

```

Methode .Modelle.ULager.Methoden.InKetteEinhaengen
-- Name: InKetteEinhaengen
-- Funktion: Hängt die Sendungen in die jeweilige Kette ein
-- Aufrufendes Objekt: InKetteEinhaengen in den TorBereichen
is
i,x : integer;
KapMax : integer;
KapMom, neuesBE : object;
do

for i:=1 to .Modelle.ULager.Tabellen.EinhaengePunkteKette.yDim loop
if .Modelle.ULager.Tabellen.EinhaengePunkteKette[1,i]=?.Standort.Name then

----- Kette1-----
if .Modelle.ULager.Tabellen.EinhaengePunkteKette[2,i]=.Modelle.ULager.Kettel then
KapMax:=.Modelle.ULager.Tabellen.Kettel[3,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]];

-- Weist auf die globale Variable der aktuellen Bestandszahl des Bereiches hin
KapMom:=str_to_obj(to_str(".Modelle.ULager.Methoden.KettelBereich",.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]));

-- Überprüfung der Füllmenge des Bereiches am Einhängpunkt und der Gesamtkapazität der Kettel
if (KapMom.Wert >= KapMax) OR (.Modelle.ULager.Kettel.AnzahlBES =.Modelle.ULager.Kettel.Kapazitaet-32) then

-- Warten, bis beide Bedingungen erfüllt sind
waituntil (KapMom.Wert < KapMax) AND (.Modelle.ULager.Kettel.AnzahlBES < .Modelle.ULager.Kettel.Kapazitaet-32) prio 1;

-- BE wird aus dem TorBereich auf die Kettel umgelagert
while neuesBE = void loop
neuesBE:=?.Inhalt.Klasse.erzeugen(.Modelle.ULager.Tabellen.EinhaengePunkteKette[2,i],.Modelle.ULager.Tabellen.EinhaengePunkteKette[3,i]);
wait 1;
end;
neuesBE.Beschaedigt:=?.Inhalt.Beschaedigt;
neuesBE.PLZ:=?.Inhalt.PLZ;
neuesBE.Sendungsnummer:=?.Inhalt.Sendungsnummer;
neuesBE.Ziel:=?.Inhalt.Ziel;
neuesBE.Zielrelation:=?.Inhalt.Zielrelation;

-- Sendungsnummer wird in der Inventarliste eingetragen
.Modelle.ULager.Tabellen.Kettel[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]][1,.Modelle.ULager.Tabellen.Kettel[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]].yDim+1]:=?.Inhalt.Sendungsnummer;
.Modelle.ULager.Tabellen.Kettel[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]][2,.Modelle.ULager.Tabellen.Kettel[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]].yDim]:=?.Inhalt.Ziel;
?.Inhalt.vernichten;

-- Anzahl der vorhandenen BE im Bereich wird um 1 erhöht
KapMom.Wert:=KapMom.Wert+1;

else
-- BE wird aus dem TorBereich auf die Kettel umgelagert
while neuesBE = void loop
neuesBE:=?.Inhalt.Klasse.erzeugen(.Modelle.ULager.Tabellen.EinhaengePunkteKette[2,i],.Modelle.ULager.Tabellen.EinhaengePunkteKette[3,i]);
wait 1;
end;
neuesBE.Beschaedigt:=?.Inhalt.Beschaedigt;
neuesBE.PLZ:=?.Inhalt.PLZ;
neuesBE.Sendungsnummer:=?.Inhalt.Sendungsnummer;
neuesBE.Ziel:=?.Inhalt.Ziel;
neuesBE.Zielrelation:=?.Inhalt.Zielrelation;

-- Sendungsnummer wird in der Inventarliste eingetragen
.Modelle.ULager.Tabellen.Kettel[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]][1,.Modelle.ULager.Tabellen.Kettel[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]].yDim+1]:=?.Inhalt.Sendungsnummer;
.Modelle.ULager.Tabellen.Kettel[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]][2,.Modelle.ULager.Tabellen.Kettel[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]].yDim]:=?.Inhalt.Ziel;
