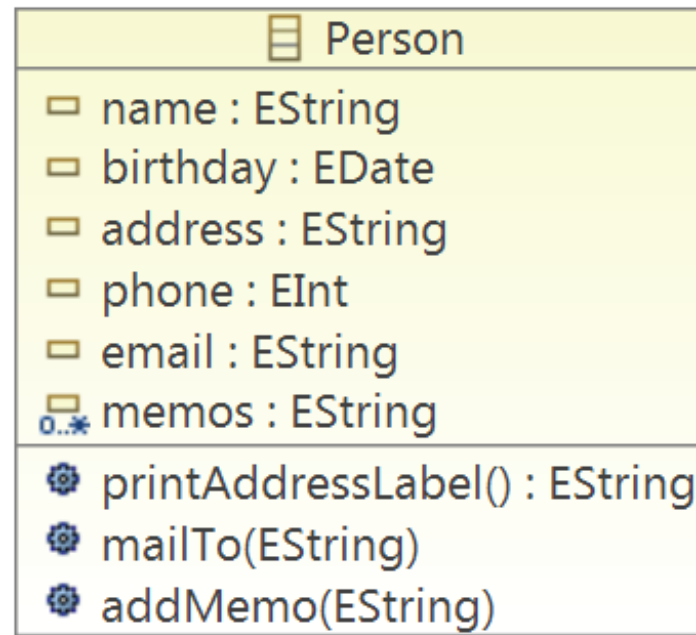


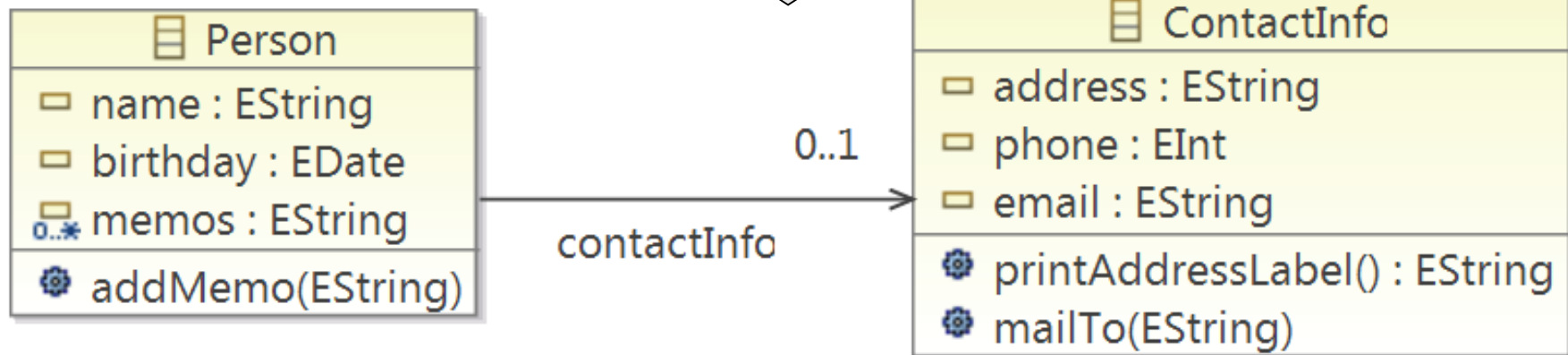
Parameter-Listen für Multi-Regeln

Manuel Ohrndorf
