

Qwalkeko user study



Reinout Stevens, Tim Molderez & Coen De Roover

Qwalkeko

Qwalkeko

- A tool to query git repositories with Java projects

Qwalkeko

- A tool to query git repositories with Java projects
- For software developers:
 - Who introduced this class?
 - Who made these changes?
 - When was this code last changed?
 - How often is this code changed?



Qwalkeko

- A tool to query git repositories with Java projects
- For software developers:
 - Who introduced this class?
 - Who made these changes?
 - When was this code last changed?
 - How often is this code changed?
- For researchers in the area of Mining Software Repositories:
 - Make predictions when to refactor, to migrate ...



Qwalkeko

- A tool to query git repositories with Java projects
- For software developers:

- Who introduced this class?
- Who made these changes?
- When was this code last changed?
- How often is this code changed?



- For researchers in the area of Mining Software Repositories:
 - Make predictions when to refactor, to migrate ...

Qwalkeko



Qwalkeko



- A query looks for occurrences of a "change pattern"
 - e.g. refactorings, api changes, library migration, ...

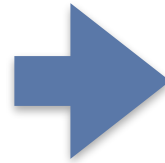
Qwalkeko



- A query looks for occurrences of a "change pattern"
 - e.g. refactorings, api changes, library migration, ...
- Qwalkeko is a **declarative** query language
 - Describe a change pattern in terms of **what** the before/after should look like, at the AST-level
 - Qwalkeko returns an executable, minimal script that describes **how** that change was made

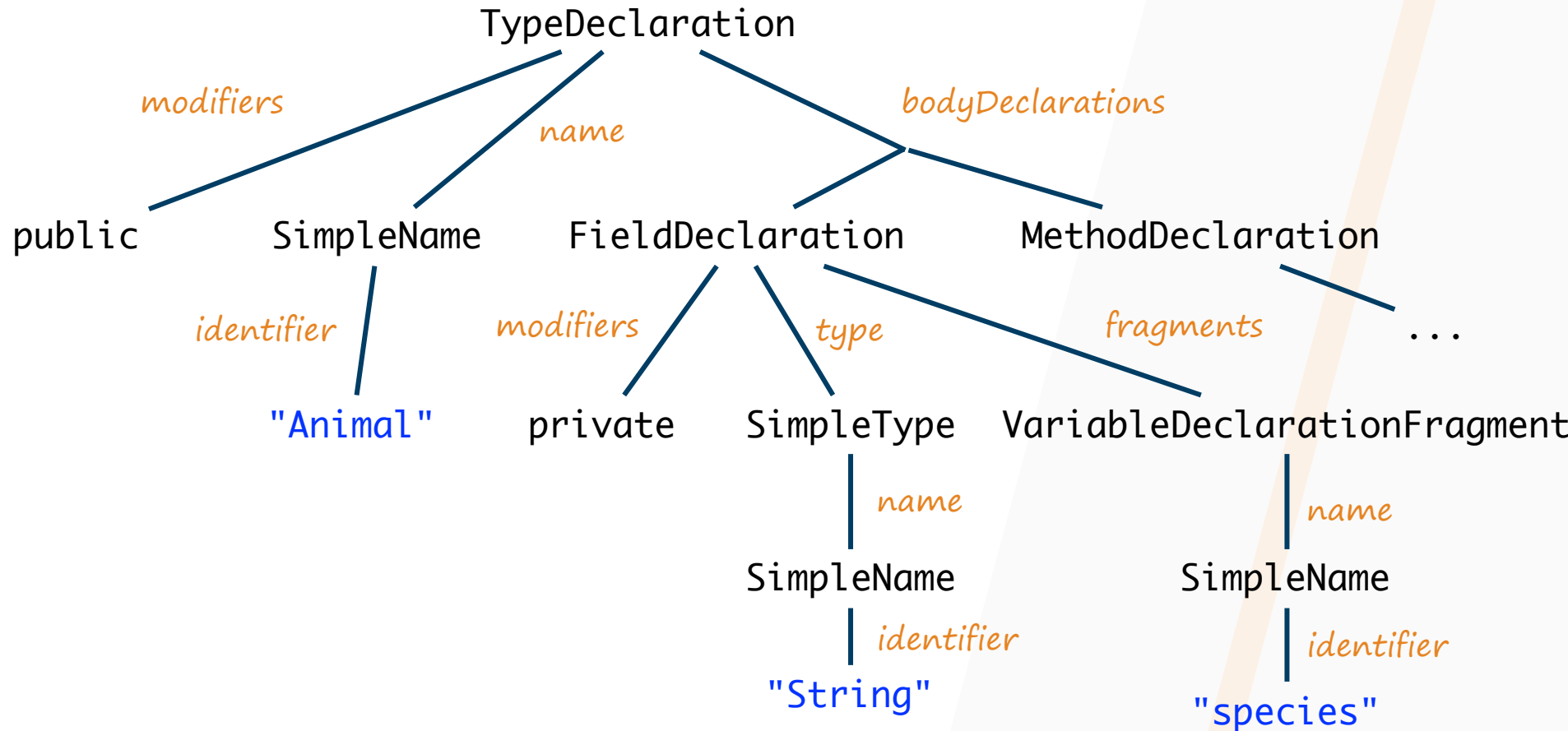
Field rename example

```
public class Animal {  
    private String species;  
  
    public Animal(String species) {  
        this.species = species;  
    }  
}
```

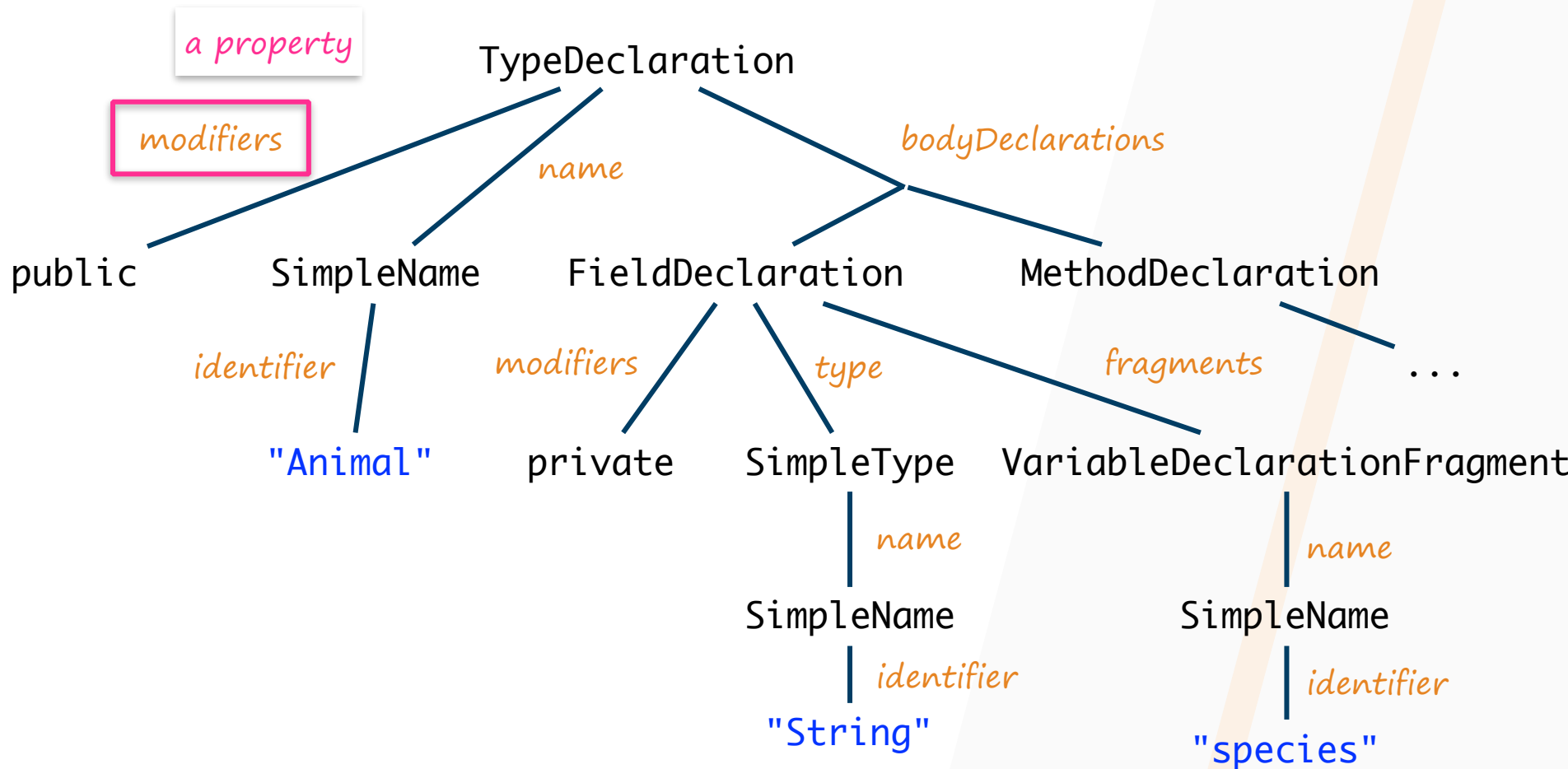


```
public class Animal {  
    private String breed;  
  
    public Animal(String breed) {  
        this.breed = breed;  
    }  
  
    public String toString() {  
        return "Breed:" + this.breed;  
    }  
}
```

