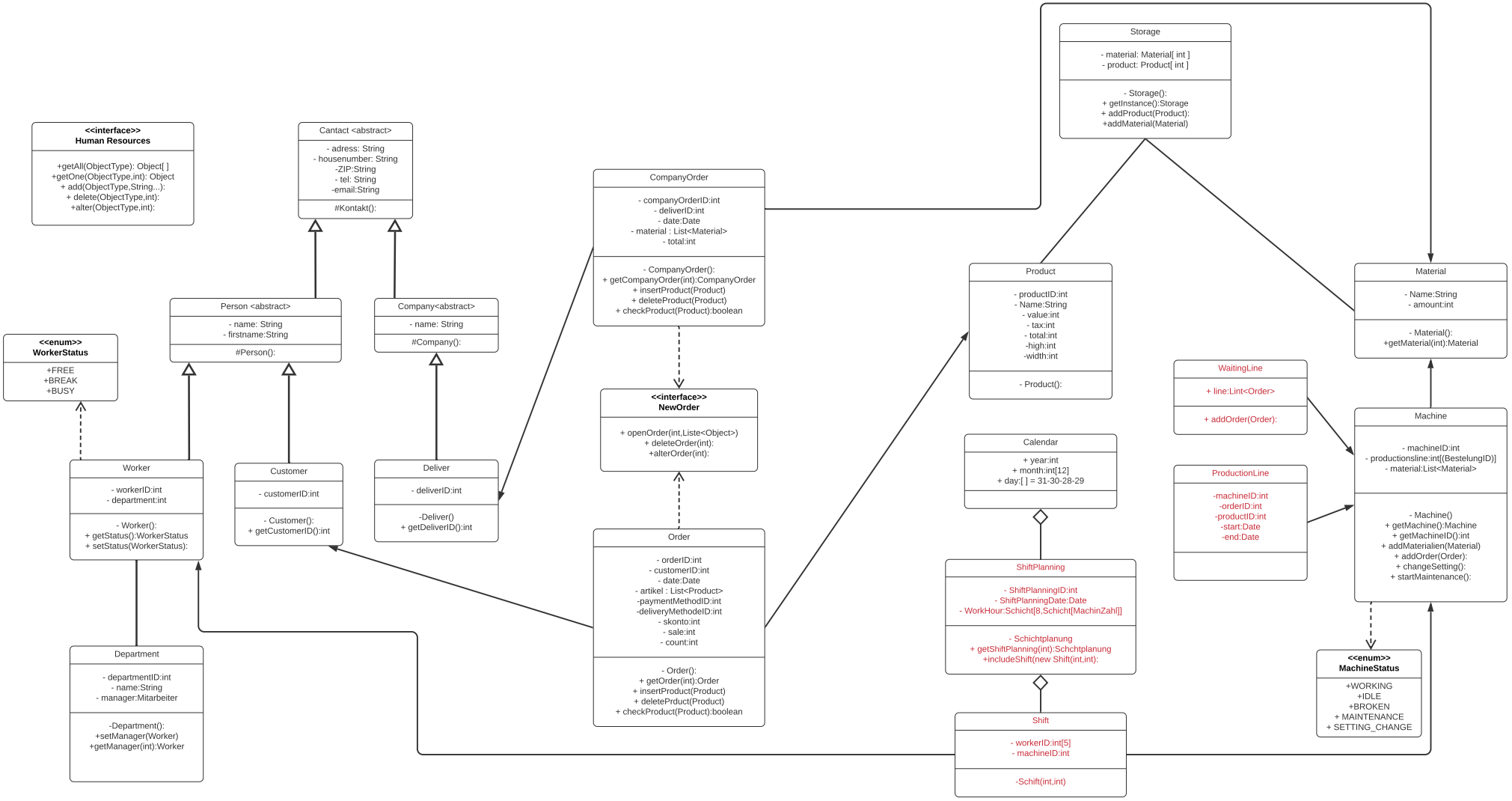
Storage
- material: Material[int]
- product: Product[int]
- Storage():
+ getInstance():Storage
+ addProduct(Product):
+addMaterial(Material)
<<interface>>
Human Resources
+getAll(ObjectType): Object[ ]
+getOne(ObjectType,int): Object
+ add(ObjectType,String...):
+ delete(ObjectType,int):
+alter(ObjectType,int):
Cantact <abstract>
- adress: String
- housenumber: String
-ZIP:String
- tel: String
-email:String
#Kontakt():
CompanyOrder
- companyOrderID:int
- deliverID:int
- date:Date
- material : List<Material>
- total:int
- CompanyOrder():
+ getCompanyOrder(int):CompanyOrder
+ insertProduct(Product)
+ deleteProduct(Product)
+ checkProduct(Product):boolean
Product
- productID:int
- Name:String
- value:int
- tax:int
- total:int
-high:int
-width:int
- Product():
Material
- Name:String
- amount:int
- Material():
+getMaterial(int):Material
Person <abstract>
- name: String
- firstname:String
#Person():
Company<abstract>
- name: String
#Company():
<<enum>>
WorkerStatus
+FREE
+BREAK
+BUSY
<<interface>>
NewOrder
+ openOrder(int,Liste<Object>)
+ deleteOrder(int):
+alterOrder(int):
WaitingLine
+ line:Lint<Order>
+ addOrder(Order):
Machine
- machineID:int
- productionsline:int[(BestelungID)]
- material:List<Material>
- Machine()
+ getMachine():Machine
+ getMachineID():int
+ addMaterialien(Material)
+ addOrder(Order):
+ changeSetting():
+ startMaintenance():
Calendar
+ year:int
+ month:int[12]
+ day:[ ] = 31-30-28-29
ProductionLine
-machineID:int
-orderID:int
-productID:int
-start:Date
-end:Date
Worker
- workerID:int
- department:int
- Worker():
+ getStatus():WorkerStatus
+ setStatus(WorkerStatus):
Customer
- customerID:int
- Customer():
+ getCustomerID():int
Deliver
- deliverID:int
-Deliver()
+ getDeliverID():int
Order
- orderID:int
- customerID:int
- date:Date
- artikel : List<Product>
-paymentMethodID:int
-deliveryMethodeID:int
- skonto:int
- sale:int
- count:int
- Order():
+ getOrder(int):Order
+ insertProduct(Product)
+ deleteProduct(Product)
+ checkProduct(Product):boolean
ShiftPlanning
- ShiftPlanningID:int
- ShiftPlanningDate:Date
- WorkHour:Schicht[8,Schicht[MachinZahl]]
- Schichtplanung
+ getShiftPlanning(int):Schchtplanung
+includeShift(new Shift(int,int):
<<enum>>
MachineStatus
+WORKING
+IDLE
+BROKEN
+ MAINTENANCE
+ SETTING_CHANGE
Department
- departmentID:int
- name:String
- manager:Mitarbeiter
-Department():
+setManager(Worker)
+getManager(int):Worker
Shift
- workerID:int[5]
- machineID:int
-Schift(int,int)