Praktische Informatik
Universität Siegen

Motivation

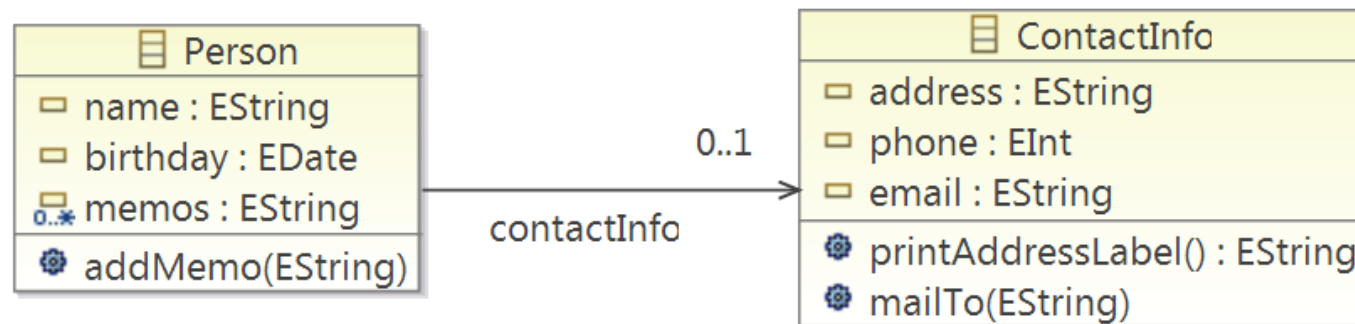
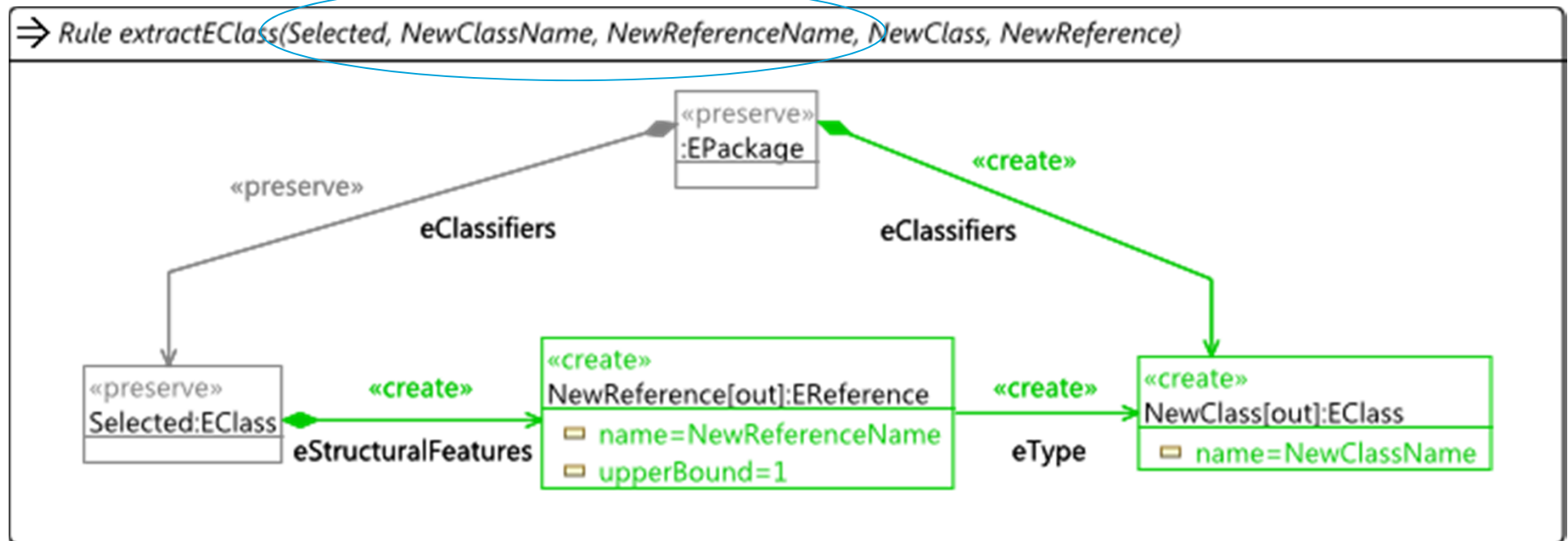