?.Inhalt.vernichten;

-- Anzahl der vorhandenen BE im Bereich wird um 1 erhöht
KapMom.Wert:=KapMom.Wert+1;
end;

----- Kette2 -----
elseif .Modelle.ULager.Tabellen.EinhaengePunkteKette[2,i]=.Modelle.ULager.Kette2 then
KapMax:=.Modelle.ULager.Tabellen.Kette2[3,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]];

-- Weist auf die globale Variable der aktuellen Bestandszahl des Bereiches hin
KapMom:=str_to_obj(to_str(".Modelle.ULager.Methoden.Kette2Bereich",.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]));

-- Überprüfung der Füllmenge des Bereiches am Einhängpunkt und der Gesamtkapazität der Kette2
if (KapMom.Wert >= KapMax) OR (.Modelle.ULager.Kette2.AnzahlBES=.Modelle.ULager.Kette2.Kapazitaet-38) then

-- Warten, bis beide Bedingungen erfüllt sind
waituntil (KapMom.Wert < KapMax) AND (.Modelle.ULager.Kette2.AnzahlBES < .Modelle.ULager.Kette2.Kapazitaet-38) prio 1;

-- BE wird aus dem TorBereich auf die Kette2 umgelagert
while neuesBE = void loop
neuesBE:=?.Inhalt.Klasse.erzeugen(.Modelle.ULager.Tabellen.EinhaengePunkteKette[2,i],.Modelle.ULager.Tabellen.EinhaengePunkteKette[3,i]+x);
wait 1;
end;
neuesBE.Beschaedigt:=?.Inhalt.Beschaedigt;
neuesBE.PLZ:=?.Inhalt.PLZ;
neuesBE.Sendungsnummer:=?.Inhalt.Sendungsnummer;
neuesBE.Ziel:=?.Inhalt.Ziel;
neuesBE.Zielrelation:=?.Inhalt.Zielrelation;

```

```

-- Sendungsnummer wird in der Inventarliste eingetragen
.Modelle.ULager.Tabellen.Kette2[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]][1,.Modelle.ULager.Tabellen.Kette2[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]].yDim+1]:=?.Inhalt.Sendungsnummer;
.Modelle.ULager.Tabellen.Kette2[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]][2,.Modelle.ULager.Tabellen.Kette2[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]].yDim]:=?.Inhalt.Ziel;
?.Inhalt.vernichten;

-- Anzahl der vorhandenen BE im Bereich wird um 1 erhöht
KapMom.Wert:=KapMom.Wert+1;
else
-- BE wird aus dem TorBereich auf die Kette2 umgelagert
while neuesBE = void loop
    neuesBE:=?.Inhalt.Klasse.erzeugen(.Modelle.ULager.Tabellen.EinhaengePunkteKette[2,i],.Modelle.ULager.Tabellen.EinhaengePunkteKette[3,i]);
    wait 1;
end;
neuesBE.Beschaedigt:=?.Inhalt.Beschaedigt;
neuesBE.PLZ:=?.Inhalt.PLZ;
neuesBE.Sendungsnummer:=?.Inhalt.Sendungsnummer;
neuesBE.Ziel:=?.Inhalt.Ziel;
neuesBE.Zielrelation:=?.Inhalt.Zielrelation;

-- Sendungsnummer wird in der Inventarliste eingetragen
.Modelle.ULager.Tabellen.Kette2[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]][1,.Modelle.ULager.Tabellen.Kette2[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]].yDim+1]:=?.Inhalt.Sendungsnummer;
.Modelle.ULager.Tabellen.Kette2[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]][2,.Modelle.ULager.Tabellen.Kette2[2,.Modelle.ULager.Tabellen.EinhaengePunkteKette[4,i]].yDim]:=?.Inhalt.Ziel;
?.Inhalt.vernichten;

-- Anzahl der vorhandenen BE im Bereich wird um 1 erhöht
KapMom.Wert:=KapMom.Wert+1;
end;
exitloop;
end;
end;
next;
end;

```

**Methode .Modelle.ULager.Methoden.KettenSteuerung**

```
-- Name: KettenSteuerung
