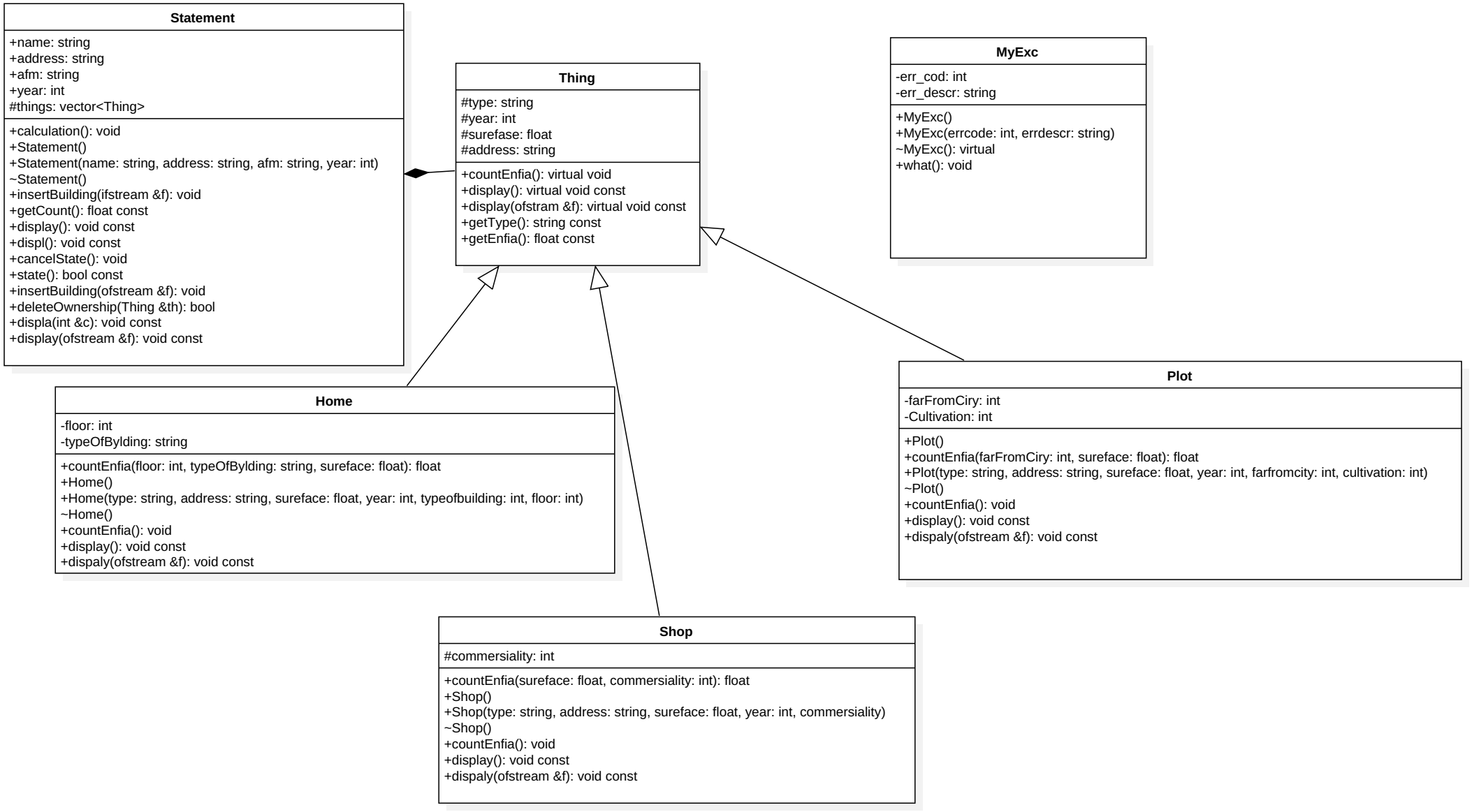
Model::Main


Statement
+name: string
+address: string
+afm: string
+year: int
#things: vector<Thing>
+calculation(): void
+Statement()
+Statement(name: string, address: string, afm: string, year: int)
~Statement()
+insertBuilding(ifstream &f): void
+getCount(): float const
+display(): void const
+displ(): void const
+cancelState(): void
+state(): bool const
+insertBuilding(ofstream &f): void
+deleteOwnership(Thing &th): bool
+displa(int &c): void const
+display(ofstream &f): void const
Thing
#type: string
#year: int
#surefase: float
#address: string
+countEnfia(): virtual void
+display(): virtual void const
+display(ofstram &f): virtual void const
+getType(): string const
+getEnfia(): float const
MyExc
-err_cod: int
-err_descr: string
+MyExc()
+MyExc(errcode: int, errdescr: string)
~MyExc(): virtual
+what(): void
Home
-floor: int
-typeOfBylding: string
+countEnfia(floor: int, typeOfBylding: string, sureface: float): float
+Home()
+Home(type: string, address: string, sureface: float, year: int, typeofbuilding: int, floor: int)
~Home()
+countEnfia(): void
+display(): void const
+dispaly(ofstream &f): void const
Plot
-farFromCiry: int
-Cultivation: int
+Plot()
+countEnfia(farFromCiry: int, sureface: float): float
+Plot(type: string, address: string, sureface: float, year: int, farfromcity: int, cultivation: int)
~Plot()
+countEnfia(): void
+display(): void const
+dispaly(ofstream &f): void const
Shop
#commersiality: int
+countEnfia(sureface: float, commersiality: int): float
+Shop()
+Shop(type: string, address: string, sureface: float, year: int, commersiality)
~Shop()
+countEnfia(): void
+display(): void const
+dispaly(ofstream &f): void const