Extract Class



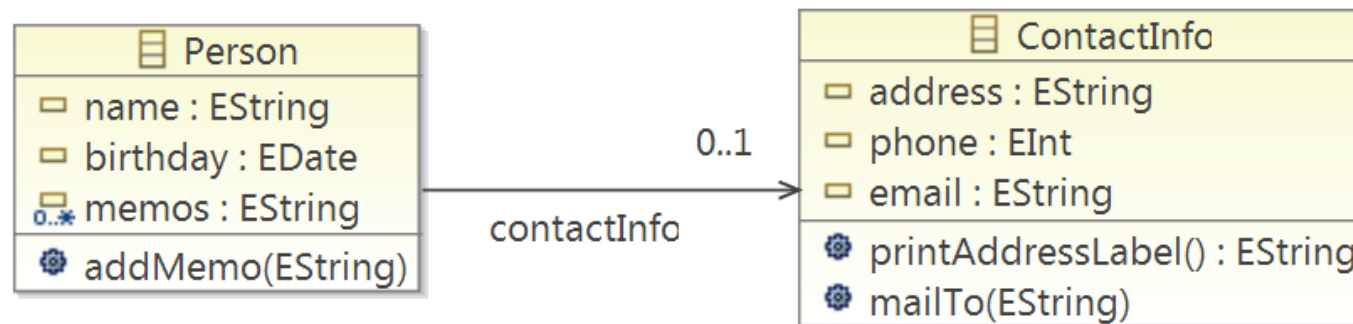
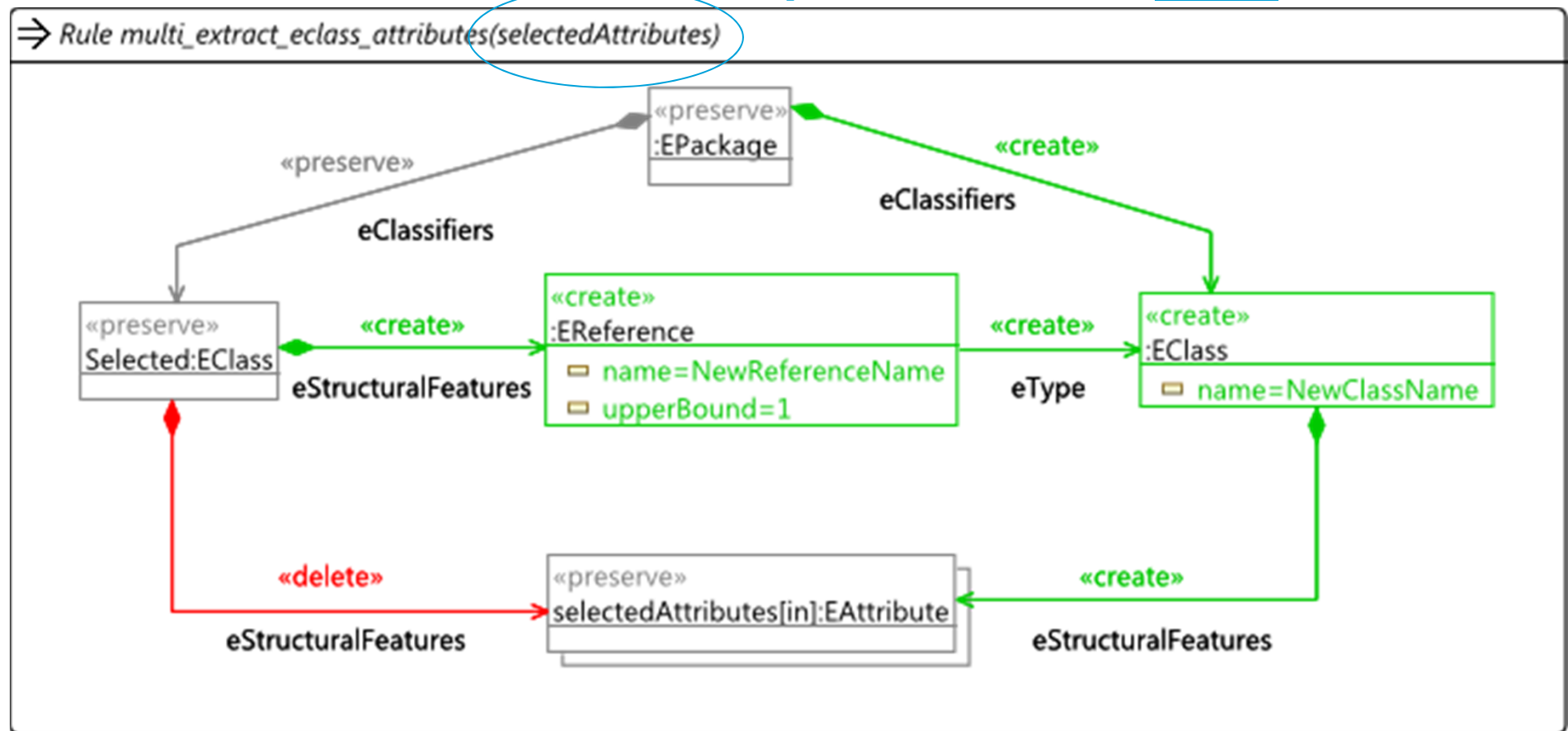
Henshin Transformation – Kernel-Regel

