

Prodotto
<ul style="list-style-type: none"> - id: Int - nome: String - marchio: String - produttore: String - formato: String - descrizione: String - disponibilità: String - prezzo: BigDecimal - categoria: String - foto: Byte[]
<ul style="list-style-type: none"> + getId() : Int + getNome() : String + getMarchio() : String + getProduttore() : String + getFormato() : String + getDescrizione() : String + getDisponibilità() : String + getPrezzo() : BigDecimal + getCategoria() : String + getFoto() : Byte [] + setId(id : Int) : void + setNome(nome : String) : void + setMarchio(marchio : String) : void + setProduttore(produttore : String) : void + setFormato(formato : String) : void + setDescription(descrizione : String) : void + setDisponibilità(disponibilità : String) : void + setPrezzo(prezzo : BigDecimal) : void + setCategoria(categoria : String) : void + setFoto(foto : byte []) : void

```

context Prodotto::getId( )
    post: result = id
context Prodotto::getNome( )
    post: result = nome
context Prodotto::getMarchio( )
    post: result = marchio
context Prodotto::getProduttore( )
    post: result = produttore
context Prodotto::getFormato( )
    post: result = formato
context Prodotto::getDescrizione( )
    post: result = descrizione
context Prodotto::getDisponibilità( )
    post: result = disponibilità
context Prodotto::getPrezzo( )
    post: result = prezzo
context Prodotto::getCategoria( )
    post: result = categoria

```

```

Context Prodotto::getFoto( )
    post: result = foto
Context Prodotto::setId(id)
    post: self.id = id
Context Prodotto::setNome(nome)
    post: self.nome = nome
Context Prodotto::setMarchio(marchio)
    post: self.marchio = marchio
Context Prodotto::setProduttore(produttore)
    post: self.produttore = produttore
Context Prodotto::setFormato(formato)
    post: self.formato = formato
Context Prodotto::setDescrizione(descrizione)
    post: self.descrizione = descrizione
Context Prodotto::setDisponibilità(disponibilità)
    post: self.disponibilità = disponibilità
Context Prodotto::setPrezzo(prezzo)
    post: self.prezzo = prezzo
Context Prodotto::setCategoria(categoria)
    post: self.categoria = categoria
Context Prodotto::setFoto(foto)
    post: self.foto = foto

```

ProdottoModel
//
+ doRetrieveByKey(id : Int) : Prodotto + doRetrieveAll(order : String) : Collection<Prodotto> + doSave(prodotto : Prodotto) : void + doUpdate(prodotto : Prodotto) : void + doUpdateCategoria(prodottoid : Int, categoria : String) : void + doDelete(prodotto : Prodotto bean) : void + doRetrieveAllByCategoria(categoria : String) : Collection<Prodotto> + doRicerca(nome : String, order : String) : Collection<Prodotto> + doRetrieveSvincolati() : Collection<Prodotto>

```

context ProdottoModel::doRetrieveByKey(id)
    pre: id != null
    post: result = database.Prodotto -> select ( p | prodotto.id = p.id )
context ProdottoModel::doRetrieveAll()
    post: result = database.Prodotto
context ProdottoModel::doSave(prodotto)
    pre: prodotto.id !=null and prodotto.nome!=null and prodotto.marchio!=null and
    prodotto.descrizione!=null and prodotto.disponibilità>=null and prodotto.prezzo>=0
    post: database.Prodotto -> exists( p | prodotto.id = p.id and prodotto.nome = p.nome and
    prodotto.marchio = p.marchio and prodotto.produttore= p.produttore and prodotto.formato =
    p.formato and prodotto.descrizione = p.descrizione and prodotto.disponibilità = p.disponibilità and
    prodotto.prezzo = p.prezzo and prodotto.categoria = p.categoria and prodotto.foto = p.foto)
context ProdottoModel::doUpdate(prodotto)

