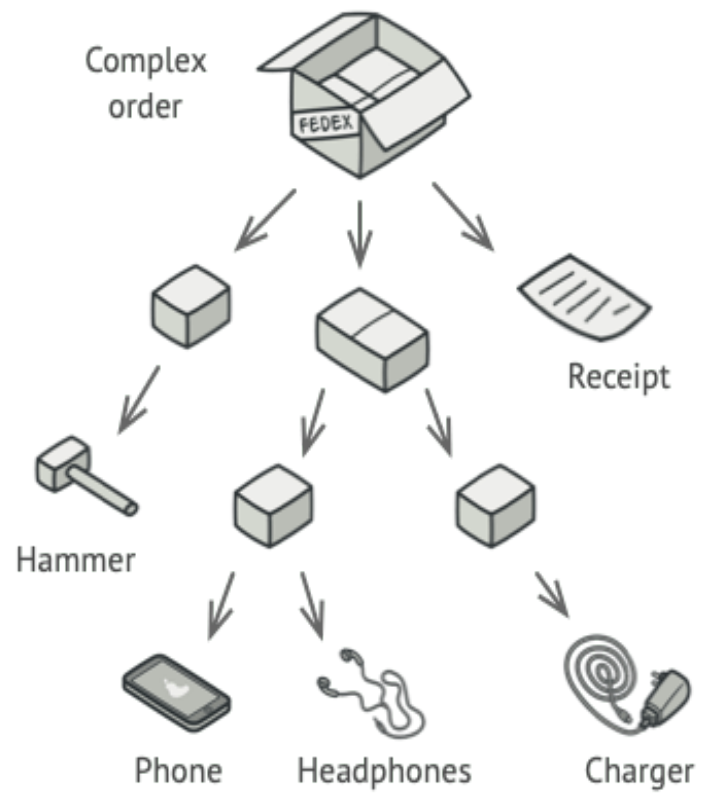


Design Mønstre

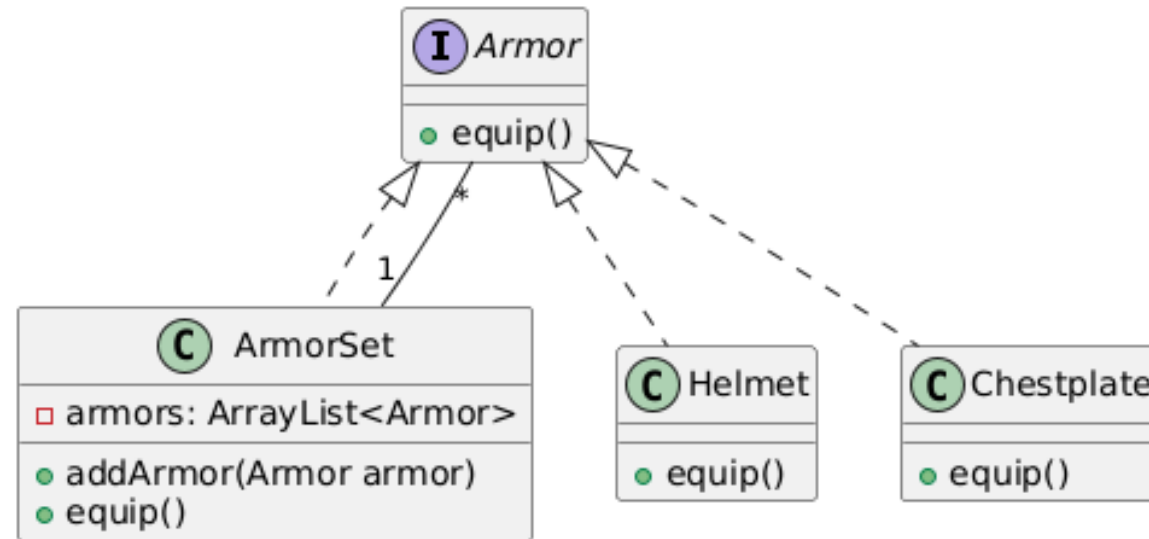
