

Class

```
class(c1, "bfo:Quality")
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(c1, "bfo:Quality")
@enduml
```

```
oClass(bfo:Quality)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(bfo:Quality)
@enduml
```

Individual

```
individual(o1, "ns1:temperature1")
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
individual(o1, "ns1:temperature1")
@enduml
```

```
oIndividual(ns1:temperature1)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oIndividual(ns1:temperature1)
@enduml
```

Complement of a class

```
class(c1, "iof:SpatialRegion")
complement(cx, c1)
class(c2, "iof:MaterialArtifact")
equivalent(c2, cx)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(c1, "iof:SpatialRegion")
complement(cx, c1)
class(c2, "iof:MaterialArtifact")
equivalent(c2, cx)
@enduml
```

Type of an individual

```
class(c1, "bfo:Quality")
individual(o1, "ns1:temperature1")
typeOf(o1, c1)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(c1, "bfo:Quality")
individual(o1, "ns1:temperature1")
typeOf(o1, c1)
@enduml
```

```
oClass(bfo:Quality)
oIndividual(ns1:temperature1)
typeOf(ns1:temperature1, bfo:Quality)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(bfo:Quality)
```

```
oIndividual(ns1:temperature1)
typeOf(ns1:temperature1, bfo:Quality)
@enduml
```

Associating individuals using object property

```
individual(p1, "ns1:plank1")
individual(l1, "ns1:length1")
property(p1, bfo:bears, l1, right)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
individual(p1, "ns1:plank1")
individual(l1, "ns1:length1")
property(p1, bfo:bears, l1, right)
@enduml
```

```
oIndividual(ns1:plank1)
oIndividual(ns1:length1)
property(ns1:plank1, bfo:bears, ns1:length1, right)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oIndividual(ns1:plank1)
oIndividual(ns1:length1)
property(ns1:plank1, bfo:bears, ns1:length1, right)
@enduml
```

Associating data to an individual

```
individual(l1, "ns1:length-value")
data(l1, iof:hasSimpleExpressionValue, 30^^xsd:decimal)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
individual(l1, "ns1:length-value")
```

```
data(l1, iof:hasSimpleExpressionValue, 30^^xsd:decimal)
@enduml
```

```
oIndividual(ns1:length-value)
data(ns1:length-value, iof:hasSimpleExpressionValue, 30^^xsd:decimal)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oIndividual(ns1:length-value)
data(ns1:length-value, iof:hasSimpleExpressionValue, 30^^xsd:decimal)
@enduml
```

Same individuals

```
oIndividual(ns1:smith)
oIndividual(ns1:simpson)
sameAs(ns1:smith, ns1:simpson)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oIndividual(ns1:smith)
oIndividual(ns1:simpson)
sameAs(ns1:smith, ns1:simpson)
@enduml
```

Different individuals

```
oIndividual(ns1:apple-as-fruit)
oIndividual(ns1:apple-as-company)
differentFrom(ns1:apple-as-fruit, ns1:apple-as-company)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oIndividual(ns1:apple-as-fruit)
oIndividual(ns1:apple-as-company)
differentFrom(ns1:apple-as-fruit, ns1:apple-as-company)
@enduml
```

All different individuals

```
oIndividual(ns1:apple-as-fruit)
oIndividual(ns1:apple-as-company)
oIndividual(ns1:apple-as-logo)
allDifferent(['"ns1:apple-as-fruit"', "ns1:apple-as-company", "ns1:apple-as-
logo"]')
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oIndividual(ns1:apple-as-fruit)
oIndividual(ns1:apple-as-company)
oIndividual(ns1:apple-as-logo)
allDifferent(['"ns1:apple-as-fruit"', "ns1:apple-as-company", "ns1:apple-as-
logo"]')
@enduml
```

Subclass

```
oClass(bfo:Quality)
oClass(iof:Temperature)
subClass(iof:Temperature, bfo:Quality)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(bfo:Quality)
oClass(iof:Temperature)
subClass(iof:Temperature, bfo:Quality)
@enduml
```

Equivalent classes

```
oClass(iof:Temperature)
oClass(qk:Temperature)
equivalent(iof:Temperature, qk:Temperature)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:Temperature)
oClass(qk:Temperature)
equivalent(iof:Temperature, qk:Temperature)
@enduml
```

Disjoint classes

```
oClass(bfo:Continuant)
oClass(bfo:Occurrent)
disjoint(bfo:Continuant, bfo:Occurrent)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(bfo:Continuant)
oClass(bfo:Occurrent)
disjoint(bfo:Continuant, bfo:Occurrent)
@enduml
```

All disjoint classes

```
oClass(bfo:FiatPoint)
oClass(bfo:FiatLine)
oClass(bfo:FiatSurface)
allDisjoint(['"bfo:FiatPoint", "bfo:FiatLine", "bfo:FiatSurface"'])
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(bfo:FiatPoint)
oClass(bfo:FiatLine)
oClass(bfo:FiatSurface)
allDisjoint(['"bfo:FiatPoint", "bfo:FiatLine", "bfo:FiatSurface"'])
@enduml
```