```

pre: prodotto.id !=null and prodotto.nome!=null and prodotto.marchio!=null and
prodotto.descrizione!=null and prodotto.disponibilità!=null and prodotto.prezzo!=null and
database.Prodotto -> exists(p | p.id = prodotto.id)

post: database.Prodotto -> exists(p | prodotto.id = p.id and prodotto.nome = p.nome and
prodotto.marchio = p.marchio and prodotto.prodotto= p.prodotto and prodotto.formato =
p.formato and prodotto.descrizione = p.descrizione and prodotto.disponibilità = p.disponibilità and
prodotto.prezzo = p.prezzo and prodotto.categoria = p.categoria and prodotto.foto = p.foto

context ProdottoModel::doUpdateCategoria(prodottold, categoria)
pre: database.Prodotto -> exists(p | p.id = prodottold) and database.Categoria -> exists(c | c.nome
= categoria and c.id = categoriald)
post: database.Prodotto -> exists(p | p.id = prodottold and p.categoria = categoriald)

context ProdottoModel::doDelete(prodotto)
pre: prodotto.databaseProdotto -> exists(id | prodotto.id = id)
post: not(database.Prodotto -> exists(id | prodotto.id = id))

context ProdottoModel::doRetrieveAllByCategoria(categoria)
post: result = database.Prodotto -> select(p | p.categoria = categoria)

context ProdottoModel::doRicerca(nome, order)
pre: nome!=null and order!=null
post: database.Prodotto -> select(p | p.nome contains nome) order by order

context ProdottoModel::doRetrieveSvincolati()
post: result = database.Prodotti -> select(p | p.categoria=null)

Ordine
<ul style="list-style-type: none"> - nomeRicevente: String - cognomeRicevente: String - email: String - cellulare: String - ncivico: Int - città: String - via: String - paese: String - provincia: String - CAP: String - data_ordine: Date - data_arrivo: Date - id: Int - prezzo: Float - stato: String - cliente: String - righeOrdine: Collection<RigaOrdine>
<ul style="list-style-type: none"> + getNomeRicevente: String + getCognomeRicevente: String + getEmail: String + getCellulare: String + getNcivico: Int + getCittà: String + getVia: String + getPaese: String + getProvincia: String + getCAP: String

```

+ getData_ordine( ) : Date
+ getData_arrivo( ) : Date
+ getId( ) : String
+ getPrezzo( ) : BigDecimal
+ getStato( ) : String
+ getCliente( ) : String
+ getRigaOrdine: Collection<RigaOrdine>
+ setNomeRicevente(nomeRicevente: String) : void
+ setCognomeRicevente(cognomeRicevente: String) : void
+ setEmail(email: String) : void
+ setCellulare(cellulare: String) : void
+ setNcivico(ncivico: Int) : void
+ setCittà(città: String) : void
+ setVia(via: String) : void
+ setPaese(paese: String) : void
+ setProvincia(provincia: String) : void
+ setCAP: String(cap: String) : void
+ setData_ordine(data_ordine : Date) : void
+ setData_arrivo(data_arrivo : Date) : void
+ setId(id : String) : void
+ setPrezzo(prezzo : BigDecimal) : void
+ setStato(stato : String) : void
+ setCliente(Cliente : String) : void
+ setRigaOrdine(rigaOrdine: Collection<RigaOrdine> : void

```

```

context Prodotto::getNomeRicevente( )
    post: result = nomeRicevente
context Prodotto::getCognomeRicevente( )
    post: result = cognomeRicevente
context Prodotto::getEmail( )
    post: result = email
context Prodotto::getCellulare( )
    post: result = cellulare
context Prodotto::getNcivico( )
    post: result = ncivico
context Prodotto::getCittà( )
    post: result = città
context Prodotto::getVia( )
    post: result = via
context Prodotto::getPaese( )
    post: result = paese
context Prodotto::getProvincia( )
    post: result = provincia
context Prodotto::getCAP( )
    post: result = cap
context Prodotto::getData_ordine( )
    post: result = data_ordine
context Prodotto::getData_arrivo( )
    post: result = data_arrivo
context Prodotto::getId( )