Builder + Decorator



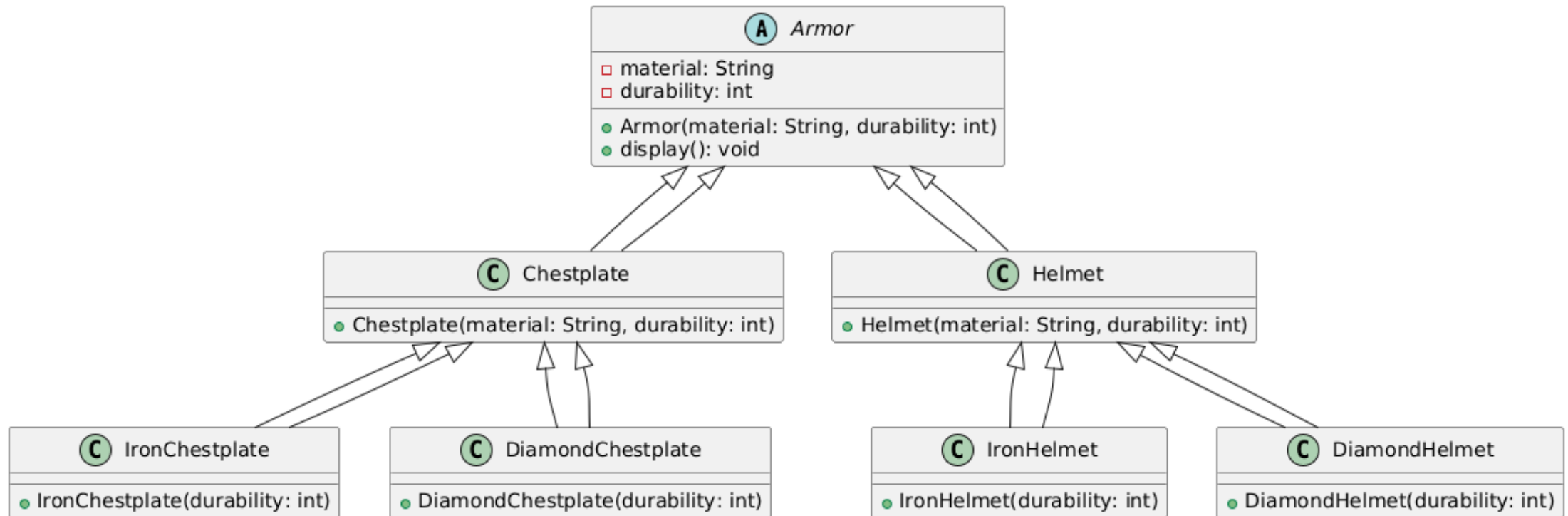
Order Håndtagering System.

finde Prisen på hvert objekt.

