

Tutorial: generic implementation

🌐 Language: [Deutsch](#) · [English](#)

📖 Tutorial: generic implementation

Hands-on tutorial for generic implementation – one abstract BO and one rule for many concrete BOs (number ranges, sum calculation).

TUTORIAL

APPLICATION DEVELOPER

UPDATED: JUL 2026

APPLIES TO NUCLOS 4.2026.X

On this page

- [Example 1: number ranges](#)
- [Example 2: sum calculation in document positions](#)
- [Related pages](#)



Overview

Goal: create a generic business object and implement a generic rule.

Prerequisite: at least two business objects with similar fields (e.g. a running number).

The [generic implementation](#) replaces repeated rule code branched by `instanceof` with **one** abstract BO serving several concrete BOs.

Example 1: number ranges

Several BOs (*quote*, *order*, *delivery note*, *purchase order*) receive an automatic number on creation. Previously solved via `instanceof`:

```
public void insert(InsertContext context) throws BusinessException {
    final Modifiable<?> businessObject = context.getBusinessObject(Modifiable.class);
    if (businessObject instanceof Angebot) {
        ((Angebot) businessObject).setAngebotsnr(NummernkreiseLogik.getNextValue(Angebot.class));
    } else if (businessObject instanceof Auftrag) {
        ((Auftrag) businessObject).setAuftragsnr(NummernkreiseLogik.getNextValue(Auftrag.class));
    } else if (businessObject instanceof Lieferschein) {
        ((Lieferschein) businessObject).setLieferscheinnr(NummernkreiseLogik.getNextValue(Lieferschein.class));
    } else if (businessObject instanceof Bestellung) {
        ((Bestellung) businessObject).setBestellnr(NummernkreiseLogik.getNextValue(Bestellung.class));
    }
}
```

1. Create the generic BO

Create a new generic BO **Nummerierbar** in the [business object wizard](#) with exactly one attribute *Nummer* (no layout/permissions) and assign it to a nucleot.



Data type

The **data type** of the generic attribute must match that of the fields in the implementing BOs.

2. Configure the mapping

For each implementing BO create a record with the attribute mapping (e.g. *Nummer* *order number*). Nuclos generates the class `Nummerierbar.java` with generic getters/setters:

```
import org.nuclos.businessentity.*;
import org.nuclos.api.businessobject.Dependent;
import org.nuclos.api.businessobject.Flag;
import org.nuclos.api.UID;
import org.nuclos.api.businessobject.GenericBusinessObject;
```

```

import java.util.List;
import java.util.ArrayList;
import java.util.HashSet;
/**
 * BusinessObject: Nummerierbar
 *<br>
 *<br>Nuclet: org.nuclet.tulteclico
 *<br>DB-Name: [generic]
 *<br>Editable: true
 *<br>Localized: false
 *<br>Statemodel: false
 */
public class Nummerierbar implements GenericBusinessObject<Long> {
private org.nuclos.api.businessobject.BusinessObject<Long> businessObject;
public Nummerierbar(org.nuclet.tulteclico.Lieferschein businessObject) {
    this.businessObject = businessObject;
}
public Nummerierbar(org.nuclet.tulteclico.Bestellung businessObject) {
    this.businessObject = businessObject;
}
public Nummerierbar(org.nuclet.tulteclico.Auftrag businessObject) {
    this.businessObject = businessObject;
}
public Nummerierbar(org.nuclet.tulteclico.Angebot businessObject) {
    this.businessObject = businessObject;
}
public org.nuclos.api.UID getEntityUid() {
    return businessObject.getEntityUid();
}
public org.nuclos.api.businessobject.BusinessObject<Long> getBusinessObject() {
    return businessObject;
}
public java.lang.String getEntity() {
    return businessObject.getEntity();
}
public java.lang.Long getId() {
    return businessObject.getId();
}
public java.lang.Integer getVersion() {
    return businessObject.getVersion();
}
public boolean isInsert() {
    return businessObject.isInsert();
}
public boolean isUpdate() {
    return businessObject.isUpdate();
}
public boolean isDelete() {
    return businessObject.isDelete();
}
}
/**
 * Getter-Method for attribute: changedAt
 *<br>
 *<br>Entity: Nummerierbar
 *<br>DB-Name: DATCHANGED
 *<br>Data type: org.nuclos.common2.InternalTimestamp
 *<br>Localized: false
 *<br>Output format: null
 *<br>Scale: null
 *<br>Precision: null
 */
public java.util.Date getChangedAt() {
    // all implemented BOs:
    if (this.businessObject instanceof org.nuclet.tulteclico.Lieferschein) {
        return ((org.nuclet.tulteclico.Lieferschein) this.businessObject).getChangedAt();
    }
    if (this.businessObject instanceof org.nuclet.tulteclico.Bestellung) {
        return ((org.nuclet.tulteclico.Bestellung) this.businessObject).getChangedAt();
    }
    if (this.businessObject instanceof org.nuclet.tulteclico.Auftrag) {
        return ((org.nuclet.tulteclico.Auftrag) this.businessObject).getChangedAt();
    }
}