```

```

        post: result = id
context Prodotto::getPrezzo( )
        post: result = prezzo
context Prodotto::getStato( )
        post: result = stato
context Prodotto::getCliente( )
        post: result = cliente
context Prodotto::getRigaOrdine ( )
        post: result = rigaOrdine
Context Prodotto::setNomeRicevente(nomeRicevente)
        post: self.nomeRicevente = nomeRicevente
Context Prodotto::setCognomeRicevente(cognomeRicevente)
        post: self.cognomeRicevente = cognomeRicevente
Context Prodotto::setEmail(email)
        post: self.email = email
Context Prodotto::setCellulare(cellulare)
        post: self.cellulare = cellulare
Context Prodotto::setNcivico(ncivico)
        post: self.ncivico = ncivico
Context Prodotto::setCittà(città)
        post: self.città = città
Context Prodotto::setVia(via)
        post: self.via = via
Context Prodotto::setPaese(paese)
        post: self.paese = paese
Context Prodotto::setProvincia(provincia)
        post: self.provincia = provincia
Context Prodotto::setCAP(cap)
        post: self.cap = cap
Context Prodotto::setData_ordine(data_ordine)
        post: self.data_ordine = data_ordine
Context Prodotto::setData_arrivo(data_arrivo)
        post: self.data_arrivo = data_arrivo
Context Prodotto::setId(id)
        post: self.id = id
Context Prodotto::setPrezzo(prezzo)
        post: self.prezzo = prezzo
Context Prodotto::setStato(stato)
        post: self.stato = stato
Context Prodotto::setCliente(cliente)
        post: self.cliente = cliente
Context Prodotto::setRigaOrdine(rigaOrdine)
        post: self.rigaOrdine = rigaOrdine

```

OrdineModel
//
+ doRetrieveByKey(key : String) : Ordine

```
+ doRetrieveAll(order : String) : Collection<Ordine>
+ doSave(ordine : Ordine bean) : void
+ doUpdate(ordine : Ordine bean) : void
+ doDelete(ordine : Ordine bean) : void
+ doSaveCheck(ordine : Ordine bean) : boolean
+ doUpdateStatus(ordine : Ordine bean, millis : long) : Boolean
+ doRetrieveAllDaSpedire() : Collection<Ordine>
```

context OrdineModel::doRetrieveByKey(key)

pre: key != null

post: result = database.Ordine -> select (o | ordine.key = o.key)

context OrdineModel::doRetrieveAll()

post: result = database.Ordine

context OrdineModel::doSave(ordine)

pre: ordine.nomeRicevente!=null and ordine.cognomeRicevente!=null and ordine.email!=null and ordine.cellulare!=null and ordine.city!=null and ordine.ncivico!=null and ordine.via!=null and ordine.paese!=null and ordine.provincia!=null and ordine.cap!=null and ordine.cliente!=null and ordine.data_ordine!=null and ordine.stato!=null and ordine.id!=null and ordine.prezzo!=null

post: database.Ordine -> exists(o | ordine.nomeRicevente = o.nomeRicevente and ordine.cognomeRicevente = o.cognomeRicevente and ordine.email = o.email and ordine.cellulare = o.cellulare and ordine.city = o.city and ordine.ncivico = o.ncivico and ordine.via = o.via and ordine.paese = o.paese and ordine.provincia = o.provincia and ordine.cap = o.cap and ordine.cliente = o.cliente and ordine.data_ordine = o.data_ordine and ordine.stato = o.stato and ordine.id = o.id and ordine.prezzo = o.prezzo)

context OrdineModel::doUpdate(ordine)