Minecraft



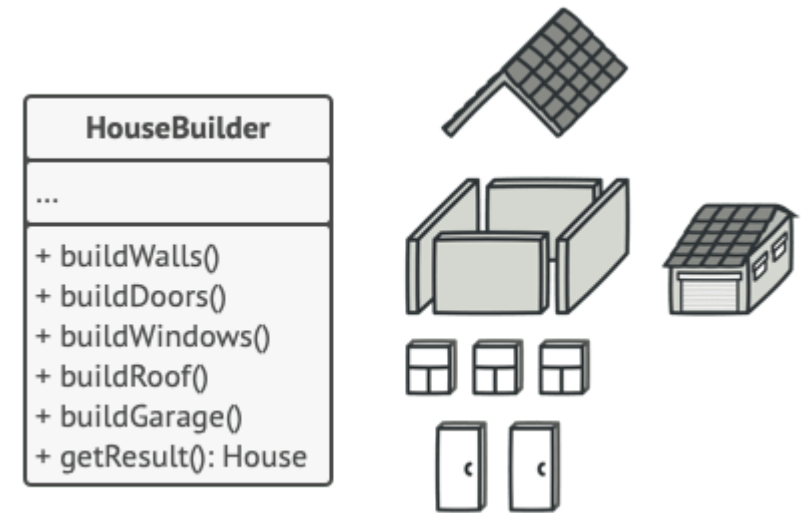
Mange Subklasser



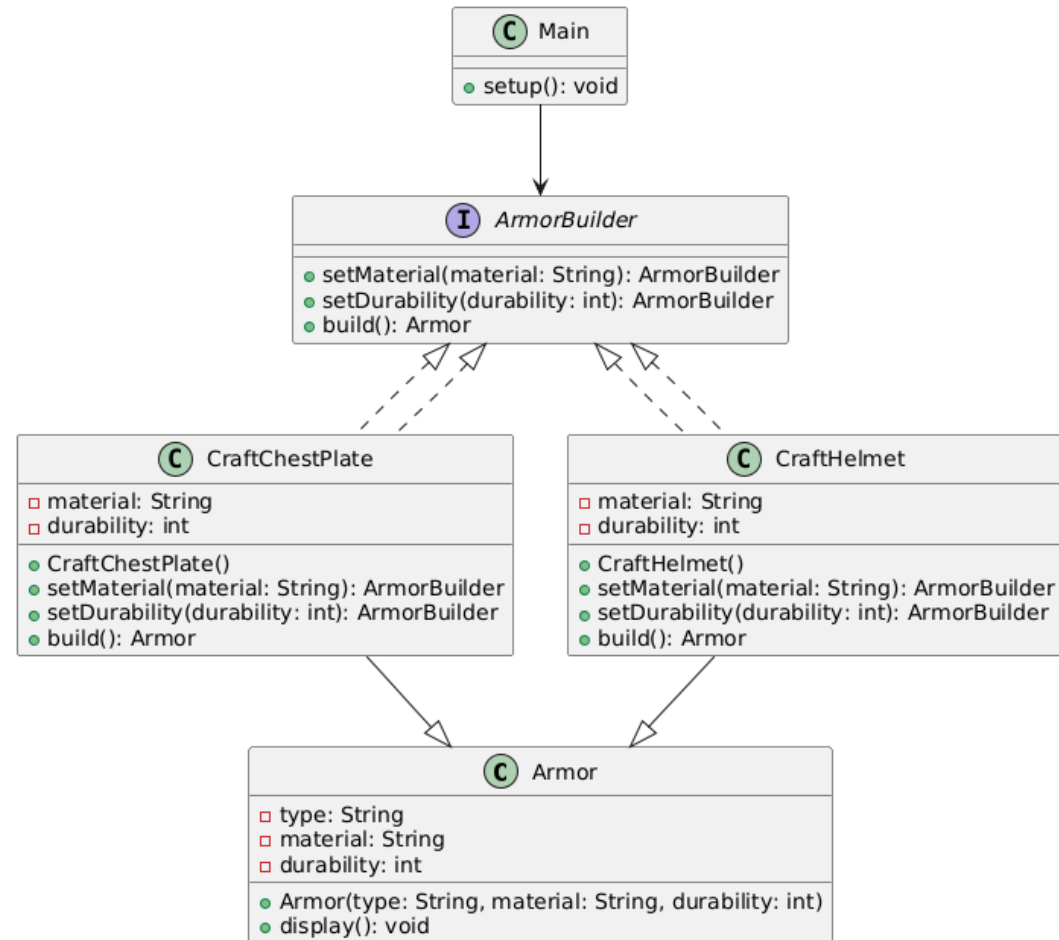
Builder Pattern

I stedet for at Direkte oprette en Armor klasse
Kan vi oprette en CraftArmor klasse.

konstruere objekter trin-for-trin, udsætte konstruktionstrin eller køre trin for trin.
genbruge den samme konstruktionskode, når du bygger forskellige repræsentationer af produkter.
Enkeltansvarsprincippet. isolere kompleks konstruktionskode fra forretningslogikken i produktet.
Den samlede kompleksitet af koden stiger, da mønsteret kræver at oprette flere nye klasser.