Intersection of classes

```

oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
intersection(['iof:PieceOfEquipment', "iof:Assembly", "iof:System"])

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
intersection(i, ["iof:PieceOfEquipment", "iof:Assembly", "iof:System"])
@enduml

```

```

oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
intersection(m1, ["iof:PieceOfEquipment", "iof:Assembly", "iof:System"])

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
intersection(m1, ["iof:PieceOfEquipment", "iof:Assembly", "iof:System"])
@enduml

```

Equivalent to the Intersection of classes

```

oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
equiIntersection(ns1:Machine, ["iof:PieceOfEquipment", "iof:Assembly",
"iof:System"])

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:PieceOfEquipment)

```

```

oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
equiIntersection(ns1:Machine, '["iof:PieceOfEquipment", "iof:Assembly",
"iof:System"]')
@enduml

```

subClass of the Intersection of classes

```

oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
subIntersection(ns1:Machine, '["iof:PieceOfEquipment", "iof:Assembly",
"iof:System"]')

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
subIntersection(ns1:Machine, '["iof:PieceOfEquipment", "iof:Assembly",
"iof:System"]')
@enduml

```

Intersection of classes as subClass

```

oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
superIntersection(ns1:Machine, '["iof:PieceOfEquipment", "iof:Assembly",
"iof:System"]')

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
superIntersection(ns1:Machine, '["iof:PieceOfEquipment", "iof:Assembly",

```



```
"iof:System"]', down, up)
@enduml
```

Union of classes

```
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
union(u, ["iof:PieceOfEquipment", "iof:Assembly", "iof:System"]')
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
union(u, ["iof:PieceOfEquipment", "iof:Assembly", "iof:System"]')
@enduml
```

```
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
union(m1, ["iof:PieceOfEquipment", "iof:Assembly", "iof:System"]')
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
union(m1, ["iof:PieceOfEquipment", "iof:Assembly", "iof:System"]')
@enduml
```

Equivalent to the Union of classes

```
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
equiUnion(ns1:Machine, ["iof:PieceOfEquipment", "iof:Assembly", "iof:System"]')
```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
equiUnion(ns1:Machine, '["iof:PieceOfEquipment", "iof:Assembly", "iof:System"]',
left, left)
@enduml

```

Alternative: reference by variable

```

class(c1, iof:PieceOfEquipment)
class(c2, iof:Assembly)
class(c3, iof:System)
class(m1, ns1:Machine)
union(m2, '["c1", "c2", "c3"]')
equivalent(m1, m2)

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(c1, iof:PieceOfEquipment)
class(c2, iof:Assembly)
class(c3, iof:System)
class(m1, ns1:Machine)
union(m2, '["c1", "c2", "c3"]')
equivalent(m1, m2)
@enduml

```

subClass of the Union of classes

```

oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
subUnion(ns1:Machine, '["iof:PieceOfEquipment", "iof:Assembly", "iof:System"]')

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:PieceOfEquipment)

```

```

oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
subUnion(ns1:Machine, '["iof:PieceOfEquipment", "iof:Assembly", "iof:System"]',
left, left)
@enduml

```

Union of classes as subClass

```

oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
superUnion(ns1:Machine, '["iof:PieceOfEquipment", "iof:Assembly", "iof:System"]')

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(iof:PieceOfEquipment)
oClass(iof:Assembly)
oClass(iof:System)
oClass(ns1:Machine)
superUnion(ns1:Machine, '["iof:PieceOfEquipment", "iof:Assembly", "iof:System"]',
down, up)
@enduml

```

OneOf restriction on class

```

oClass(ns:ProductionManager)
oIndividual(ns1:John)
oIndividual(ns1:Mary)
oIndividual(ns1:Avery)
oneOf(ns:ProductionManager, '["ns1:John", "ns1:Mary", "ns1:Avery"]')

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(ns:ProductionManager)
oIndividual(ns1:John)
oIndividual(ns1:Mary)
oIndividual(ns1:Avery)
oneOf(ns:ProductionManager, '["ns1:John", "ns1:Mary", "ns1:Avery"]')
@enduml

```

OneOf restriction on class

```
oClass(ns:ProductionManager)
oIndividual(ns1:John)
oIndividual(ns1:Mary)
oIndividual(ns1:Avery)
oneOf(ns:ProductionManager, '["ns1:John", "ns1:Mary", "ns1:Avery"]')
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(ns:ProductionManager)
oIndividual(ns1:John)
oIndividual(ns1:Mary)
oIndividual(ns1:Avery)
oneOf(ns:ProductionManager, '["ns1:John", "ns1:Mary", "ns1:Avery"]', up)
@enduml
```

Value restriction by class (quantified class)