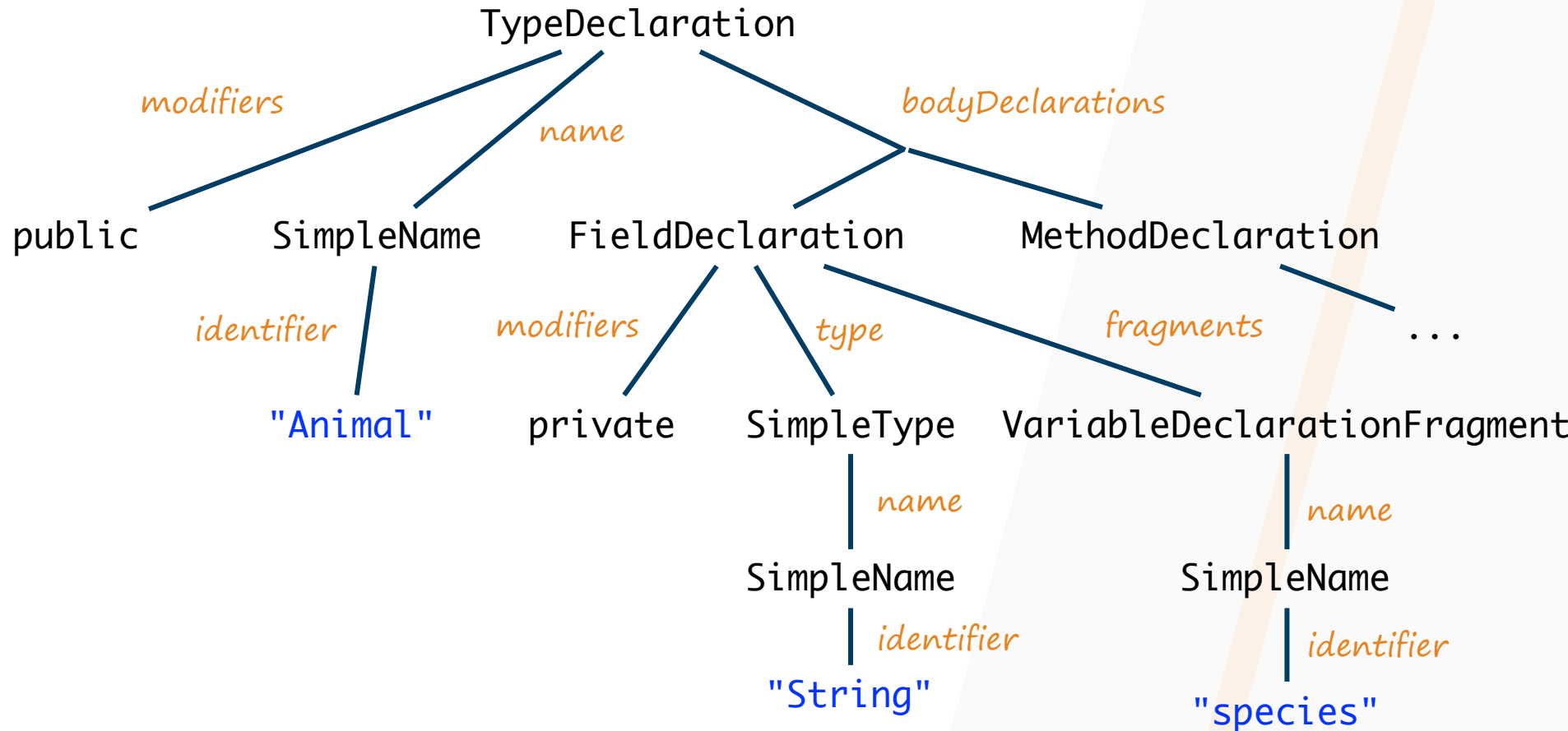
Field rename example - ASTs



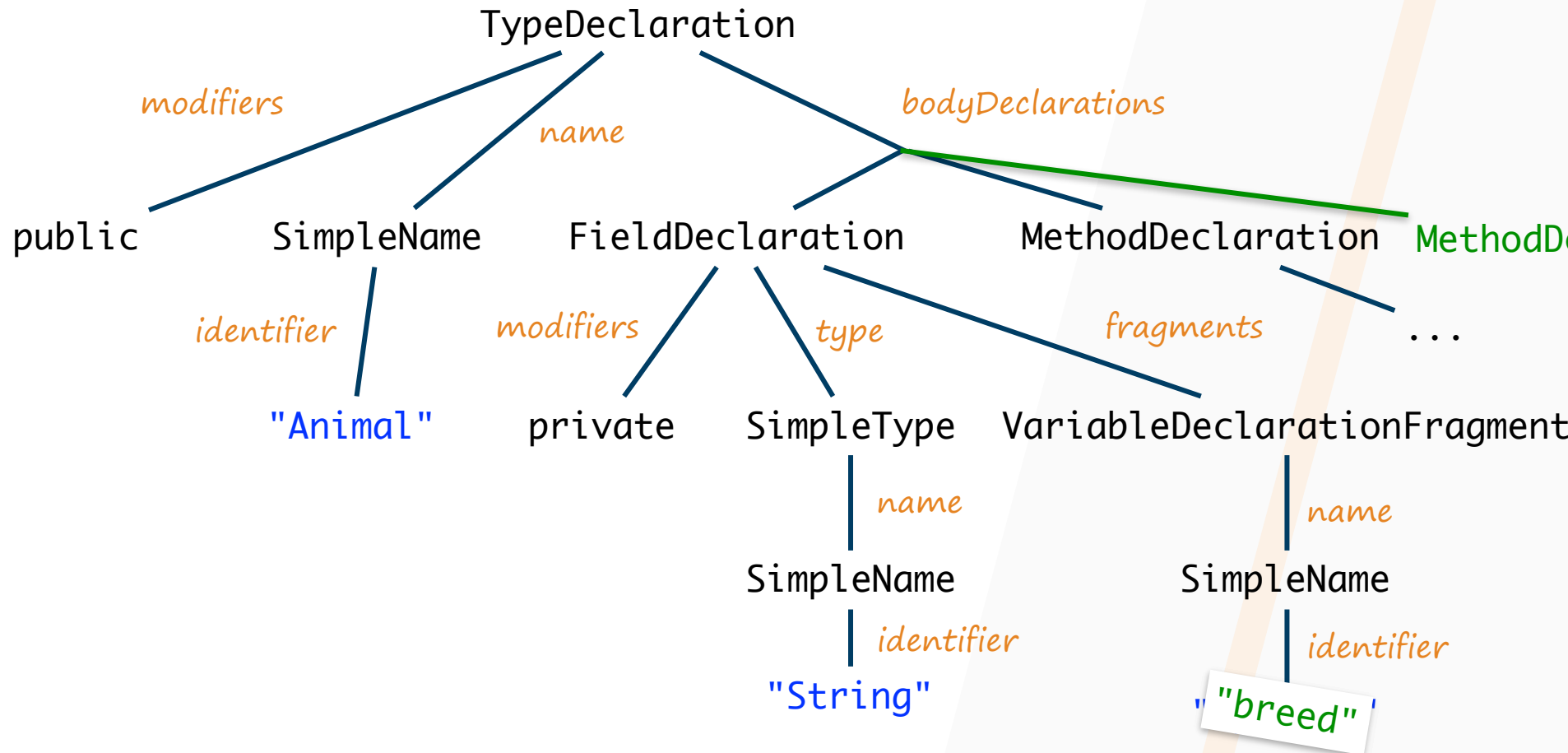
Field rename example - ASTs



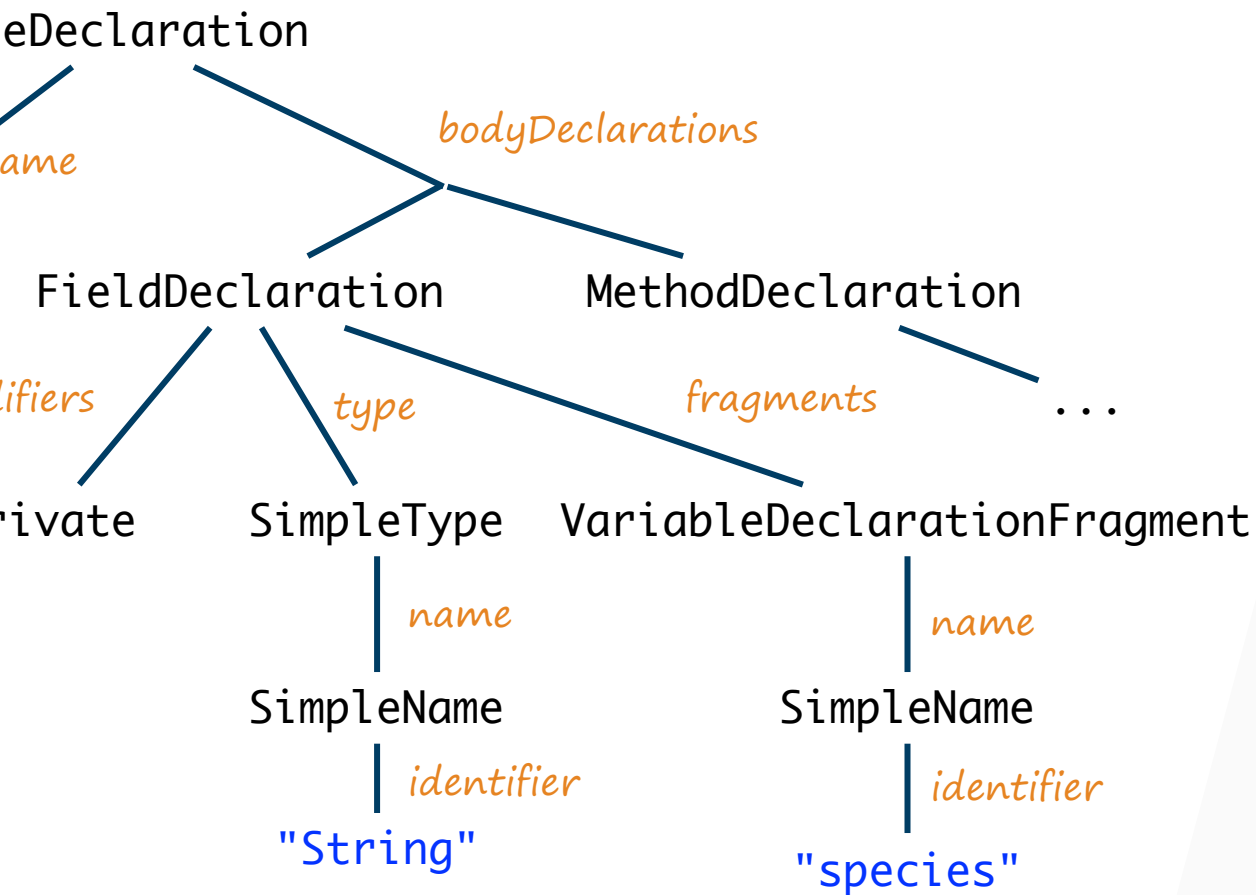
Field rename example - ASTs



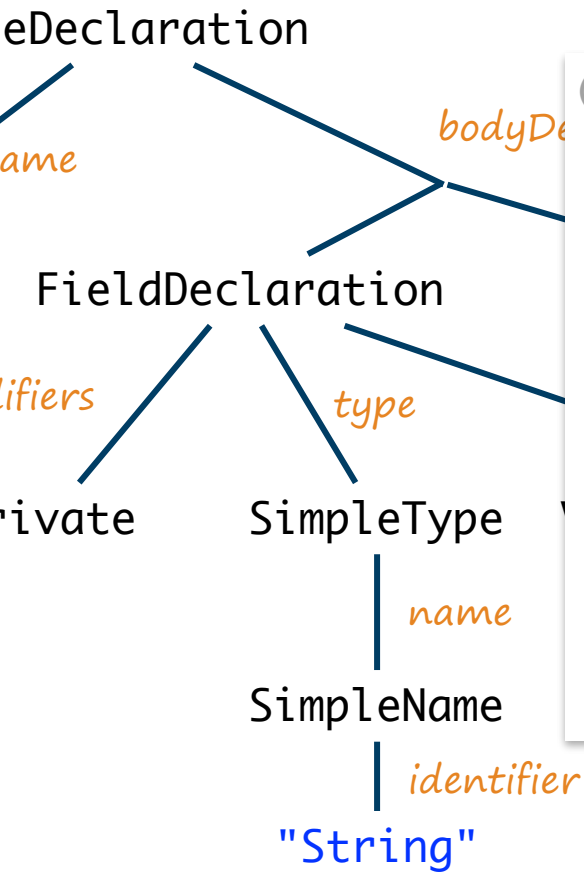
Field rename example - ASTs



Field rename example - query



Field rename example - query