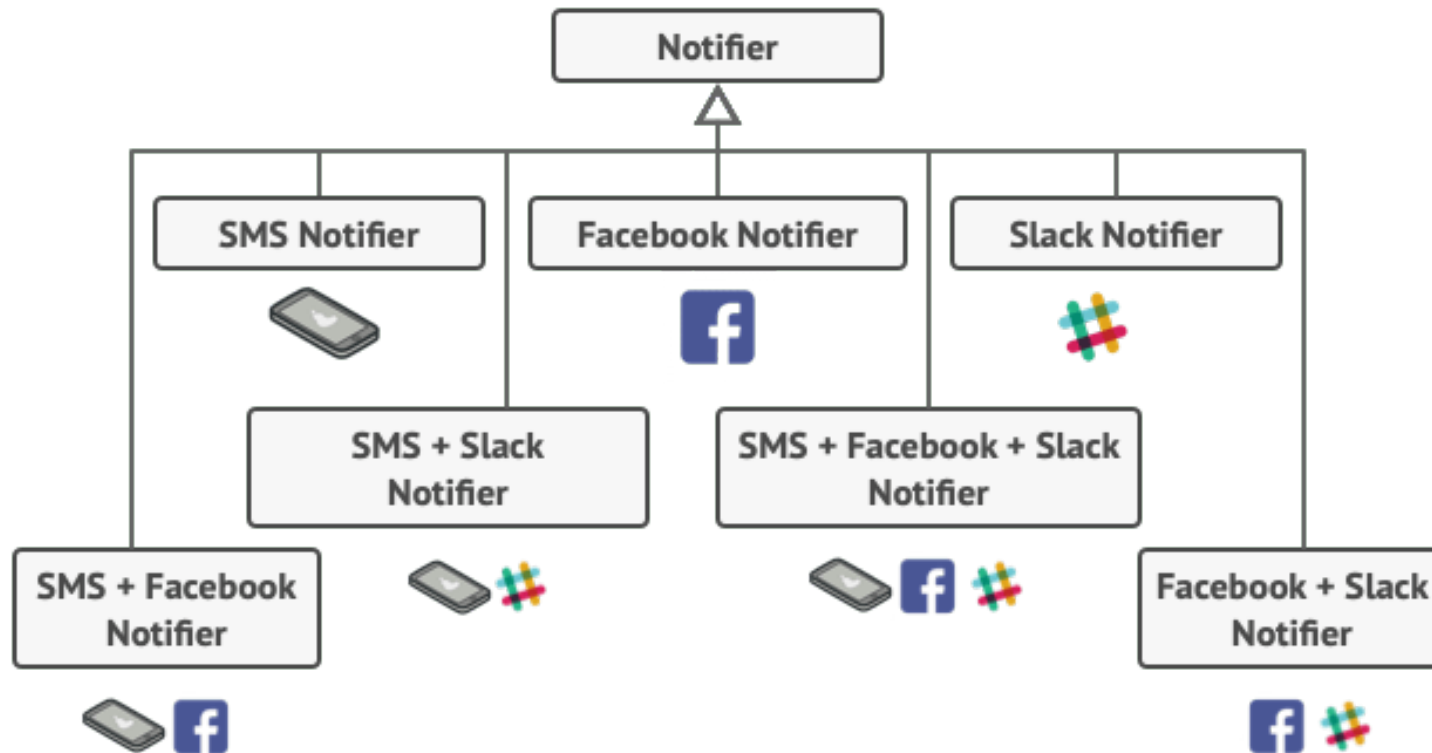
Minecraft Builder Diagram



Decorator Mønstre

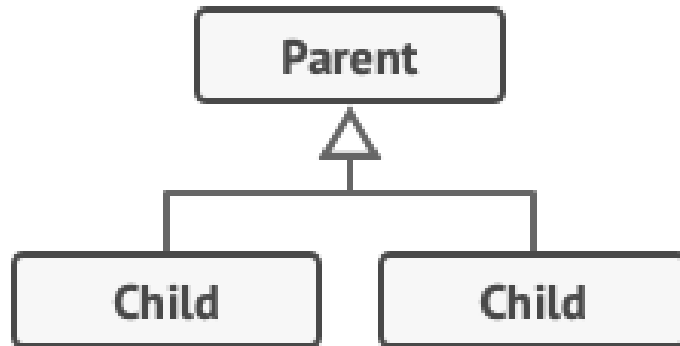
Objekter med forskellige adfærd

- Objekter kan have forskellige adfærd og hver har sin egen ansvar.



