JQASSISTANT: IMPLEMENTIERUNG EINES SCANNERS

Softwarevisualisierung
Projektarbeit
von Falk Müller

Inhalt

1. Aufgabe

- Implementierung eines Scanners zur Extraktion von Informationen aus Softwareartefakten als jQAssistant-Plugin
- Zu scannende Software: PHP
- Programmiersprache: Java
- jQAssistant: https://jqassistant.org/

2. PHP

- **P**HP: **H**ypertext **P**reprocessor
- Serverseitige Programmierung
- 80% der Websites mit PHP

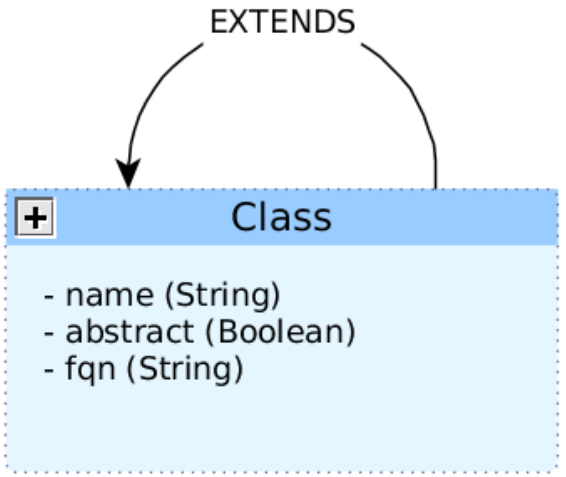

```
<?php
    echo 'Hallo Welt!';
?>
```

2. PHP

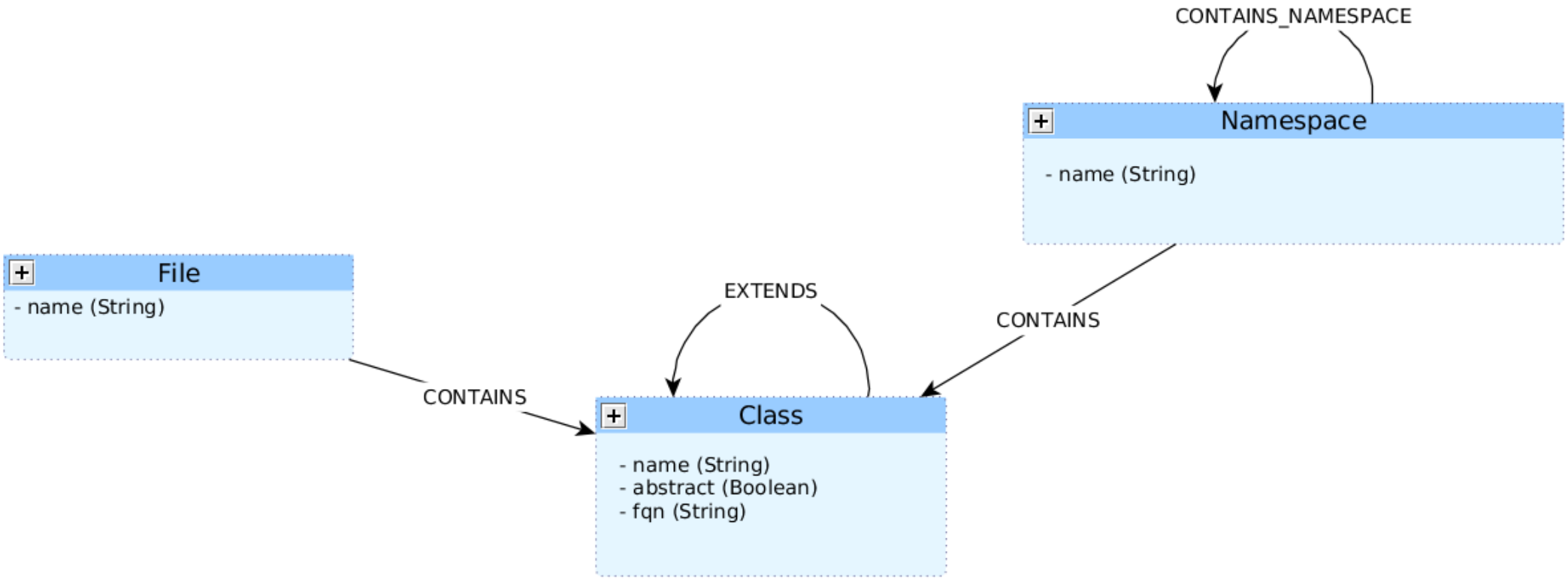
```php
<?php

namespace a\b;

use c\d;
use e\f as g;

class index extends g implements h, d, d\i, \j\k {

    private $_l = 1;
    public static $m;
    const N = 'Konstanter Wert';

    public function __construct(){
        echo "text index contructor";
    }

    public static function m($c, $x = 1){
        $a = new o\p();
        $b = a::test();
    }

    public function test(){
        echo "test";
    }
}
```

```php
<?php

namespace a\b;

function q($x){

    $r = new r();

}

$a = new index();
$a->m("test")
$a->test();
```