```
class(eo, ns:ElectricVehicle)
class(bat, ns:Battery)
only(pb, ns:poweredBy, bat) # for universal quantifier
-/-
some(pb, ns:poweredBy, bat) # for existential quantifier
equivalent(eo, pb, right)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(eo, ns:ElectricVehicle)
class(bat, ns:Battery)
only(pb, ns:poweredBy, bat)
equivalent(eo, pb, right)
@enduml
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(eo, ns:ElectricVehicle)
class(bat, ns:Battery)
some(pb, ns:poweredBy, bat)
```

```
equivalent(eo, pb, right)
@enduml
```

Equivalent to a class and a quantified Class

```
class(v, ns:Vehicle)
class(ev, ns:ElectricVehicle)
class(bat, ns:Battery)
andSome(pb, v, ns:poweredBy, bat)
-/-
andOnly(pb, v, ns:poweredBy, bat)
equivalent(ev, pb)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(v, ns:Vehicle)
class(ev, ns:ElectricVehicle)
class(bat, ns:Battery)
andOnly(pb, v, ns:poweredBy, bat)
equivalent(ev, pb)
@enduml
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(v, ns:Vehicle)
class(ev, ns:ElectricVehicle)
class(bat, ns:Battery)
andSome(pb, v, ns:poweredBy, bat)
equivalent(ev, pb)
@enduml
```

Equivalent to a class or an quantified Class

```
class(v, ns:Vehicle)
class(ev, ns:ElectricVehicle)
class(bat, ns:Battery)
orSome(pb, v, ns:poweredBy, bat)
-/-
orOnly(pb, v, ns:poweredBy, bat)
equivalent(ev, pb)
```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(v, ns:Vehicle)
class(ev, ns:ElectricVehicle)
class(bat, ns:Battery)
orOnly(pb, v, ns:poweredBy, bat)
equivalent(ev, pb)
@enduml

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(v, ns:Vehicle)
class(ev, ns:ElectricVehicle)
class(bat, ns:Battery)
orSome(pb, v, ns:poweredBy, bat)
equivalent(ev, pb)
@enduml

```

Equivalent to a cardinally constrained Class

```

class(v, ns:Vehicle)
class(ev, ns:ElectricVehicle)
class(bat, ns:Battery)
someCard(pb, ns:poweredBy, bat, min/max/exactly, 2)
equivalent(ev, pb)

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(ev, ns:ElectricVehicle)
class(bat, ns:Battery)
someCard(pb, ns:poweredBy, bat, min, 1)
equivalent(ev, pb, right)
@enduml

```

```

@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(ev, ns:ElectricVehicle)
class(bat, ns:Battery)
someCard(pb, ns:poweredBy, bat, max, 4)

```

```
equivalent(ev, pb, right)
@enduml
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
class(ev, ns:ElectricVehicle)
class(bat, ns:Battery)
someCard(pb, ns:poweredBy, bat, exactly, 2)
equivalent(ev, pb, right)
@enduml
```

Shorter version

equivalent to a quantified Class

```
oClass(ns:ElectricVehicle)
oClass(ns:Battery)
xSome(ns:ElectricVehicle, ns:poweredBy, ns:Battery)
-/-
xOnly(ns:ElectricVehicle, ns:poweredBy, ns:Battery)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(ns:ElectricVehicle)
oClass(ns:Battery)
xSome(ns:ElectricVehicle, ns:poweredBy, ns:Battery)
@enduml
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(ns:ElectricVehicle)
oClass(ns:Battery)
xOnly(ns:ElectricVehicle, ns:poweredBy, ns:Battery)
@enduml
```

equivalent to a class and a quantified Class

```
oClass(ns:ElectricVehicle)
oClass(ns:Battery)
```

```
oClass(ns:Vehicle)
subClass(ns:ElectricVehicle, ns:Vehicle)
xSome(ns:ElectricVehicle, ns:poweredBy, ns:Battery)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(ns:ElectricVehicle)
oClass(ns:Battery)
oClass(ns:Vehicle)
subClass(ns:ElectricVehicle, ns:Vehicle)
xSome(ns:ElectricVehicle, ns:poweredBy, ns:Battery)
@enduml
```

equivalent to a class and multiple class expressions as clauses

```
oClass(ns:ElectricVehicle)
oClass(ns:Battery)
oClass(ns:Vehicle)
subClass(ns:ElectricVehicle, ns:Vehicle)
xSome(ns:ElectricVehicle, ns:poweredBy, ns:Battery)
```

```
@startuml
!include
https://raw.githubusercontent.com/iofoundry/ontopuml/refs/heads/main/iof.iuml
oClass(ns:ElectricVehicle)
oClass(ns:Battery)
oClass(ns:Vehicle)
oClass(ns:AutomaticDriver)
subClass(ns:ElectricVehicle, ns:Vehicle)
xSome(ns:ElectricVehicle, ns:poweredBy, ns:Battery)
xSome(ns:ElectricVehicle, ns:drivenBy, ns:AutomaticDriver)
@enduml
```