Arv vs Aggregation

Inheritance

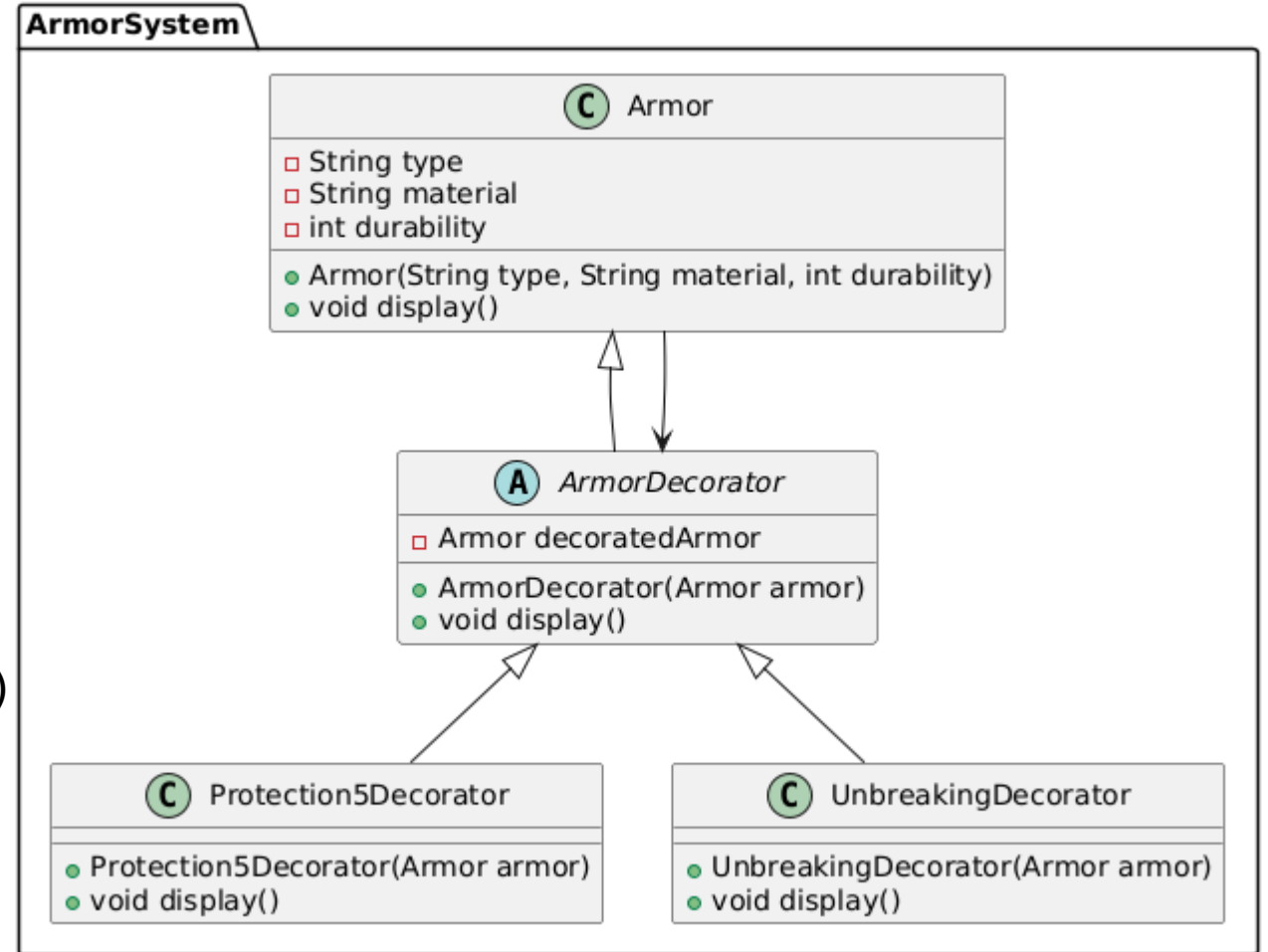


Aggregation



Minecraft Decorator

- **Armor** (interface)
- **BasicArmor** (concrete class)
- **ArmorDecorator** (abstract decorator class)
- **EnchantedArmor** (concrete decorator)
- **UnbrakingArmor** (another concrete decorator)



Minecraft Opgave

- Opgave 1: Udefra Builder patten diagram og Decorator Diagram, Lav en diagram som viser disse to patten i samme program.
- Opgave 2: Find ud af hvordan composite Patten kan bruges som ekstra værktøj til at organiser objekter med Decorator og Builder pattens.
- Opgave 3: (ekstra frivilig opgave) Arbejd videre med Minecraft kode eksemplet fra sidst og implementer Builder og decorator pattens i programmet.