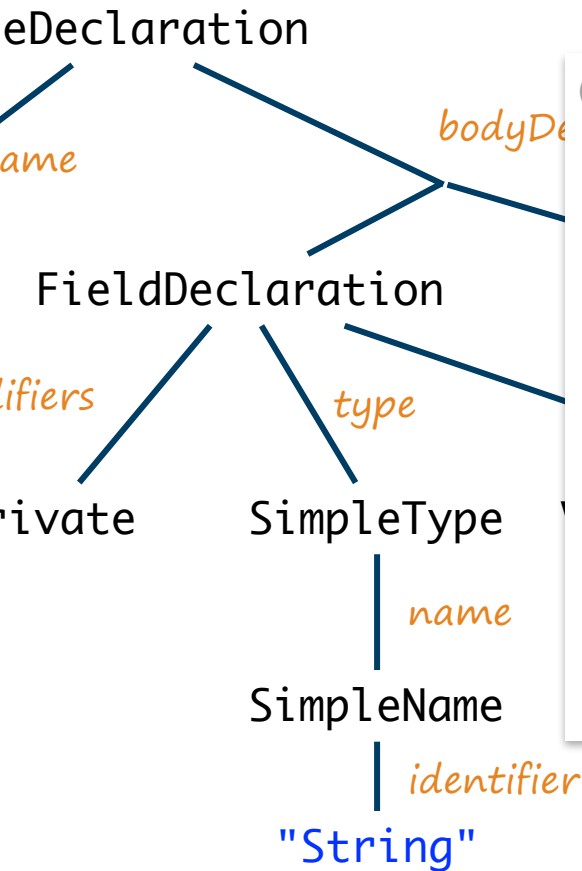
```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

"species"

Field rename example - query

These represent the Compilation Unit ASTs before / after the desired change



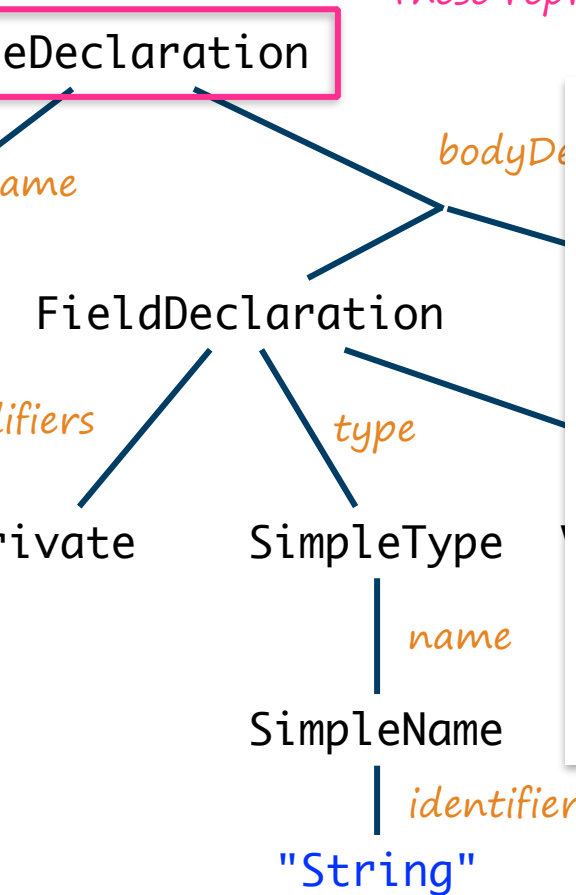
```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

identifier
"species"