```

```

    }
    if (this.businessObject instanceof org.nuclet.tulteclico.Angebot) {
        return ((org.nuclet.tulteclico.Angebot) this.businessObject).getChangedAt();
    }
    // not implemented attribute:
    return null;
}
/**
 * Getter-Method for attribute: changedBy
 *<br>
 *<br>Entity: Nummerierbar
 *<br>DB-Name: STRCHANGED
 *<br>Data type: java.lang.String
 *<br>Localized: false
 *<br>Output format: null
 *<br>Scale: 255
 *<br>Precision: null
 */
public java.lang.String getChangedBy() {
    // all implemented BOs:
    if (this.businessObject instanceof org.nuclet.tulteclico.Lieferschein) {
        return ((org.nuclet.tulteclico.Lieferschein) this.businessObject).getChangedBy();
    }
    if (this.businessObject instanceof org.nuclet.tulteclico.Bestellung) {
        return ((org.nuclet.tulteclico.Bestellung) this.businessObject).getChangedBy();
    }
    if (this.businessObject instanceof org.nuclet.tulteclico.Auftrag) {
        return ((org.nuclet.tulteclico.Auftrag) this.businessObject).getChangedBy();
    }
    if (this.businessObject instanceof org.nuclet.tulteclico.Angebot) {
        return ((org.nuclet.tulteclico.Angebot) this.businessObject).getChangedBy();
    }
    // not implemented attribute:
    return null;
}
/**
 * Getter-Method for attribute: createdAt
 *<br>
 *<br>Entity: Nummerierbar
 *<br>DB-Name: DATCREATED
 *<br>Data type: org.nuclos.common2.InternalTimestamp
 *<br>Localized: false
 *<br>Output format: null
 *<br>Scale: null
 *<br>Precision: null
 */
public java.util.Date getCreatedAt() {
    // all implemented BOs:
    if (this.businessObject instanceof org.nuclet.tulteclico.Lieferschein) {
        return ((org.nuclet.tulteclico.Lieferschein) this.businessObject).getCreatedAt();
    }
    if (this.businessObject instanceof org.nuclet.tulteclico.Bestellung) {
        return ((org.nuclet.tulteclico.Bestellung) this.businessObject).getCreatedAt();
    }
    if (this.businessObject instanceof org.nuclet.tulteclico.Auftrag) {
        return ((org.nuclet.tulteclico.Auftrag) this.businessObject).getCreatedAt();
    }
    if (this.businessObject instanceof org.nuclet.tulteclico.Angebot) {
        return ((org.nuclet.tulteclico.Angebot) this.businessObject).getCreatedAt();
    }
    // not implemented attribute:
    return null;
}
/**
 * Getter-Method for attribute: createdBy
 *<br>
 *<br>Entity: Nummerierbar
 *<br>DB-Name: STRCREATED
 *<br>Data type: java.lang.String
 *<br>Localized: false
 *<br>Output format: null

```

```

*

```

```

    if (this.businessObject instanceof org.nuclet.tulteclico.Auftrag) {
        ((org.nuclet.tulteclico.Auftrag) this.businessObject).setAuftragsnr(pNumber);
    }
    if (this.businessObject instanceof org.nuclet.tulteclico.Angebot) {
        ((org.nuclet.tulteclico.Angebot) this.businessObject).setAngebotsnr(pNumber);
    }
}
}

```

3. Generic rule

In the [server rule manager](#) create a rule on creation, assign it to all implementing BOs (order/conditions as usual) and, instead of instanceof, fetch the generic BO from the context:

```

public void insert(InsertContext context) throws BusinessException {
    Nummerierbar n = context.getBusinessObject(Nummerierbar.class);
    n.setNummer(NummernkreiseLogik.getNextValue(n.getEntityUid()));
}

```

Example 2: sum calculation in document positions

Quote, order and invoice each have a subform with positions and the rule $\text{total} = \text{unit price} \times \text{quantity}$. Instead of doing this per BO, it is solved generically.

Two generic BOs

- **Beleg** (document, totals),
- **Belegposition** (unit price, quantity, total + reference to the document).

After mapping, Nuclos generates `Beleg.java` and `Belegposition.java`. The latter maps e.g. `getMenge()` to `getPos()` of the respective concrete BO and provides `getBelegposition()` as a list.

Generic rule

```

public void save(SaveContext context) throws BusinessException {
    Belegposition pos = context.getBusinessObject(Belegposition.class);
    BigDecimal einzel = pos.getEinzelpreis();
    Integer menge = pos.getMenge();
    if (einzel != null && menge != null) {
        pos.setGesamtpreis(einzel.multiply(new BigDecimal(menge)));
    }
}

```



One code for all

The rule is assigned to all relevant position BOs – one implementation covers all document types.

Related pages



Generic implementation

Concept.

[Open](#)



Server rule manager

Rules.

[Open](#)

