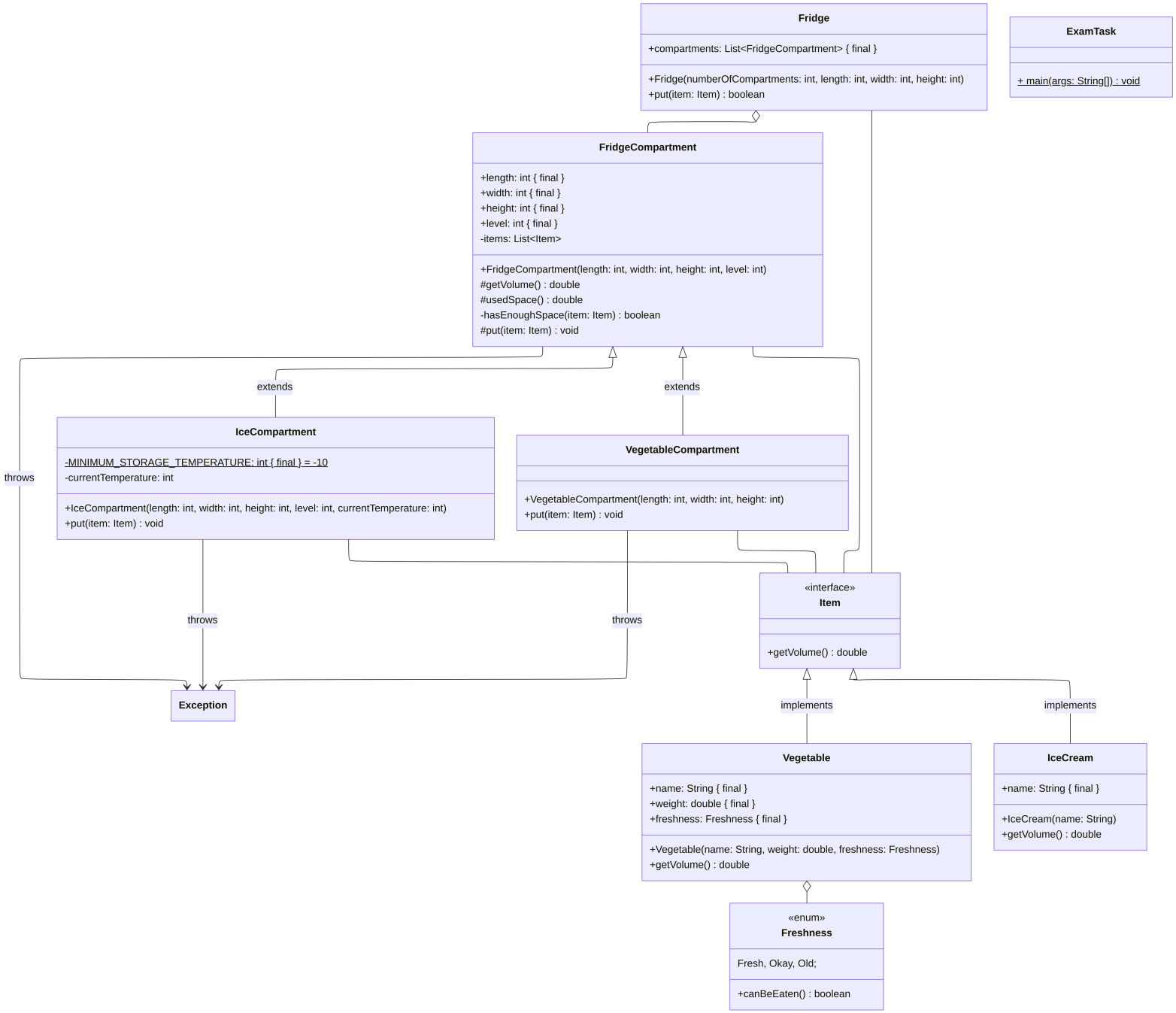
Fridge
+compartments: List<FridgeCompartment> { final }
+Fridge(numberOfCompartments: int, length: int, width: int, height: int)
+put(item: Item) : boolean
ExamTask
+ main(args: String[]) : void
FridgeCompartment
+length: int { final }
+width: int { final }
+height: int { final }
+level: int { final }
-items: List<Item>
+FridgeCompartment(length: int, width: int, height: int, level: int)
#getVolume() : double
#usedSpace() : double
-hasEnoughSpace(item: Item) : boolean
#put(item: Item) : void
extends
extends
throws
IceCompartment
-MINIMUM_STORAGE_TEMPERATURE: int { final } = -10
-currentTemperature: int
+IceCompartment(length: int, width: int, height: int, level: int, currentTemperature: int)
+put(item: Item) : void
VegetableCompartment
+VegetableCompartment(length: int, width: int, height: int)
+put(item: Item) : void
throws
throws
«interface»
Item
+getVolume() : double
Exception
implements
implements
Vegetable
+name: String { final }
+weight: double { final }
+freshness: Freshness { final }
+Vegetable(name: String, weight: double, freshness: Freshness)
+getVolume() : double
IceCream
+name: String { final }
+IceCream(name: String)
+getVolume() : double
«enum»
Freshness
Fresh, Okay, Old;
+canBeEaten() : boolean