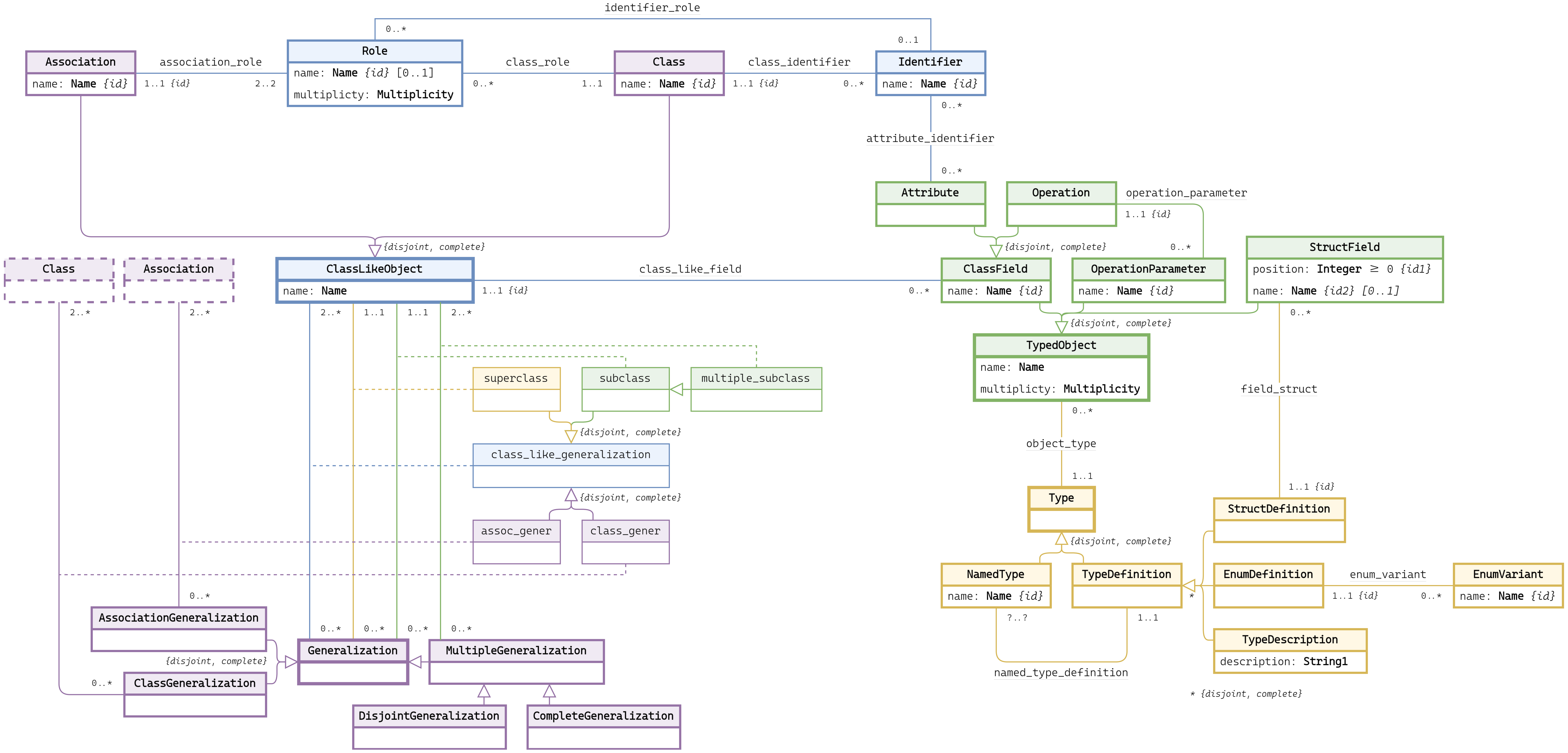
# Logic Flows

First Order Logic based domain design

## Contents

## 1 Introduction

## 2 UML class diagram



## 3 Data types specification

Name = String matching `/^[A-Za-z][A-Za-z\d]*$/`
Multiplicity = (min: Integer $\geq 0$, max: Integer $\geq 0$ [0..1])

### 3.1 Multiplicity

C.Multiplicity.min_less_than_max

```
∀ multiplicity, mult_min, mult_max
    (
        Multiplicity(multiplicity) ∧
        min(multiplicity, mult_min) ∧
        max(multiplicity, mult_max)
    ) →
        mult_min ≤ mult_max
```

## 4 Classes specification

### 4.1 Association

If both roles of an association are connected to the same class, then these roles must have names, and their names must be different.

C.Association.same_class_association_mandatory_and_different_role_names

```
∀ association, class, role1, role2
    (
        association_role(association, role1) ∧
        association_role(association, role2) ∧
        class_role(class, role1) ∧
        class_role(class, role2)
    ) →
        ∃ name1, name2
            name(role1, name1) ∧
            name(role2, name2) ∧
            name1 ≠ name2
```

### 4.2 Attribute

Normally the cycle in the diagram could be removed, but identifiers are required to be unique in each class.

C.Attribute.identifier_in_class

```
∀ class, attribute, identifier
    (
        attribute_identifier(attribute, identifier)
        class_like_field(class, attribute)
    ) →
        class_identifier(class, identifier)
```

## 4.3 Role

C.Role.identifier_in_class

```
∀ class, role, identifier
    (
        identifier_role(identifier, identifier)
        class_role(class, role)
    ) →
        class_identifier(class, identifier)
```