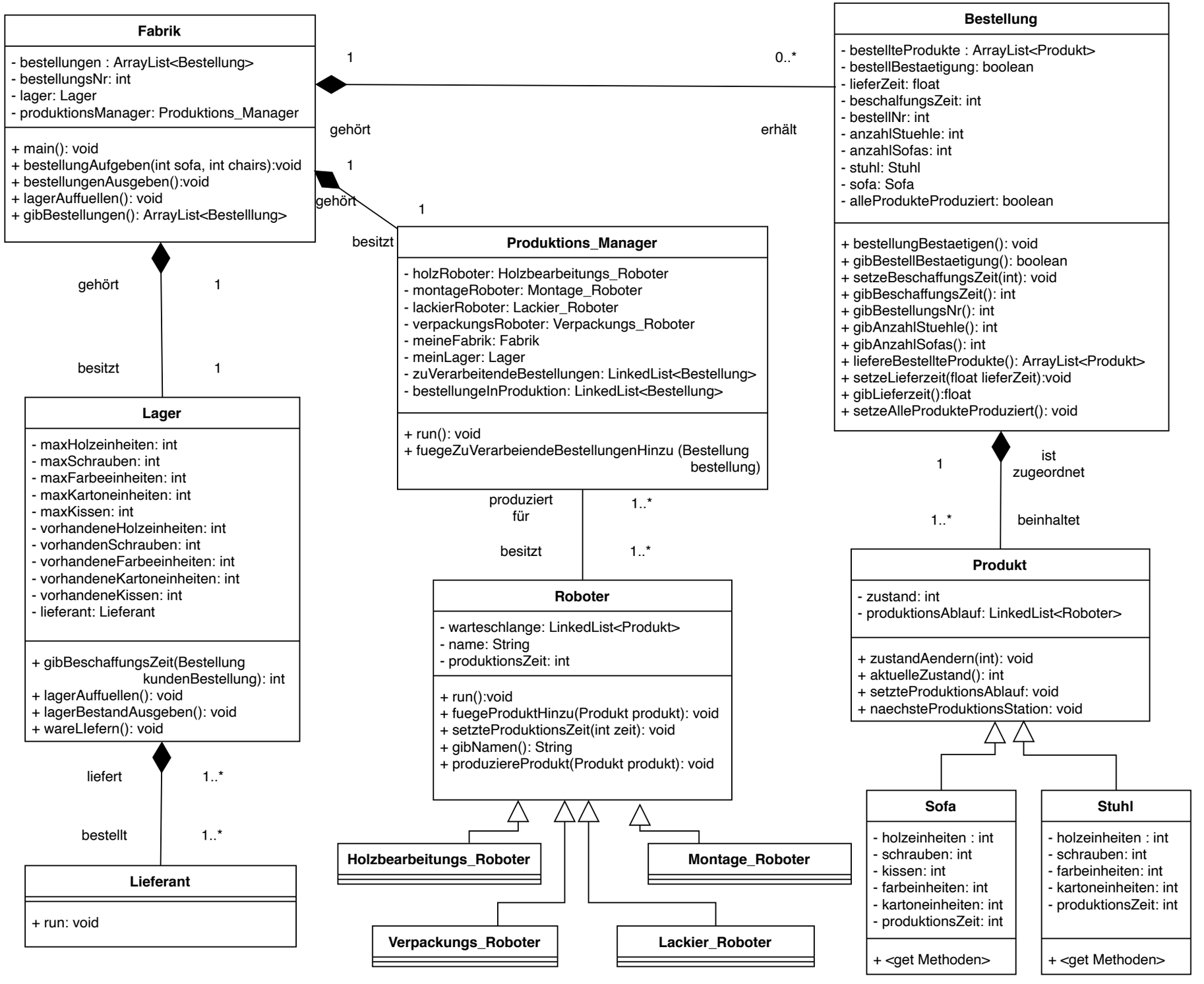
Fabrik
- bestellungen : ArrayList<Bestellung>
- bestellungsNr: int
- lager: Lager
- produktionsManager: Produktions_Manager
+ main(): void
+ bestellungAufgeben(int sofa, int chairs):void
+ bestellungenAusgeben():void
+ lagerAuffuellen(): void
+ gibBestellungen(): ArrayList<Bestelllung>
1
0..*
gehört
erhält
1
gehört
1
besitzt
Bestellung
- bestellteProdukte : ArrayList<Produkt>
- bestellBestaetigung: boolean
- lieferZeit: float
- beschalfungsZeit: int
- bestellNr: int
- anzahlStuehle: int
- anzahlSofas: int
- stuhl: Stuhl
- sofa: Sofa
- alleProdukteProduziert: boolean
+ bestellungBestaetigen(): void
+ gibBestellBestaetigung(): boolean
+ setzeBeschaffungsZeit(int): void
+ gibBeschaffungsZeit(): int
+ gibBestellungsNr(): int
+ gibAnzahlStuehle(): int
+ gibAnzahlSofas(): int
+ liefereBestellteProdukte(): ArrayList<Produkt>
+ setzeLieferzeit(float lieferZeit):void
+ gibLieferzeit():float
+ setzeAlleProdukteProduziert(): void
gehört
1
besitzt
1
Produktions_Manager
- holzRoboter: Holzbearbeitungs_Roboter
- montageRoboter: Montage_Roboter
- lackierRoboter: Lackier_Roboter
- verpackungsRoboter: Verpackungs_Roboter
- meineFabrik: Fabrik
- meinLager: Lager
- zuVerarbeitendeBestellungen: LinkedList<Bestellung>
- bestellungeInProduktion: LinkedList<Bestellung>
+ run(): void
+ fuegeZuVerarbeiendeBestellungenHinzu (Bestellung bestellung)
produziert für
1..*
besitzt
1..*
1
ist zugeordnet
1..*
beinhaltet
Lager
- maxHolzeinheiten: int
- maxSchrauben: int
- maxFarbeeinheiten: int
- maxKartoneinheiten: int
- maxKissen: int
- vorhandeneHolzeinheiten: int
- vorhandenSchrauben: int
- vorhandeneFarbeeinheiten: int
- vorhandeneKartoneinheiten: int
- vorhandeneKissen: int
- lieferant: Lieferant
+ gibBeschaffungsZeit(Bestellung kundenBestellung): int
+ lagerAuffuellen(): void
+ lagerBestandAusgeben(): void
+ wareLIefern(): void
Produkt
- zustand: int
- produktionsAblauf: LinkedList<Roboter>
+ zustandAendern(int): void
+ aktuelleZustand(): int
+ setzteProduktionsAblauf: void
+ naechsteProduktionsStation: void
Roboter
- warteschlange: LinkedList<Produkt>
- name: String
- produktionsZeit: int
+ run():void
+ fuegeProduktHinzu(Produkt produkt): void
+ setzteProduktionsZeit(int zeit): void
+ gibNamen(): String
+ produziereProdukt(Produkt produkt): void
liefert
1..*
bestellt
1..*
Lieferant
+ run: void
Holzbearbeitungs_Roboter
Montage_Roboter
Verpackungs_Roboter
Lackier_Roboter
Sofa
- holzeinheiten : int
- schrauben: int
- kissen: int
- farbeinheiten: int
- kartoneinheiten: int
- produktionsZeit: int
+ <get Methoden>
Stuhl
- holzeinheiten : int
- schrauben: int
- farbeinheiten: int
- kartoneinheiten: int
- produktionsZeit: int
+ <get Methoden>