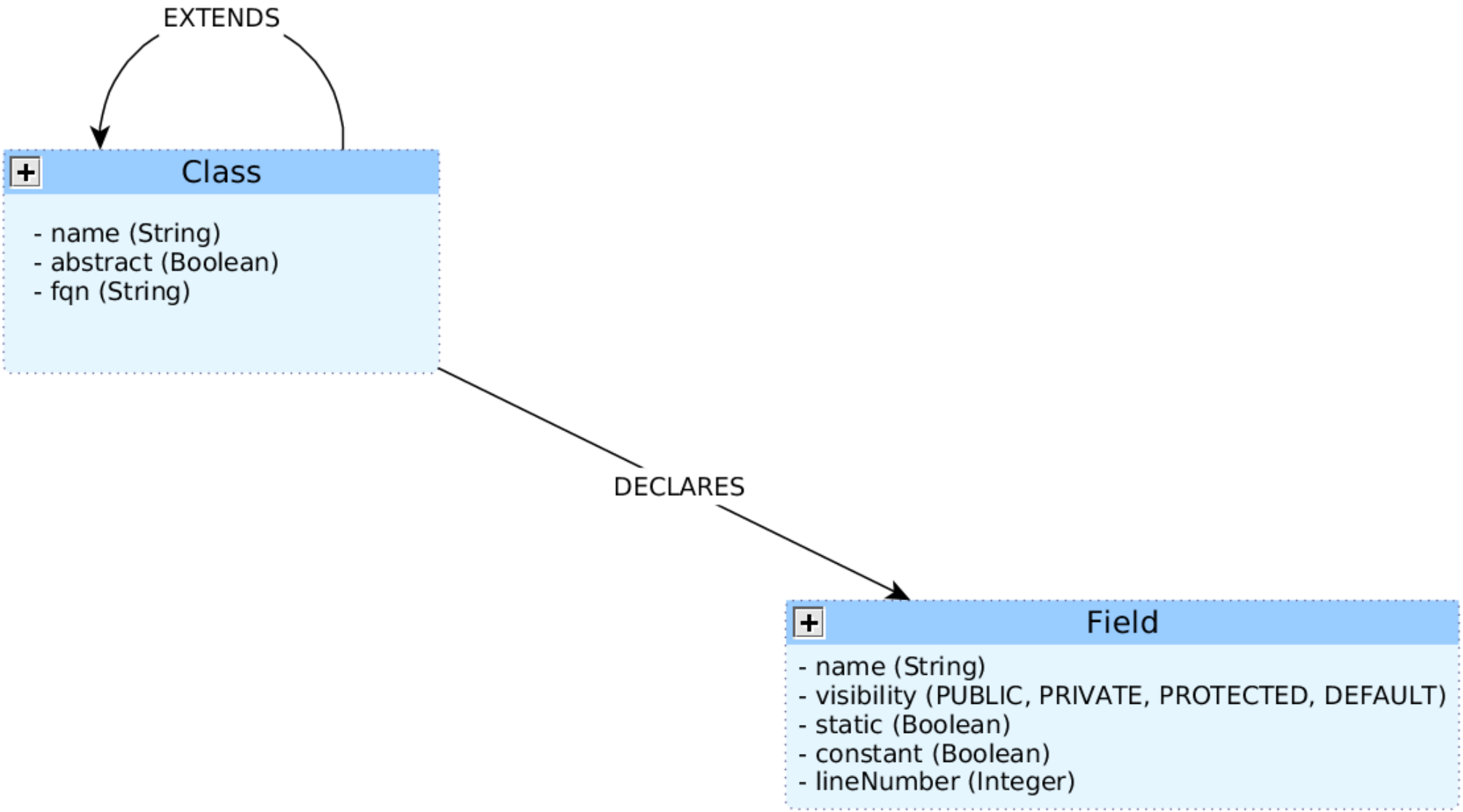
3.1. Struktur



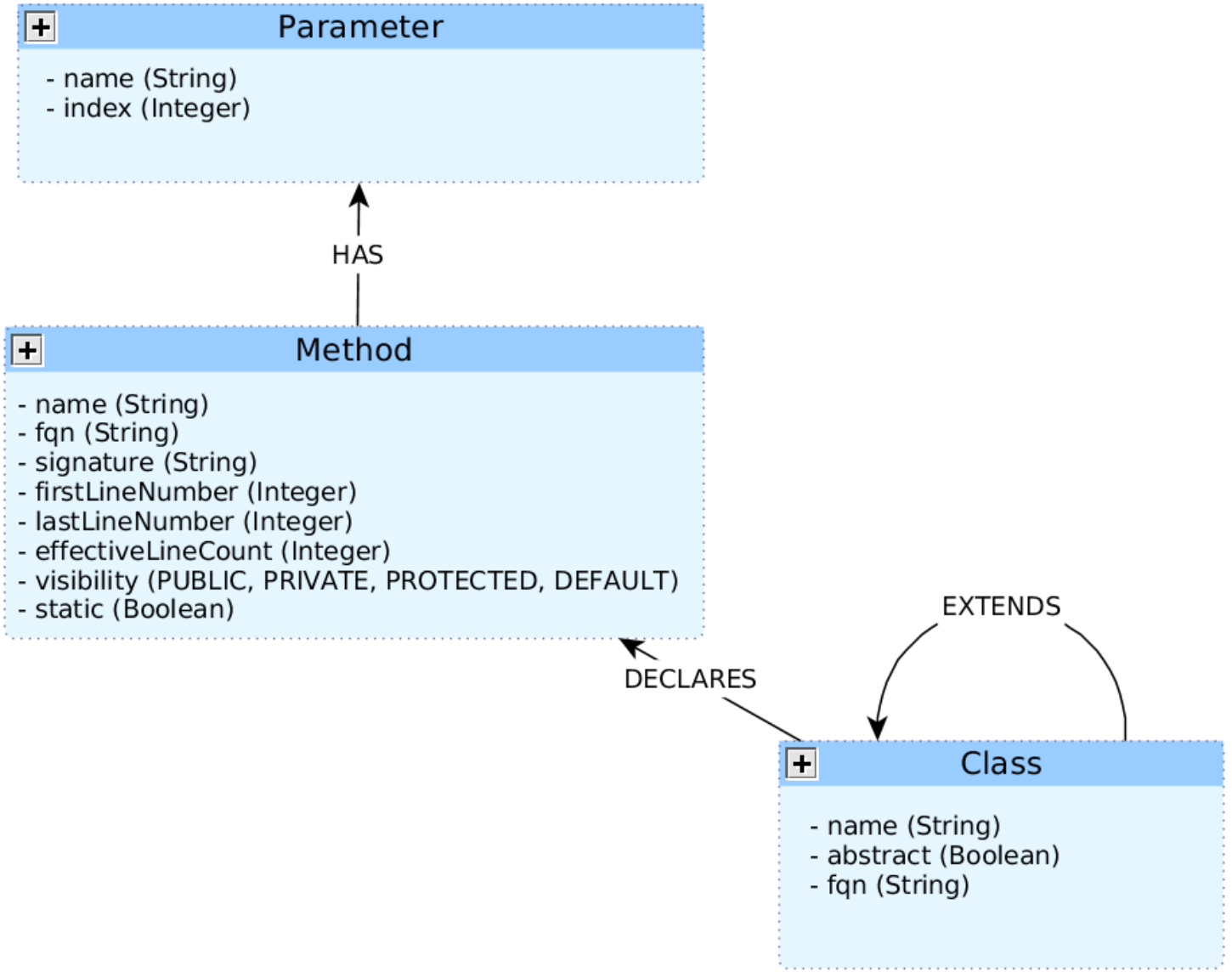
3.1. Struktur


CONTAINS_NAMESPACE
Namespace
- name (String)
File
- name (String)