Field rename example - query

before



These represent the Compilation Unit ASTs before / after the desired change

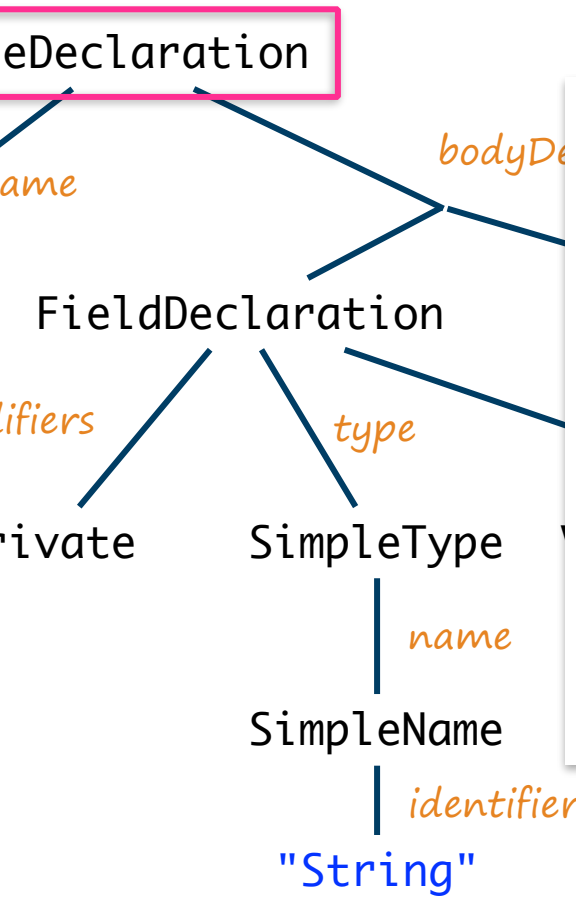
```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

"species"

Field rename example - query

before



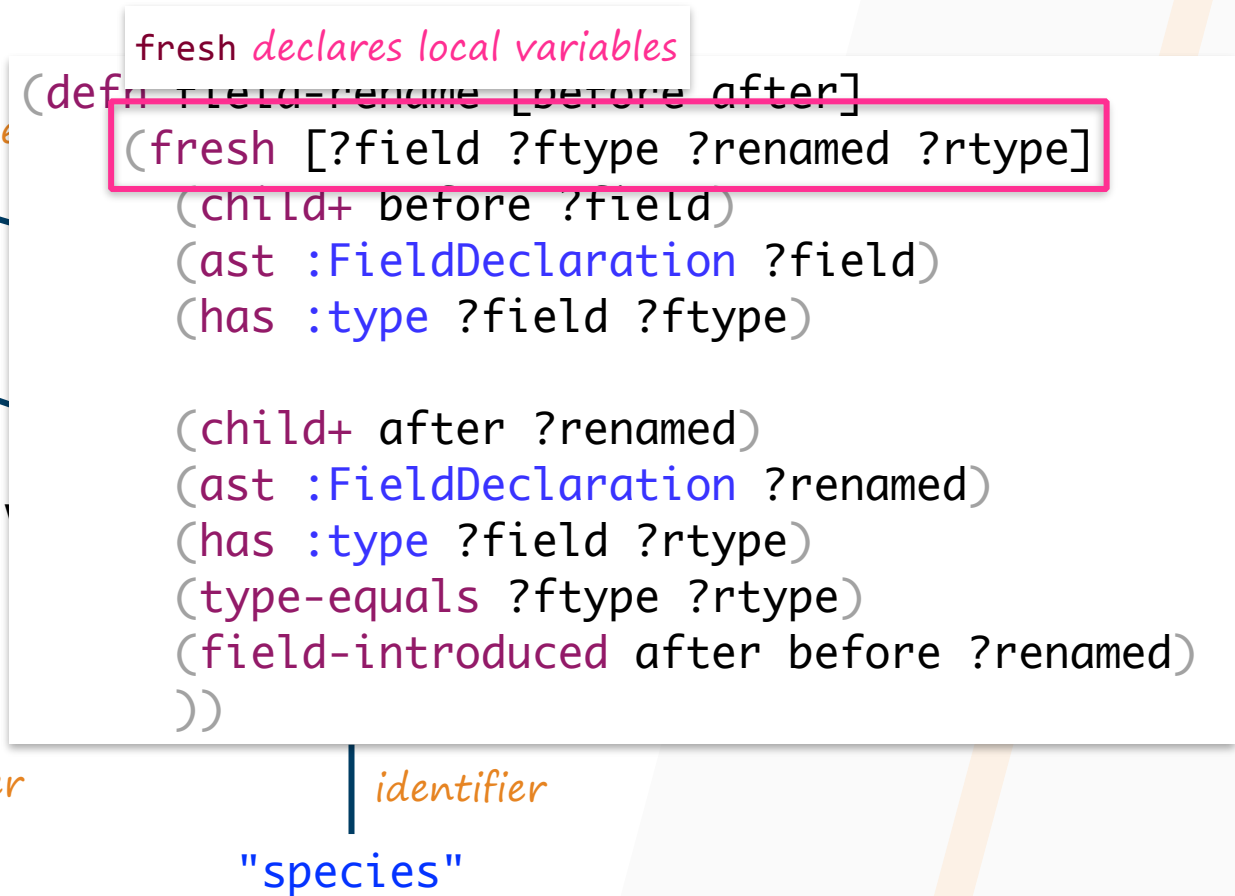
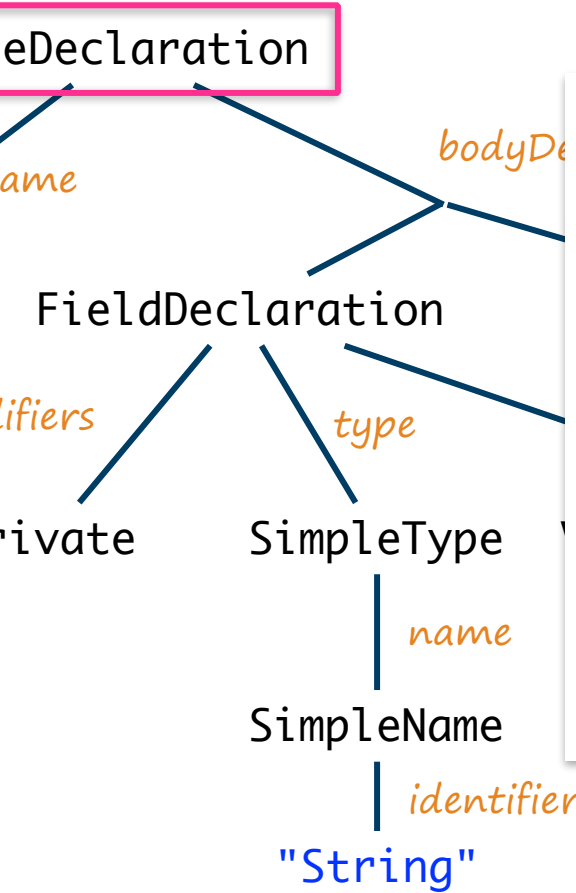
```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

`identifier`
`"species"`

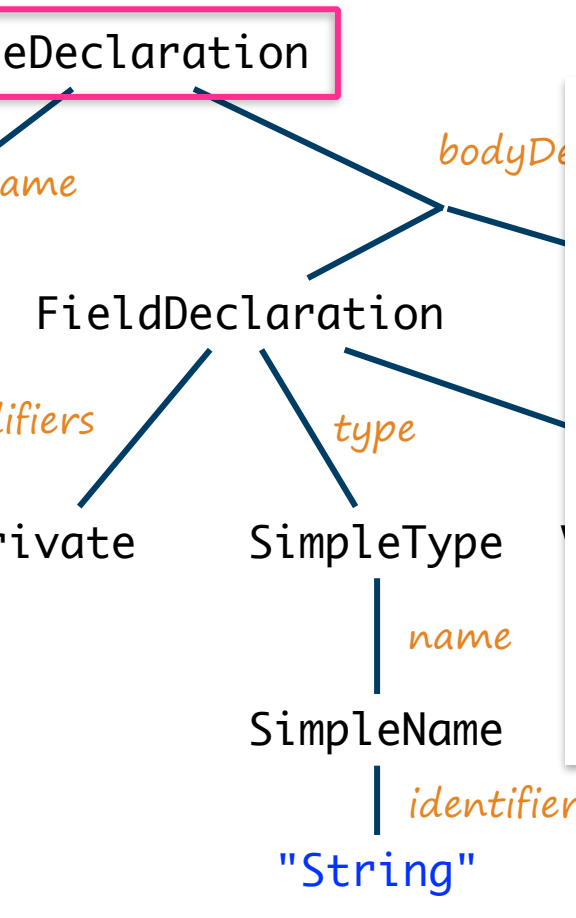
Field rename example - query

before



Field rename example - query

before

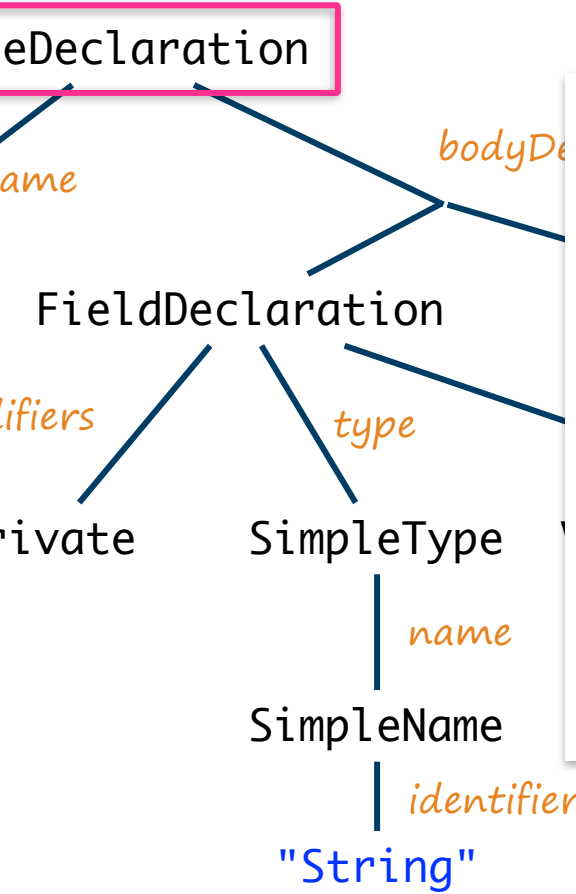


```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

Field rename example - query

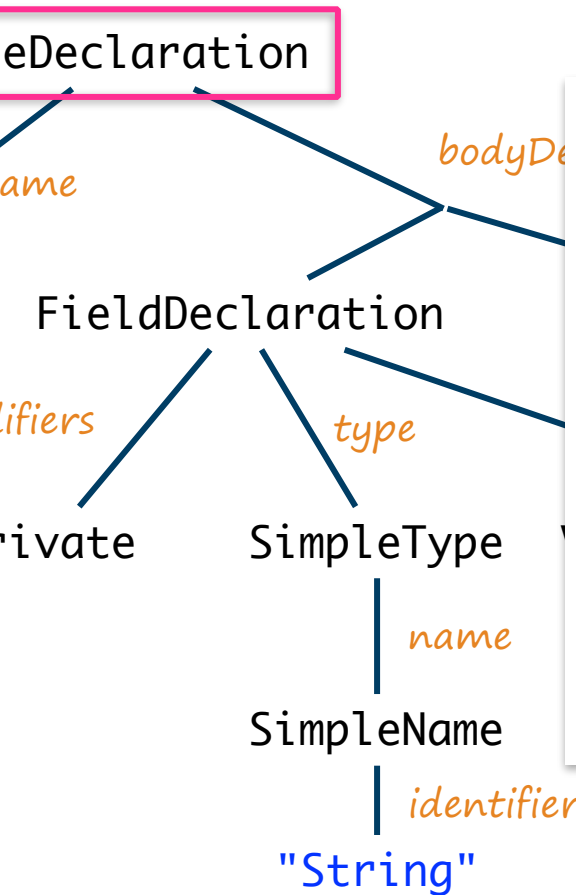
before