-- Funktion: Steuert das Verhalten der Sendungen auf der Kette
-- Aufrufendes Objekt: Sendung auf Kette an den Sensoren

(sensorID : integer) -- Übergabe der SensorID
is
  i, j, x : integer;
  KapMom : object;
  virtuellesBE : object;
do

----- Kette 1 -----
  if @.Standort.Name=.Modelle.ULager.Kettel.Name then

-- Löschen des auftragerzeugenen BEs auf der Arbeitsstation
  if sensorID=1 then
    for i:=1 to .Modelle.ULager.Tabellen.Kettel[2,8].yDim loop
      -- Suche, ob die Sendung im Teilbereich enthalten ist
      if .Modelle.ULager.Tabellen.Kettel[2,8][1,i]=@.Sendungsnummer then
        j:=1;
        -- Suche, ob das Sendungsziel im Teilbereich enthalten ist
        while .Modelle.ULager.Tabellen.ZieleBereicheKettel[1,j]=8 loop
          if .Modelle.ULager.Tabellen.ZieleBereicheKettel[2,j]=@.Ziel then
            if .Modelle.ULager.Tabellen.ZieleBereicheKettel[4,j].belegt AND .Modelle.ULager.Tabellen.ZieleBereicheKettel[4,j].Inhalt.Sendungsnummer=@.Sendungsnummer then
              -- Löschen des virtuellen Collis auf der Arbeitsstation
              .Modelle.ULager.Tabellen.ZieleBereicheKettel[4,j].Inhalt.vernichten;
              exitloop;
            end;
          end;
          j:=j+1;
        end;
      end;
    next;
  else
    for i:=1 to .Modelle.ULager.Tabellen.Kettel[2,sensorID-1].yDim loop
      -- Suche, ob die Sendung im Teilbereich enthalten ist
      if .Modelle.ULager.Tabellen.Kettel[2,sensorID-1][1,i]=@.Sendungsnummer then

        -- Suche nach der Startposition in der Tabelle
        for x:=1 to .Modelle.ULager.Tabellen.ZieleBereicheKettel.yDim loop
          if .Modelle.ULager.Tabellen.ZieleBereicheKettel[1,x]=sensorID then
            j:=x;
            exitloop;
          end;
        next;

        -- Suche, ob das Sendungsziel im Teilbereich enthalten ist
        while .Modelle.ULager.Tabellen.ZieleBereicheKettel[1,j]=sensorID-1 loop
          if .Modelle.ULager.Tabellen.ZieleBereicheKettel[2,j]=@.Ziel then
            if .Modelle.ULager.Tabellen.ZieleBereicheKettel[4,j].belegt AND .Modelle.ULager.Tabellen.ZieleBereicheKettel[4,j].Inhalt.Sendungsnummer=@.Sendungsnummer then
              -- Löschen des virtuellen Collis auf der Arbeitsstation
              .Modelle.ULager.Tabellen.ZieleBereicheKettel[4,j].Inhalt.vernichten;
              exitloop;
            end;
          end;
          j:=j+1;
        end;
      end;
    next;
  end;

-- Löschen des BEs aus der Inventarliste des alten Bereiches
  if sensorID=1 then -- gilt nur für den ersten Bereich - muss auf Bereich 8 zugreifen
    for i:=1 to .Modelle.ULager.Tabellen.Kettel[2,8].yDim loop
      if .Modelle.ULager.Tabellen.Kettel[2,8][1,i]=@.Sendungsnummer then
        -- Sendungsnummer wird aus der Inventarliste des Kettenbereichs 8 gelöscht
        .Modelle.ULager.Tabellen.Kettel[2,8].entferneZeile(i);
        -- Anzahl der BEs im Kettenbereich8 wird um 1 reduziert
        .Modelle.ULager.Methoden.KettelBereich8:=.Modelle.ULager.Methoden.KettelBereich8-1;
        exitloop;
      end;
    next;
  else
