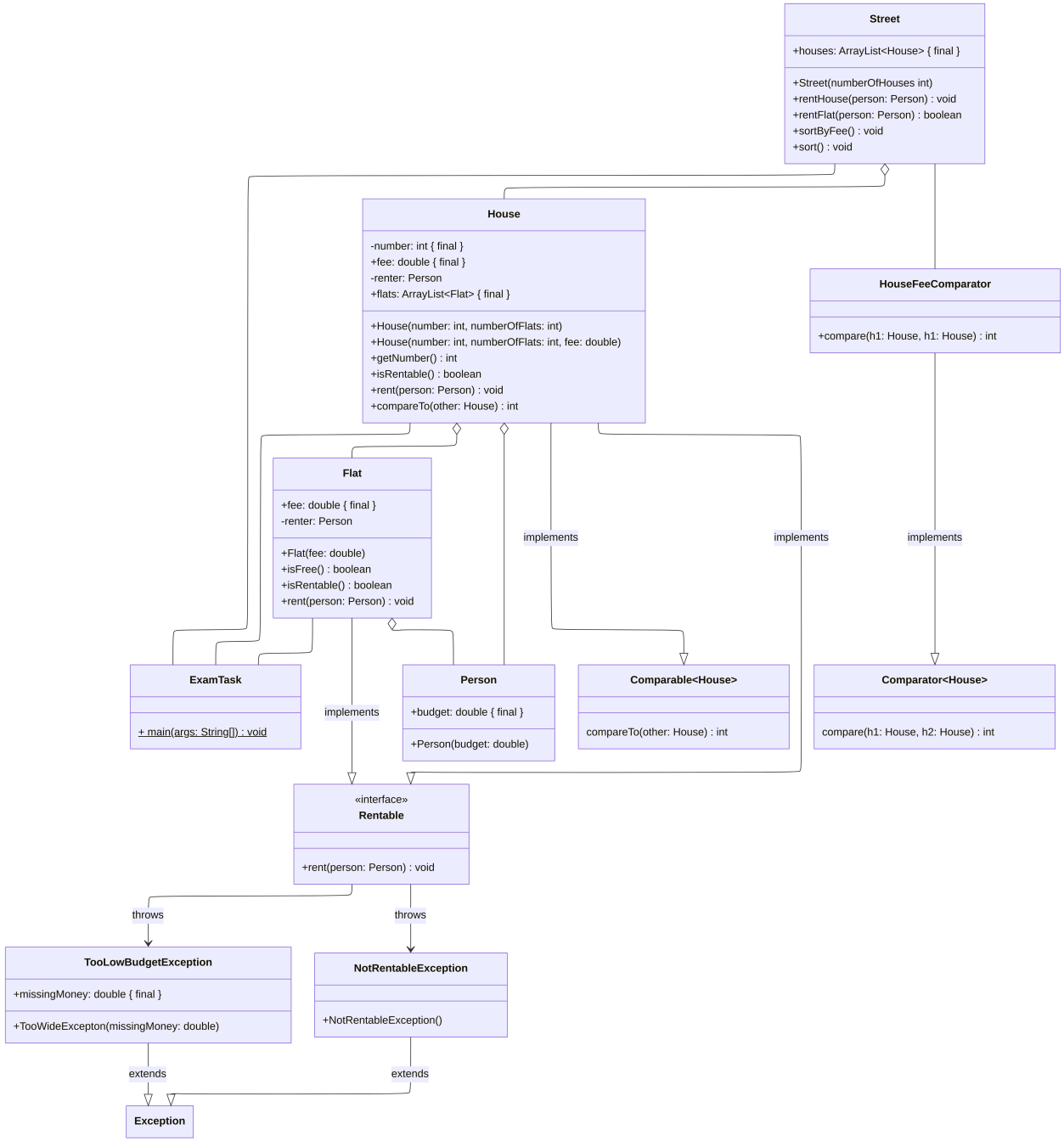
Street
+houses: ArrayList<House> { final }
+Street(numberOfHouses int)
+rentHouse(person: Person) : void
+rentFlat(person: Person) : boolean
+sortByFee() : void
+sort() : void
House
-number: int { final }
+fee: double { final }
-renter: Person
+flats: ArrayList<Flat> { final }
+House(number: int, numberOfFlats: int)
+House(number: int, numberOfFlats: int, fee: double)
+getNumber() : int
+isRentable() : boolean
+rent(person: Person) : void
+compareTo(other: House) : int
HouseFeeComparator
+compare(h1: House, h1: House) : int
Flat
+fee: double { final }
-renter: Person
+Flat(fee: double)
+isFree() : boolean
+isRentable() : boolean
+rent(person: Person) : void
implements
implements
implements
ExamTask
+ main(args: String[]) : void
implements
Person
+budget: double { final }
+Person(budget: double)
Comparable<House>
compareTo(other: House) : int
Comparator<House>
compare(h1: House, h2: House) : int
«interface»
Rentable
+rent(person: Person) : void
throws
throws
TooLowBudgetException
+missingMoney: double { final }
+TooWideExcepton(missingMoney: double)
NotRentableException
+NotRentableException()
extends
extends
Exception