Input Parameter



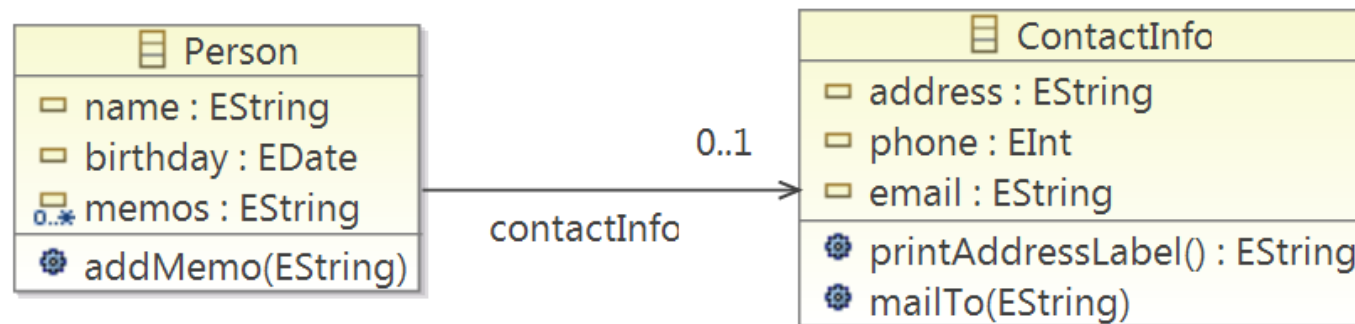
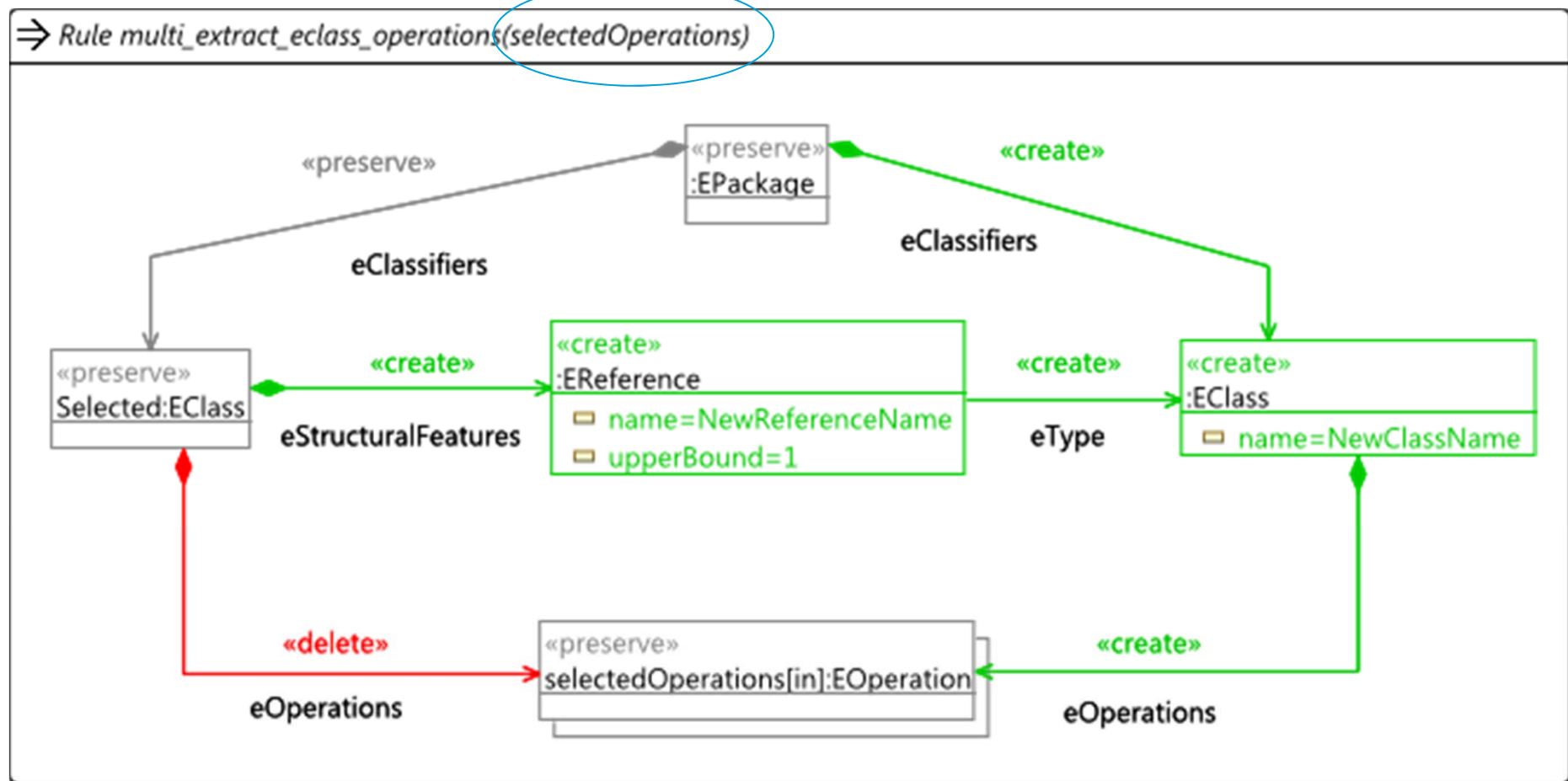
Henshin Transformation – Multi-Regel