```
(defn field-rename [before ?field must-be-a-descendant-of before  
(fresh [?field ?renamed ?ftype] [?field ?renamed :ftype])  
(child+ before ?field)  
(ast :FieldDeclaration ?field)  
(has :type ?field ?ftype)  
  
(child+ after ?renamed)  
(ast :FieldDeclaration ?renamed)  
(has :type ?field ?rtype)  
(type-equals ?ftype ?rtype)  
(field-introduced after before ?renamed)  
)
```

Field rename example - query

before



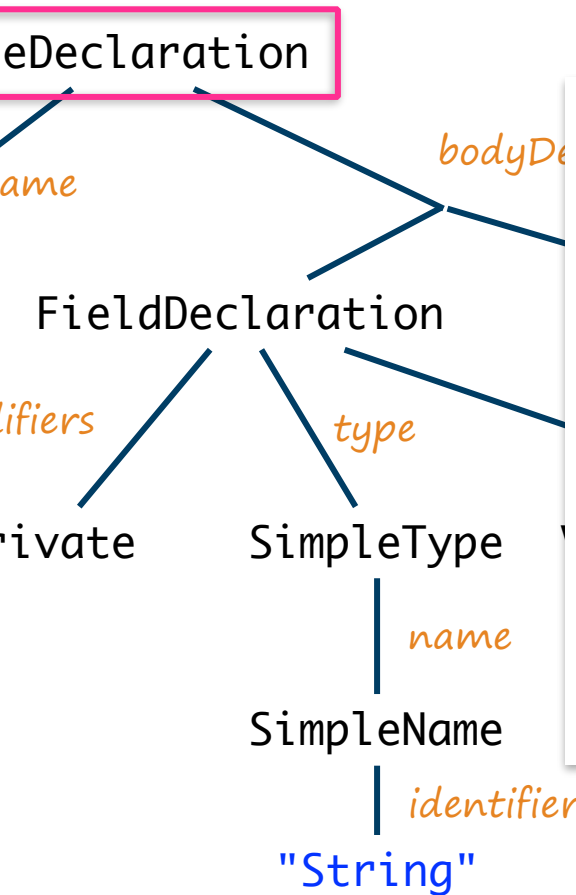
```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

identifier
"species"

Field rename example - query

before

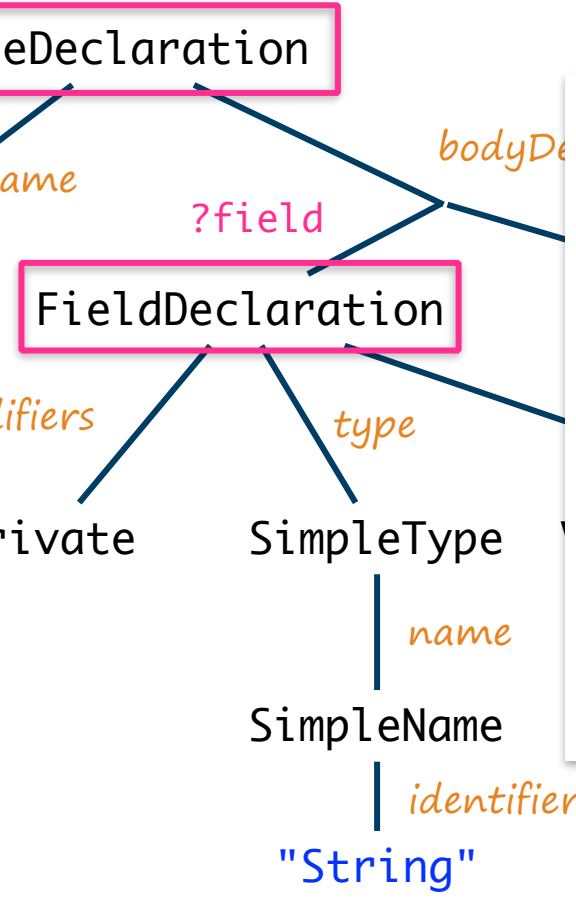


```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

Field rename example - query

before



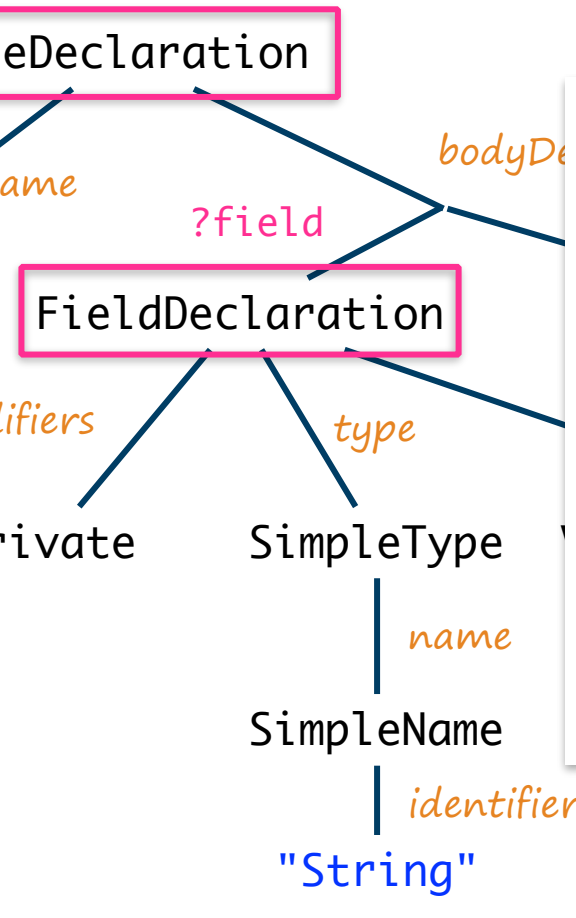
```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

"species"

Field rename example - query

before

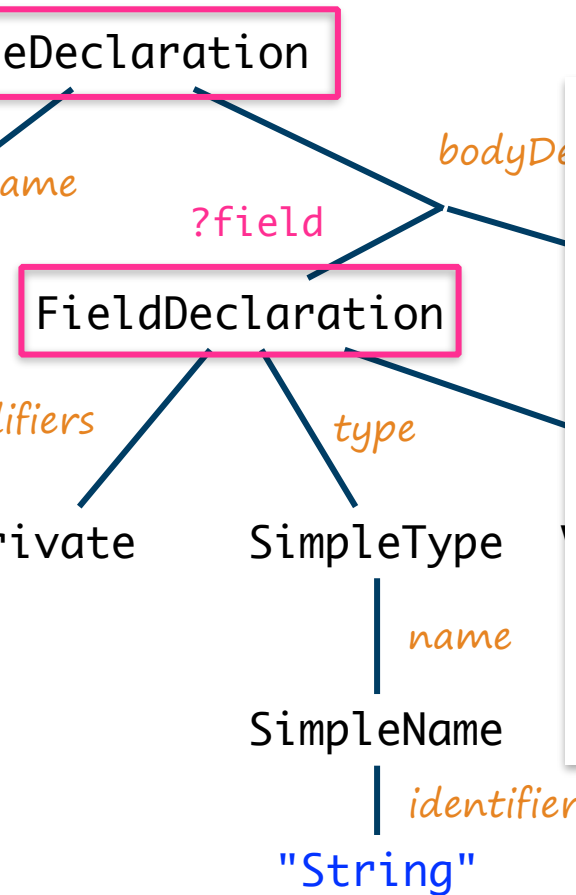


```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

Field rename example - query

before



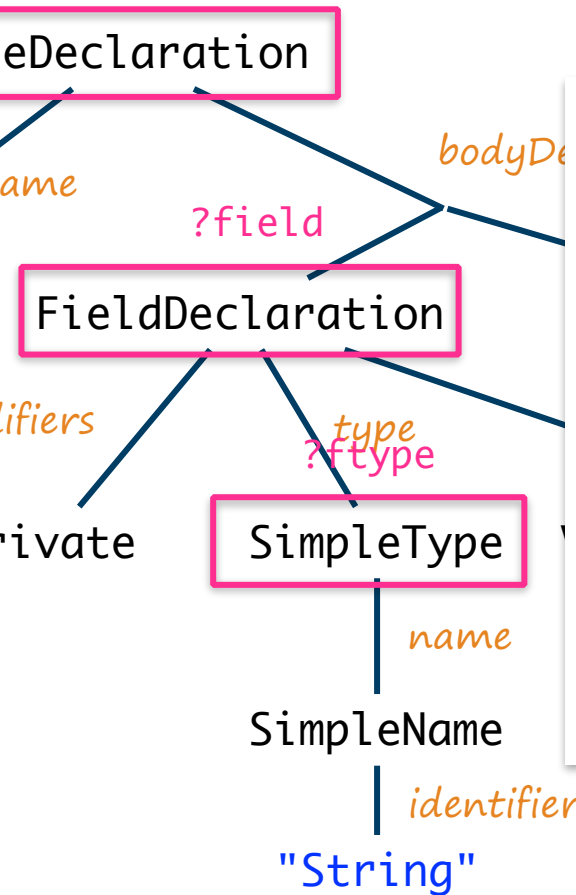
```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field ?field has a :type child, ?ftype)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

"species"