    for i:=1 to .Modelle.ULager.Tabellen.Kettel[2,sensorID-1].yDim loop
      if .Modelle.ULager.Tabellen.Kettel[2,sensorID-1][1,i]=@.Sendungsnummer then
        -- Sendungsnummer wird aus der Inventarliste des Kettenbereichs gelöscht
        .Modelle.ULager.Tabellen.Kettel[2,sensorID-1].entferneZeile(i);
        -- Anzahl der BEs im Kettenbereich wird um 1 reduziert
        KapMom:=str_to_obj(to_str(".Modelle.ULager.Methoden.KettelBereich",sensorID-1));
        KapMom.Wert:=KapMom.Wert-1;
        exitloop;
      end;
    next;
  end;

-- Einfügen des BEs in die Inventarliste des neuen Bereiches
  if sensorID=1 then
    .Modelle.ULager.Tabellen.Kettel[2,1].einfuegenZeile(1);
    .Modelle.ULager.Tabellen.Kettel[2,1][1,1]=@.Sendungsnummer;
    .Modelle.ULager.Tabellen.Kettel[2,1][2,1]=@.Ziel;
    -- Anzahl der BEs im Kettenbereich 1 wird um 1 erhöht
```

```

        .Modelle.ULager.Methoden.KettelBereich1:=.Modelle.ULager.Methoden.KettelBereich1+1;
    else
        .Modelle.ULager.Tabellen.Kettel[2,sensorID].einfuegenZeile(1);
        .Modelle.ULager.Tabellen.Kettel[2,sensorID][1,1]:=@.Sendungsnummer;
        .Modelle.ULager.Tabellen.Kettel[2,sensorID][2,1]:=@.Ziel;
        -- Anzahl der BEs im Kettenbereich wird um 1 erhöht
        KapMom:=str_to_obj(to_str(".Modelle.ULager.Methoden.KettelBereich",sensorID));
        KapMom.Wert:=KapMom.Wert+1;
    end;

-- Erzeugen des BEs auf der Arbeitsstation, wenn das Ziel in diesem Teilbereich der Kette liegt
if sensorID=1 then
    j:=1;
    -- Erzeugen des BEs auf der Zielstation
    while .Modelle.ULager.Tabellen.ZieleBereicheKettel[1,j]=1 loop
        if .Modelle.ULager.Tabellen.ZieleBereicheKettel[2,j]=@.Ziel then
            -- Überprüfung, ob Zielstation zurzeit nicht belegt ist
            if .Modelle.ULager.Tabellen.ZieleBereicheKettel[4,j].belegt=false then
                virtuellesBE:=@.Klasse.erzeugen(.Modelle.ULager.Tabellen.ZieleBereicheKettel[3,j]);
                virtuellesBE.Sendungsnummer:=@.Sendungsnummer;
                exitloop;
            end;
        end;
        j:=j+1;
    end;
else
    -- Suche nach der Startposition in der Tabelle
    for i:=1 to .Modelle.ULager.Tabellen.ZieleBereicheKettel.yDim loop
        if .Modelle.ULager.Tabellen.ZieleBereicheKettel[1,i]=sensorID then
            j:=i;
            exitloop;
        end;
    next;
    -- Erzeugen des BEs auf der Zielstation
    while .Modelle.ULager.Tabellen.ZieleBereicheKettel[1,j]=sensorID loop
        if .Modelle.ULager.Tabellen.ZieleBereicheKettel[2,j]=@.Ziel then
            -- Überprüfung, ob Zielstation zurzeit nicht belegt ist
            if .Modelle.ULager.Tabellen.ZieleBereicheKettel[4,j].belegt=false then
                virtuellesBE:=@.Klasse.erzeugen(.Modelle.ULager.Tabellen.ZieleBereicheKettel[3,j]);
                virtuellesBE.Sendungsnummer:=@.Sendungsnummer;
                exitloop;
            end;
        end;
        j:=j+1;
    end;
end;

----- Kette 2 -----
elseif @.Standort.Name=.Modelle.ULager.Kette2.Name then

-- Löschen des auftragerzeugenen BEs auf der Arbeitsstation