Input Parameter-Liste

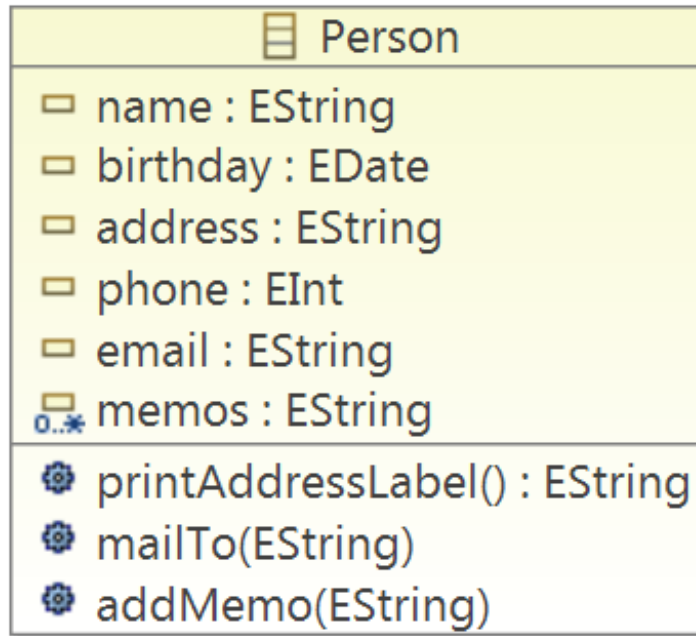


Henshin Transformation – Multi-Regel

Input Parameter-Liste



Ausführen der Regeln



Exract Attribute

| | Selected Attributes |
|---------|---------------------|
| Match 1 | address |
| Match 2 | phone |
| Match 3 | email |

Exract Operation

| | Selected Operations |
|---------|---------------------|
| Match 4 | printAddressLabel() |
| Match 5 | mailto() |

Semantik der Regeln

- `amalgamation-unit(unit-para_1, ... , unit-para_n)`
 - `kernel-regel(rule-para_1, ... , rule-para_i)`
 - `multi-regel_1(multi-para_1.1, ... , multi-para_1.j)`
 - ...
 - `multi-regel_m(multi-para_m.1, ... , multi-para_m.k)`
 - `mapping<unit-para -> rule-para | multi-para>`
- `rule-para` \in {value, object}
- `multi-para` \in {value, object, **list**}

Grundlegende Semantik bleibt erhalten:

Eine Amalgamation-Unit ist anwendbar, wenn die Kern-Regel anwendbar ist.

Multi-Regel 1

| | parameter-list_1.1 |
|---------|--------------------|
| Match 1 | value/object |
| Match 2 | value/object |
| Match 3 | value/object |

Multi-Regel 2

| | parameter-list_2.1 |
|---------|--------------------|
| Match 4 | value/object |
| Match 5 | value/object |

Semantik der Regeln

- `amalgamation-unit(unit-para_1, ... , unit-para_n)`
 - `kernel-regel(rule-para_1, ... , rule-para_i)`
 - `multi-regel_1(multi-para_1.1, ... , multi-para_1.j)`
 - ...
 - `multi-regel_m(multi-para_m.1, ... , multi-para_m.k)`
 - `mapping<unit-para -> rule-para | multi-para>`
- `rule-para` \in {value, object}
- `multi-para` \in {value, object, **list**}

Kein Match für einen einzelnen Parameterwert.