Field rename example - query

before

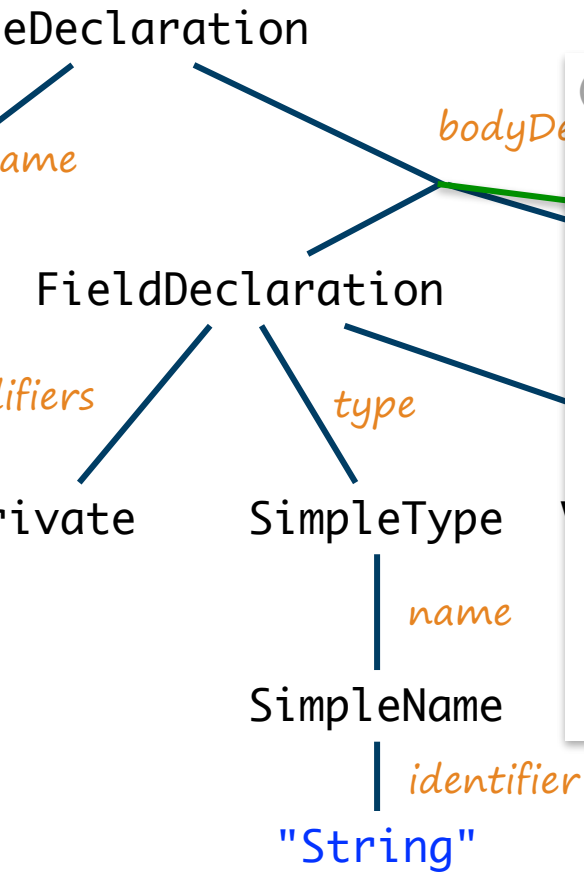


```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field ?field has a :type child, ?ftype)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?field ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

identifier
"species"

Field rename example - query



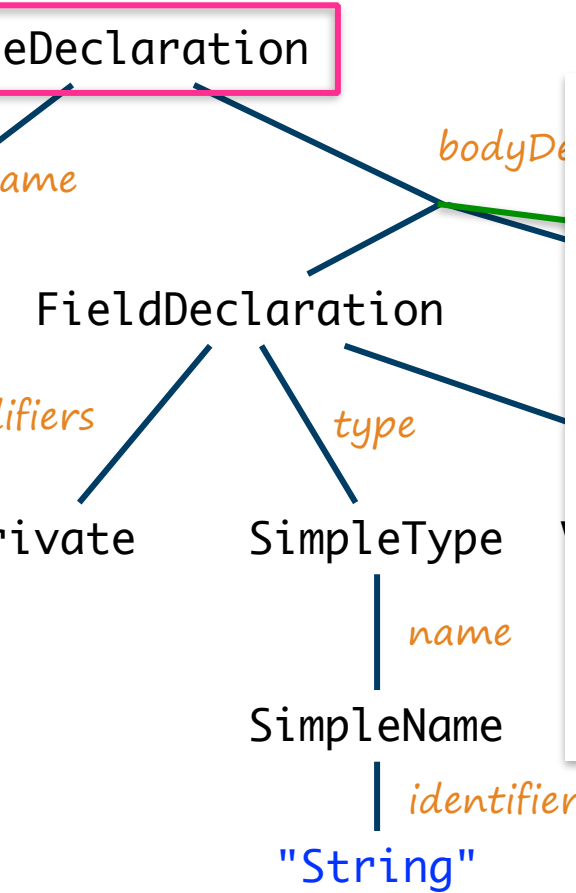
```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?renamed ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

`"breed"`

Field rename example - query

after



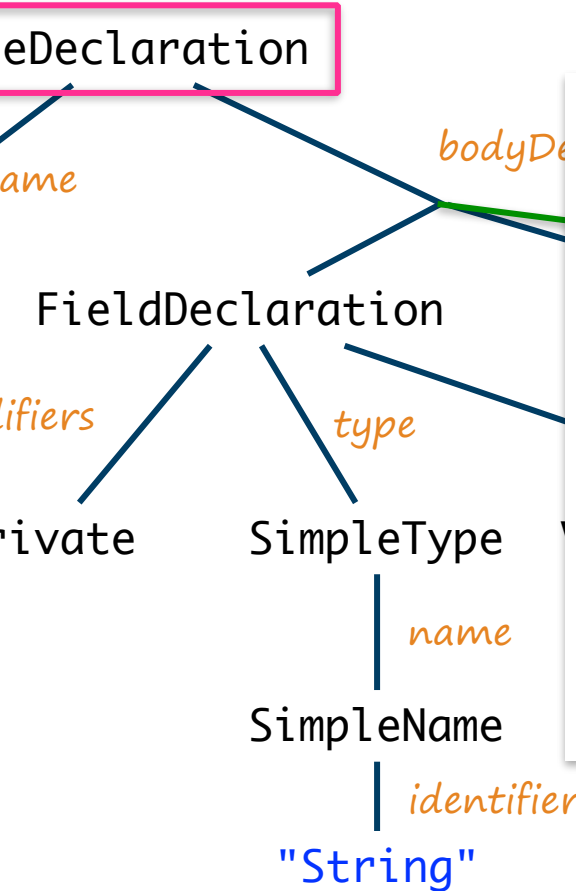
```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?renamed ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

identifier
"breed"

Field rename example - query

after

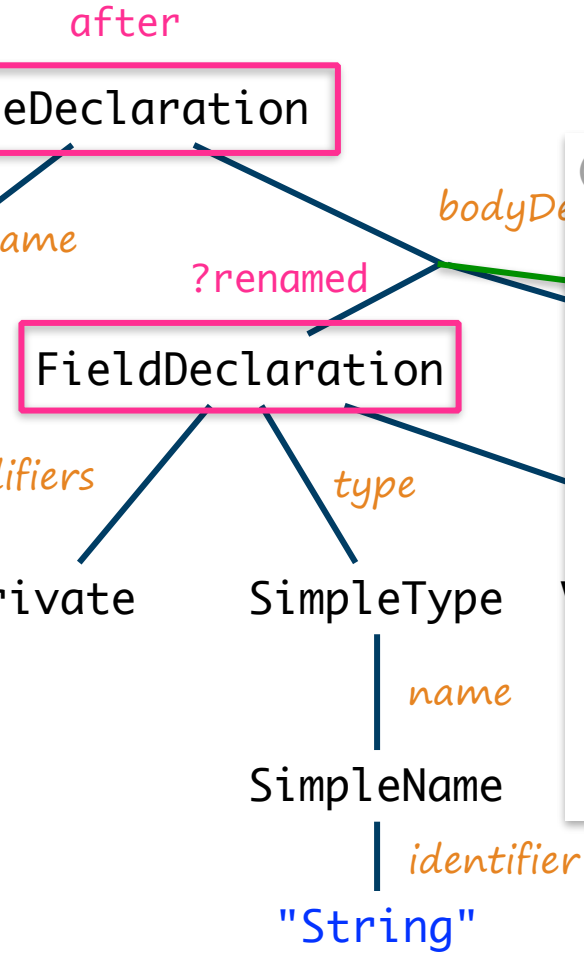


```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?renamed ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

identifier
"breed"

Field rename example - query

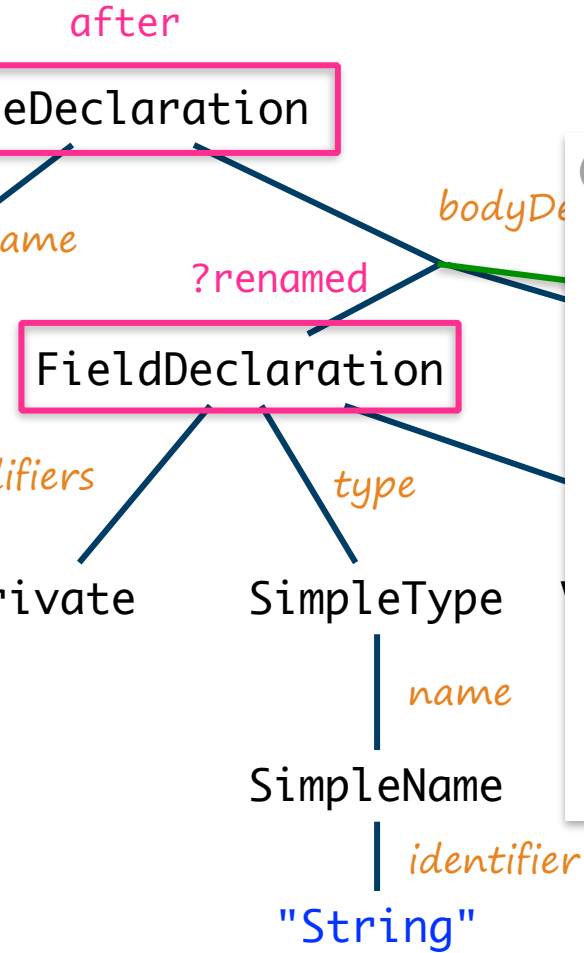


```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?renamed ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

"breed"

Field rename example - query

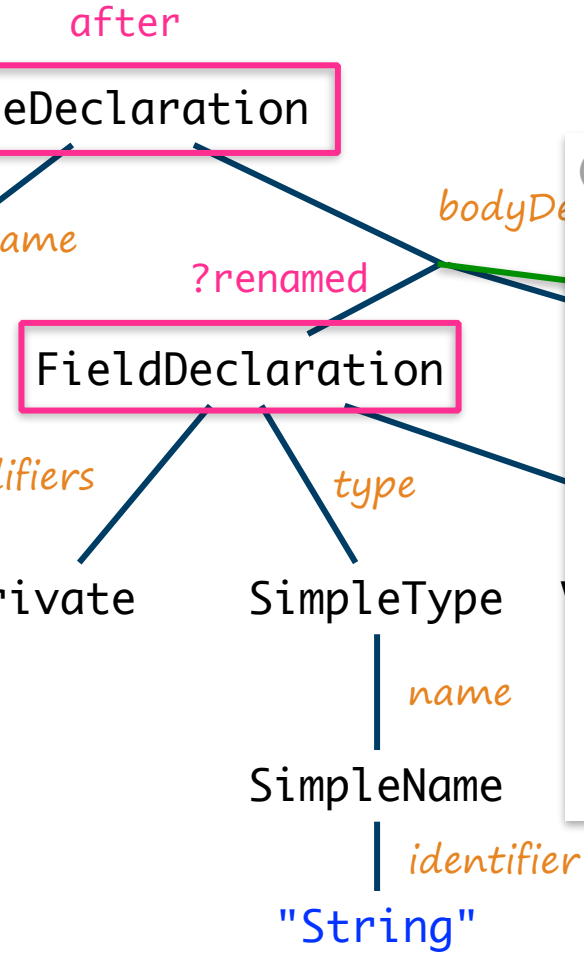