if sensorID=1 then
    for i:=1 to .Modelle.ULager.Tabellen.Kette2[2,10].yDim loop
        -- Suche, ob die Sendung im Teilbereich enthalten ist
        if .Modelle.ULager.Tabellen.Kette2[2,10][1,i]=@.Sendungsnummer then
            j:=1;
            -- Suche, ob das Sendungsziel im Teilbereich enthalten ist
            while .Modelle.ULager.Tabellen.ZieleBereicheKette2[1,j]=10 loop
                if .Modelle.ULager.Tabellen.ZieleBereicheKette2[2,j]=@.Ziel then
                    if .Modelle.ULager.Tabellen.ZieleBereicheKette2[4,j].belegt=true AND .Modelle.ULager.Tabellen.ZieleBereicheKette2[4,j].Inhalt.Sendungsnummer=@.Sendungsnummer then
                        -- Löschen des virtuellen Collis auf der Arbeitsstation
                        .Modelle.ULager.Tabellen.ZieleBereicheKette2[4,j].Inhalt.vernichten;
                        exitloop;
                    end;
                end;
                j:=j+1;
            end;
        end;
    next;
else
    for i:=1 to .Modelle.ULager.Tabellen.Kette2[2,sensorID-1].yDim loop
        -- Suche, ob die Sendung im Teilbereich enthalten ist
        if .Modelle.ULager.Tabellen.Kette2[2,sensorID-1][1,i]=@.Sendungsnummer then
            j:=1;
            -- Suche nach der Startposition in der Tabelle
            for x:=1 to .Modelle.ULager.Tabellen.ZieleBereicheKette2.yDim loop
                if .Modelle.ULager.Tabellen.ZieleBereicheKette2[1,x]=sensorID-1 then
                    j:=x;
                    exitloop;
                end;
            next;

            -- Suche, ob das Sendungsziel im Teilbereich enthalten ist
            while .Modelle.ULager.Tabellen.ZieleBereicheKette2[1,j]=sensorID-1 loop
                if .Modelle.ULager.Tabellen.ZieleBereicheKette2[2,j]=@.Ziel then
                    if .Modelle.ULager.Tabellen.ZieleBereicheKette2[4,j].belegt=true AND .Modelle.ULager.Tabellen.ZieleBereicheKette2[4,j].Inhalt.Sendungsnummer=@.Sendungsnummer then
                        -- Löschen des virtuellen Collis auf der Arbeitsstation
                        .Modelle.ULager.Tabellen.ZieleBereicheKette2[4,j].Inhalt.vernichten;
                        exitloop;
                    end;
                end;
                j:=j+1;
            end;
        end;
    next;
end;

```



## Methode .Modelle.ULager.Methoden.AusKetteAushaengen

```
-- Name: AusKetteAushaengen
-- Funktion: Lagert die realen BEs nach der Bearbeitung in den TorBereich um und löscht die virtuellen BEs auf den Bausteinen
-- Aufrufendes Objekt: Ausgangsteuerung AusKette in den TorBereichen
is
  i,j,x : integer;
  Sendungsnummer, KettenBereich : integer;
  enthalten : boolean;
  Klasse, KapMom : object;
  Inhaltsliste : table;
do
  enthalten:=false;

----- Kette 1 -----
  for i:=1 to .Modelle.ULager.Tabellen.ZieleBereicheKettel.yDim loop
    -- Suche, ob das aufrufende BE sich in der Kette 1 befindet
    if .Modelle.ULager.Tabellen.ZieleBereicheKettel[2,i]=@.Standort.~ then
      -- Sicherung des Kettenbereichs, Sendungsnummer und der Klasse
      KettenBereich:=.Modelle.ULager.Tabellen.ZieleBereicheKettel[1,i];