Multi-Regel 1

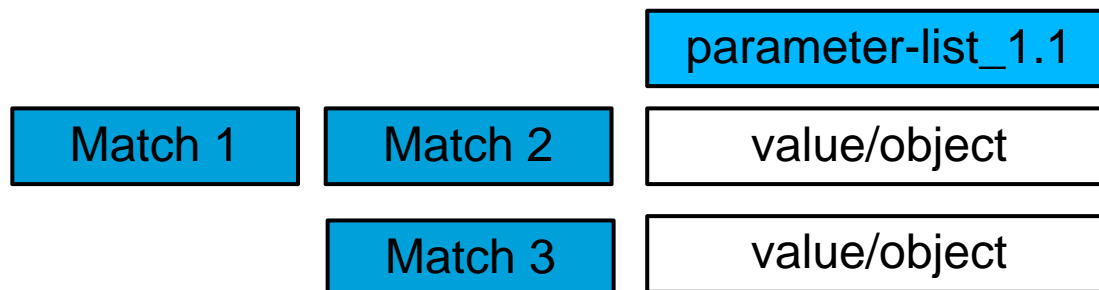
| | parameter-list_1.1 |
|---------|--------------------|
| Match 1 | value/object |
| | value/object |
| Match 2 | value/object |

Semantik der Regeln

- `amalgamation-unit(unit-para_1, ... , unit-para_n)`
 - `kernel-regel(rule-para_1, ... , rule-para_i)`
 - `multi-regel_1(multi-para_1.1, ... , multi-para_1.j)`
 - ...
 - `multi-regel_m(multi-para_m.1, ... , multi-para_m.k)`
 - `mapping<unit-para -> rule-para | multi-para>`
- `rule-para` \in {value, object}
- `multi-para` \in {value, object, **list**}

Mehrere Matches für einen einzelnen Parameterwert.

Multi-Regel 1



Semantik der Regeln

- `amalgamation-unit(unit-para_1, ... , unit-para_n)`
 - `kernel-regel(rule-para_1, ... , rule-para_i)`
 - `multi-regel_1(multi-para_1.1, ... , multi-para_1.j)`
 - ...
 - `multi-regel_m(multi-para_m.1, ... , multi-para_m.k)`
 - `mapping<unit-para -> rule-para | multi-para>`
- `rule-para` \in {value, object}
- `multi-para` \in {value, object, **list**}

Fehlende Parameterwerte.

Multi-Regel 1

| | parameter-list_1.1 | parameter-list_1.2 | parameter-list_1.3 |
|---------|--------------------|--------------------|--------------------|
| Match 1 | value object | value object | x |
| Match 2 | value object | value object | x |
| Match 3 | value object | x | x |

Semantik der Regeln

- `amalgamation-unit(unit-para_1, ... , unit-para_n)`
 - `kernel-regel(rule-para_1, ... , rule-para_i)`
 - `multi-regel_1(multi-para_1.1, ... , multi-para_1.j)`
 - ...
 - `multi-regel_m(multi-para_m.1, ... , multi-para_m.k)`
 - `mapping<unit-para -> rule-para | multi-para>`
- `rule-para` \in {value, object}
- `multi-para` \in {value, object, **list**}

Normales Amalgamation-Unit Verhalten.

Multi-Regel 1

| | parameter-list_1.1 | parameter-list_1.2 | parameter-list_1.3 |
|---------|--------------------|--------------------|--------------------|
| Match 1 | x | x | x |
| ... | x | x | x |
| Match N | x | x | x |

Semantik der Regeln

- `amalgamation-unit(unit-para_1, ... , unit-para_n)`
 - `kernel-regel(rule-para_1, ... , rule-para_i)`
 - `multi-regel_1(multi-para_1.1, ... , multi-para_1.j)`
 - ...
 - `multi-regel_m(multi-para_m.1, ... , multi-para_m.k)`
 - `mapping<unit-para -> rule-para | multi-para>`
- `rule-para` \in {value, object}
- `multi-para` \in {value, object, **list**}

Normales Amalgamation-Unit Verhalten & Parameter-Listen.

Multi-Regel 1

| | parameter-list_1.1 |
|---------|--------------------|
| Match 1 | x |
| ... | x |
| Match N | x |

Multi-Regel 2

| | parameter-list_2.1 |
|---------|--------------------|
| Match 1 | value object |
| Match 2 | value object |
| Match 3 | value object |

Hands on...

...Live Demo