```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?renamed ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

identifier
"breed"

Field rename example - query

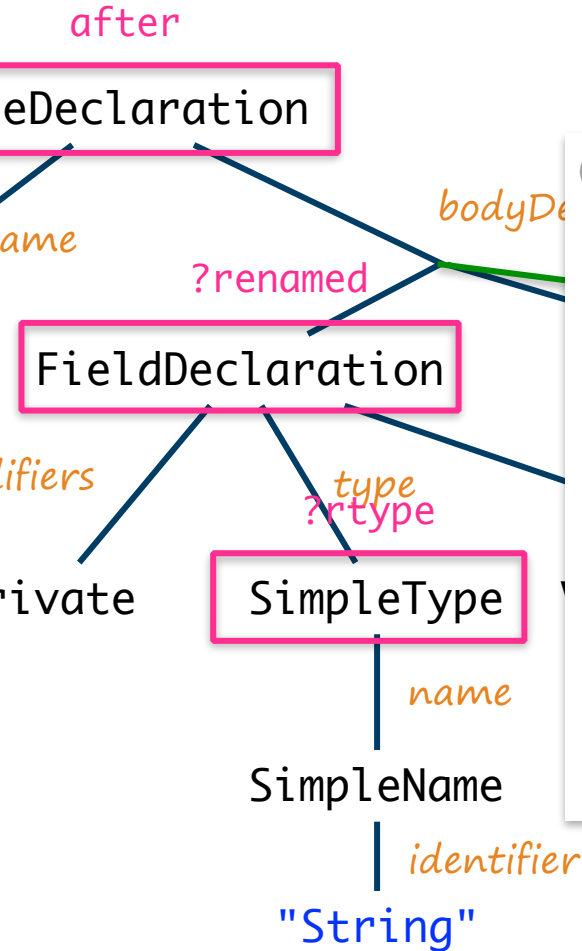


```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?renamed ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

"breed"

Field rename example - query

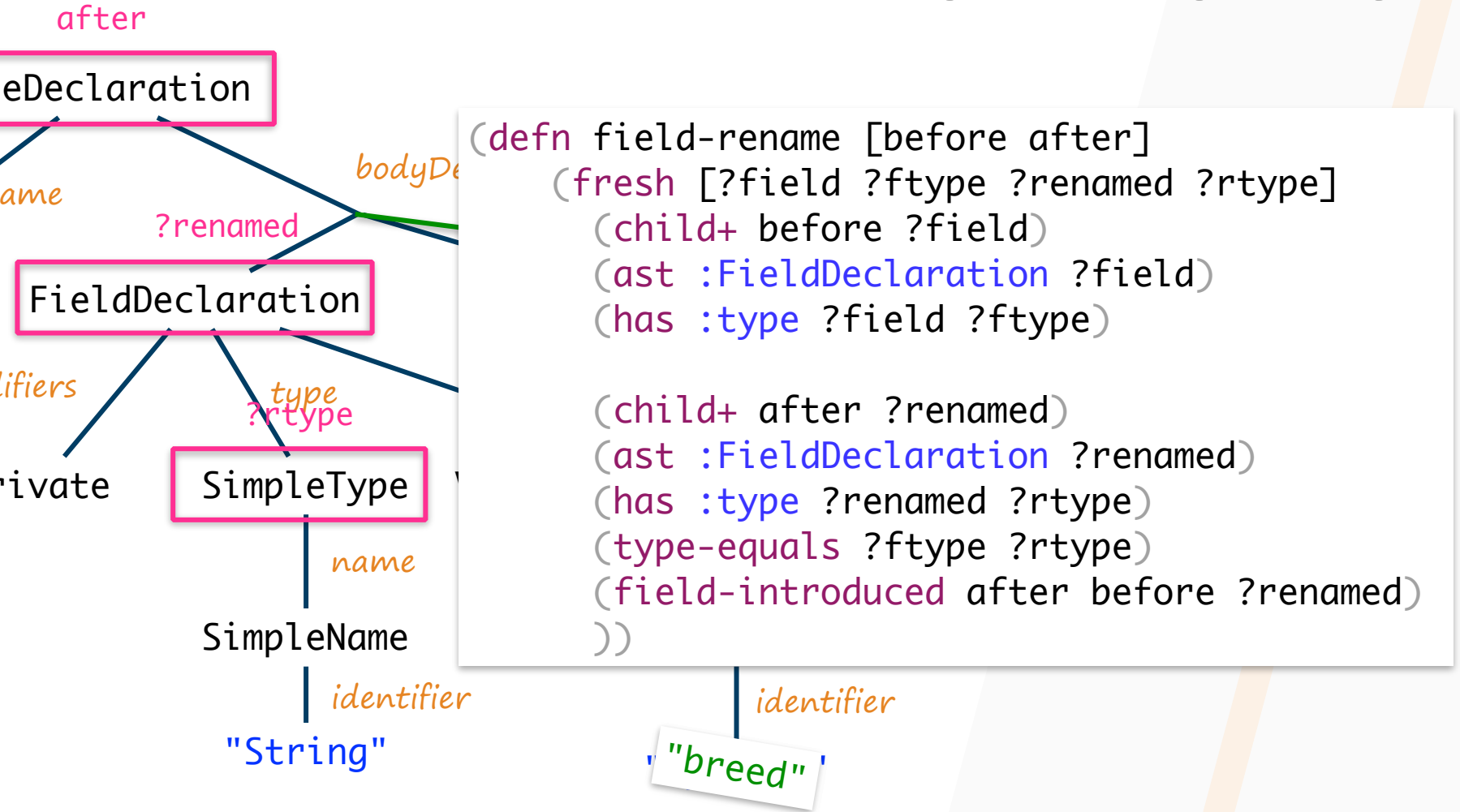


```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

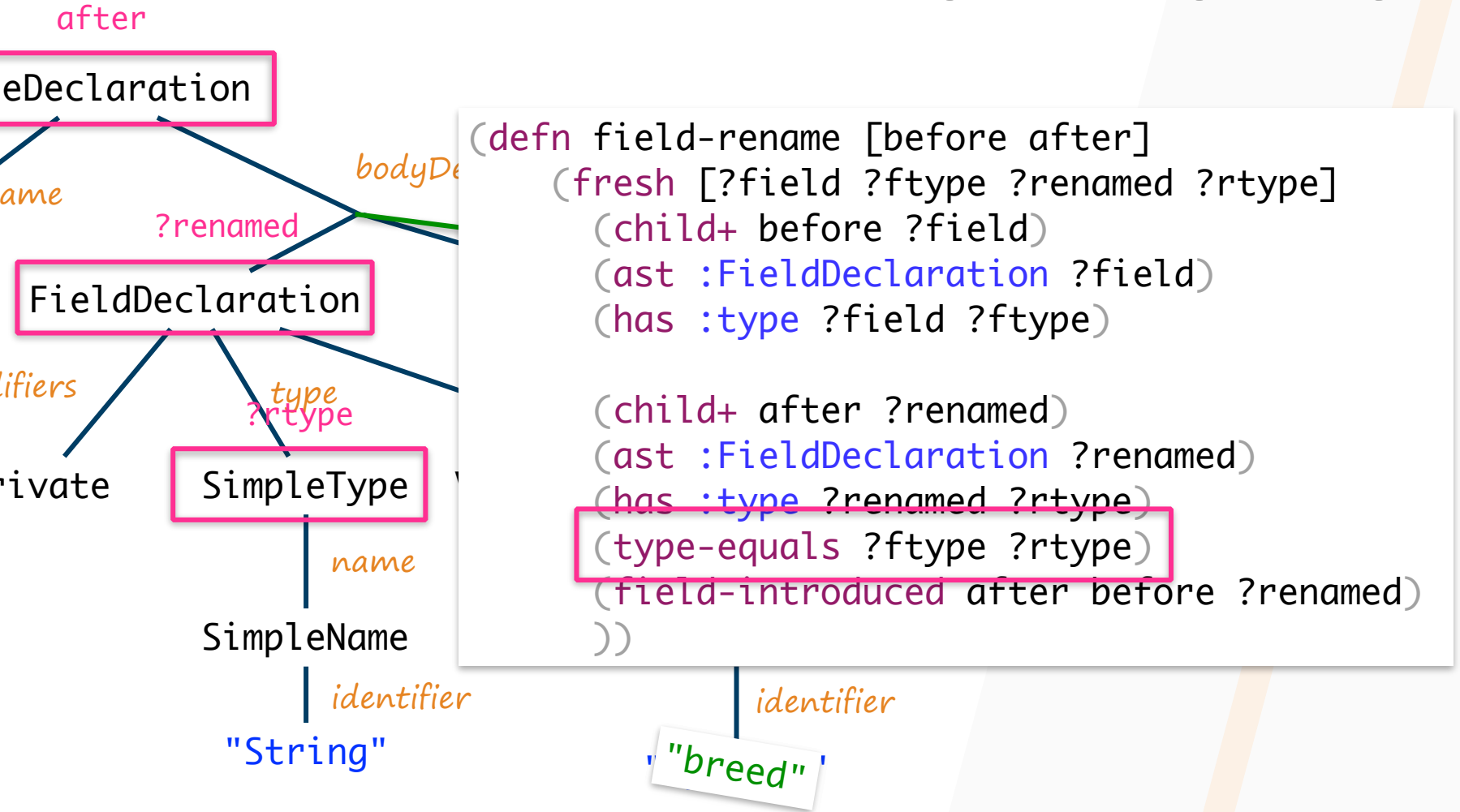
    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?renamed ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