      Sendungsnummer:=@.Sendungsnummer;
      Klasse:=@.Klasse;

      -- Suche der Sendung den Inventarlisten, löschen der Zeile in der Liste
      for j:=1 to .Modelle.ULager.Tabellen.Kettel[2,KettenBereich].yDim loop
        if .Modelle.ULager.Tabellen.Kettel[2,KettenBereich][1,j]=Sendungsnummer then
          .Modelle.ULager.Tabellen.Kettel[2,KettenBereich].entferneZeile(j);
          -- Verringerung der KapMom um 1
          KapMom:=str_to_obj(to_str(".Modelle.ULager.Methoden.KettelBereich",KettenBereich));
          KapMom.Wert:=KapMom.Wert-1;
          exitloop;
        end;
      next;

      -- Reale BE wird umgelagert
      Inhaltsliste.create;
      Inhaltsliste.setzeDatenTyp(1,"object");
      Inhaltsliste.setzeDatenTyp(2,"length");
      Inhaltsliste.setzeDatenTyp(3,"length");
      .Modelle.ULager.Kettel.Inhaltsliste(Inhaltsliste);
      for x:=1 to Inhaltsliste.yDim loop
        if Inhaltsliste[1,x].Sendungsnummer=@.Sendungsnummer then
          Inhaltsliste[1,x].umlagern(@.Standort.~.Stellplatz);
          exitloop;
        end;
      next;
    end;
  end;
end;
```

```

        end;
    next;

    -- Virtuelles BE wird gelöscht
    @.vernichten;
    enthalten:=true;
    exitloop;
end;
next;

```

----- Kette2 -----

```

if enthalten=false then
    for i:=1 to .Modelle.ULager.Tabellen.ZieleBereicheKette2.yDim loop
        -- Suche, ob das aufrufende BE sich in der Kette 1 befindet
        if .Modelle.ULager.Tabellen.ZieleBereicheKette2[2,i]=@.Standort.~ then
            -- Sicherung des Kettenbereichs, Sendungsnummer und der Klasse
            KettenBereich:=.Modelle.ULager.Tabellen.ZieleBereicheKette2[1,i];
            Sendungsnummer:=@.Sendungsnummer;
            Klasse:=@.Klasse;

            -- Suche der Sendung den Inventarlisten, löschen der Zeile in der Liste
            for j:=1 to .Modelle.ULager.Tabellen.Kette2[2,KettenBereich].yDim loop
                if .Modelle.ULager.Tabellen.Kette2[2,KettenBereich][1,j]=Sendungsnummer then
                    .Modelle.ULager.Tabellen.Kette2[2,KettenBereich].entferneZeile(j);
                    -- Verringerung der KapMom um 1
                    KapMom:=str_to_obj(to_str(".Modelle.ULager.Methoden.Kette2Bereich",KettenBereich));
                    KapMom.Wert:=KapMom.Wert-1;
                    exitloop;
                end;
            next;

            -- Reale BE wird umgelagert
            Inhaltsliste.create;
            Inhaltsliste.setzeDatenTyp(1,"object");
            Inhaltsliste.setzeDatenTyp(2,"length");
            Inhaltsliste.setzeDatenTyp(3,"length");
            .Modelle.ULager.Kette2.Inhaltsliste(Inhaltsliste);
            for x:=1 to Inhaltsliste.yDim loop
                if Inhaltsliste[1,x].Sendungsnummer=@.Sendungsnummer then
                    Inhaltsliste[1,x].umlagern(@.Standort.~.Stellplatz);
                    exitloop;
                end;
            next;

```

```
        -- Virtuelles BE wird gelöscht
        @.vernichten;
        enthalten:=true;
        exitloop;
    end;
next;
end;
end;
```