pre: ordine.nomeRicevente!=null and ordine.cognomeRicevente!=null and ordine.email!=null and ordine.cellulare!=null and ordine.city!=null and ordine.ncivico!=null and ordine.via!=null and ordine.paese!=null and ordine.provincia!=null and ordine.cap!=null and ordine.cliente!=null and ordine.data_ordine!=null and ordine.stato!=null and ordine.id!=null and ordine.prezzo!=null

post: database.Ordine -> exists(o | ordine.nomeRicevente = o.nomeRicevente and ordine.cognomeRicevente = o.cognomeRicevente and ordine.email = o.email and ordine.cellulare = o.cellulare and ordine.city = o.city and ordine.ncivico = o.ncivico and ordine.via = o.via and ordine.paese = o.paese and ordine.provincia = o.provincia and ordine.cap = o.cap and ordine.cliente = o.cliente and ordine.data_ordine = o.data_ordine and ordine.stato = o.stato and ordine.id = o.id and ordine.prezzo = o.prezzo)

context OrdineModel::doDelete(ordine)

pre: ordine.databaseOrdine -> exists(id | ordine.id = id)

post: not(ordine.databaseOrdine -> exists(id | ordine.id = id))

context OrdineModel::doSaveCheck(ordine)

pre: ordine.nomeRicevente!=null and ordine.cognomeRicevente!=null and ordine.email!=null and ordine.cellulare!=null and ordine.city!=null and ordine.ncivico!=null and ordine.via!=null and ordine.paese!=null and ordine.provincia!=null and ordine.cap!=null and ordine.cliente!=null and ordine.data_ordine!=null and ordine.stato!=null and ordine.id!=null and ordine.prezzo!=null

post: result = database.Ordine -> exists(o | ordine.nomeRicevente = o.nomeRicevente and ordine.cognomeRicevente = o.cognomeRicevente and ordine.email = o.email and ordine.cellulare = o.cellulare and ordine.city = o.city and ordine.ncivico = o.ncivico and ordine.via = o.via and ordine.paese = o.paese and ordine.provincia = o.provincia and ordine.cap = o.cap and ordine.cliente = o.cliente and ordine.data_ordine = o.data_ordine and ordine.stato = o.stato and ordine.id = o.id and ordine.prezzo = o.prezzo))

context OrdineModel::doUpdateStatus(ordine, millis)
pre: ordine.id!=null and ordine.millis>=172800000 and ordine.millis<=864000000
post: : ordine.databaseOrdine -> exists(id | ordine.id = id) and exists(millis | ordine.millis = millis)
context OrdineModel::doRetrieveAllDaSpedire()
post: result = database.Ordine

RigaOrdine
- quantità: Int - prezzo_Al_Pezzo: Int
+ getQuantità() : Int + getPrezzo_Al_Pezzo() : Int + setQuantità(Quantità : Int) : void + setPrezzo_Al_pezzo(prezzo_al_pezzo : Int) : void

context Prodotto::getQuantità()
post: result = quantità
context Prodotto::getPrezzo_Al_pezzo()
post: result = prezzo_al_pezzo
Context Prodotto::setQuantità(quantità)
post: self.quantità = quantità
Context Prodotto::setPrezzo_Al_Pezzo(prezzo_al_pezzo)
post: self.prezzo_al_pezzo = prezzo_al_pezzo

RigaOrdineModel
//
+ doRetrieveAllByOrder(orderID : String) : Collection<RigaOrdine> + doSave(rigaOrdine : RigaOrdine bean) : boolean

context RigaOrdineModel::doRetrieveAllByOrder(orderID)
pre: orderID!=null
post: result = database.Ordine
context RigaOrdineModel::doSave(rigaOrdine)
pre: : ordine.prodotto!=null and ordine.ordine!=null and ordine.quantity!=null and ordine.prezzo_al_pezzo!=null
post: database.Ordine -> exists(o | ordine.prodotto = o.prodotto and ordine.ordine = o.ordine and ordine.quantity = o.quantity and ordine.prezzo_al_pezzo = o.prezzo_al_pezzo)