"breed"

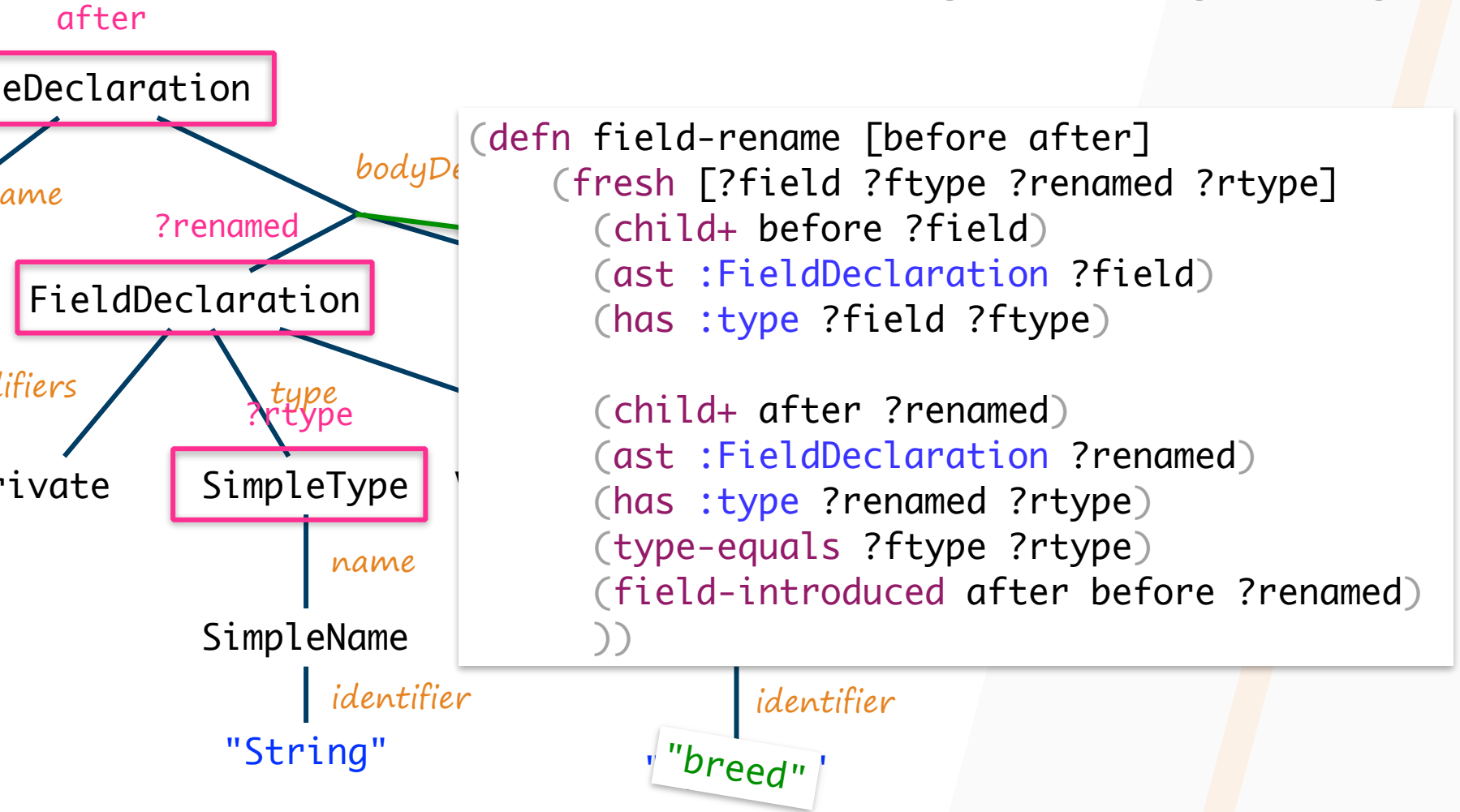
Field rename example - query



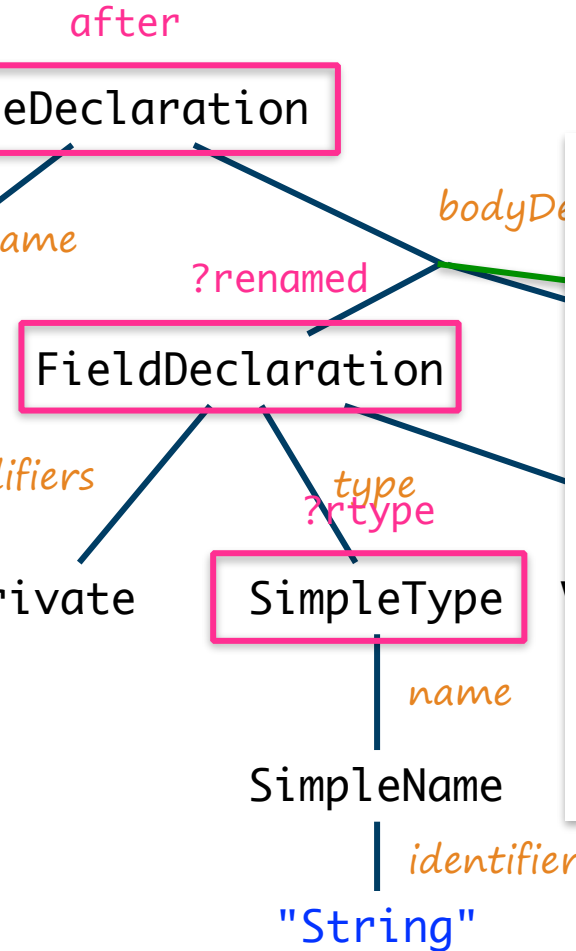
Field rename example - query



Field rename example - query



Field rename example - query



```
(defn field-rename [before after]
  (fresh [?field ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?renamed ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)
  ))
```

Checks that ?renamed did not exist before

"breed"

Field rename example - query

```
(defn field-rename [before after]
  (fresh [?access ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)
    (child+ before ?access)
    (fieldname-accesses ?field ?access)

    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?renamed ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)

    (fails
      (fresh [?caccess]
        (child+ after ?caccess)
        (fieldname-accesses ?renamed ?caccess))))))
```

Field rename example - query

```
(defn field-rename [before after]
  (fresh [?access ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)
    (child+ before ?access)
    (fieldname-accesses ?field ?access)

    Also retrieve accesses to ?field
    (child+ after ?renamed)
    (ast :FieldDeclaration ?renamed)
    (has :type ?renamed ?rtype)
    (type-equals ?ftype ?rtype)
    (field-introduced after before ?renamed)

    (fails
      (fresh [?caccess]
        (child+ after ?caccess)
        (fieldname-accesses ?renamed ?caccess))))))
```

Field rename example - query

```
(defn field-rename [before after]
  (fresh [?access ?ftype ?renamed ?rtype]
    (child+ before ?field)
    (ast :FieldDeclaration ?field)
    (has :type ?field ?ftype)
    (child+ before ?access)
    (fieldname-accesses ?field ?access)))
```

Also retrieve accesses to ?field

```
(child+ after ?renamed)
(ast :FieldDeclaration ?renamed)
(has :type ?renamed ?rtype)
(type-equals ?ftype ?rtype)
(field-introduced after before
```

Also retrieve accesses

```
(fails
  (fresh [?caccess]
    (child+ after ?caccess)
    (fieldname-accesses ?renamed ?caccess))))))
```

Core relations

- General-purpose/logic
 - **fresh** - declare local vars
 - **conde** - or operator
 - **fails** - not operator
- AST navigation
 - **ast** - check node type
 - **has** - get specific child
 - **child+** - get descendant

User study

- Goal: evaluate **usability and expressiveness** of QwalKeko
- Four phases:
 - **Reading** tasks: you get a query, and need to indicate which of the given before/after pieces of code are valid matches
 - First half of the questionnaire
 - **Writing** tasks: given a before/after piece of code, write a query for it
 - Second half of the questionnaire
- You can ask questions during exercises, but only clarifying ones
- You're free to take a break in between tasks
- To get started: timmolderez.be/qwalkeko