EXTENDS
CONTAINS
CONTAINS
Class
- name (String)
- abstract (Boolean)
- fqn (String)

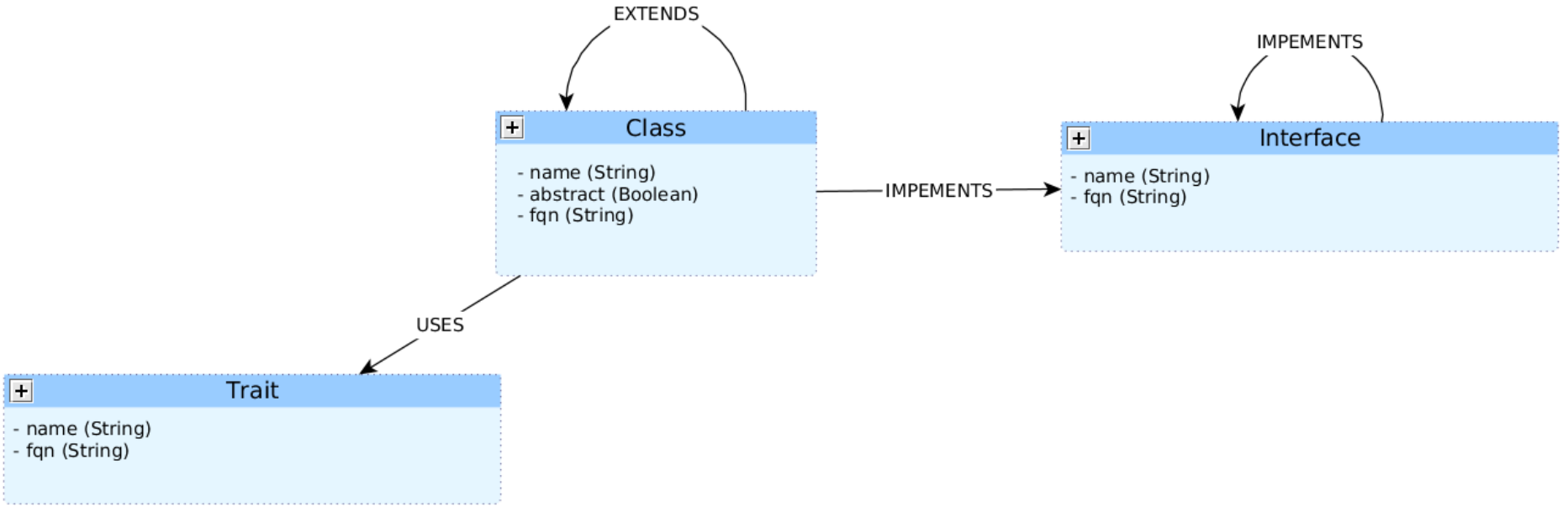
3.1. Struktur


EXTENDS
Class
- name (String)
- abstract (Boolean)
- fqn (String)
DECLARES
Field
- name (String)
- visibility (PUBLIC, PRIVATE, PROTECTED, DEFAULT)
- static (Boolean)
- constant (Boolean)
- lineNumber (Integer)

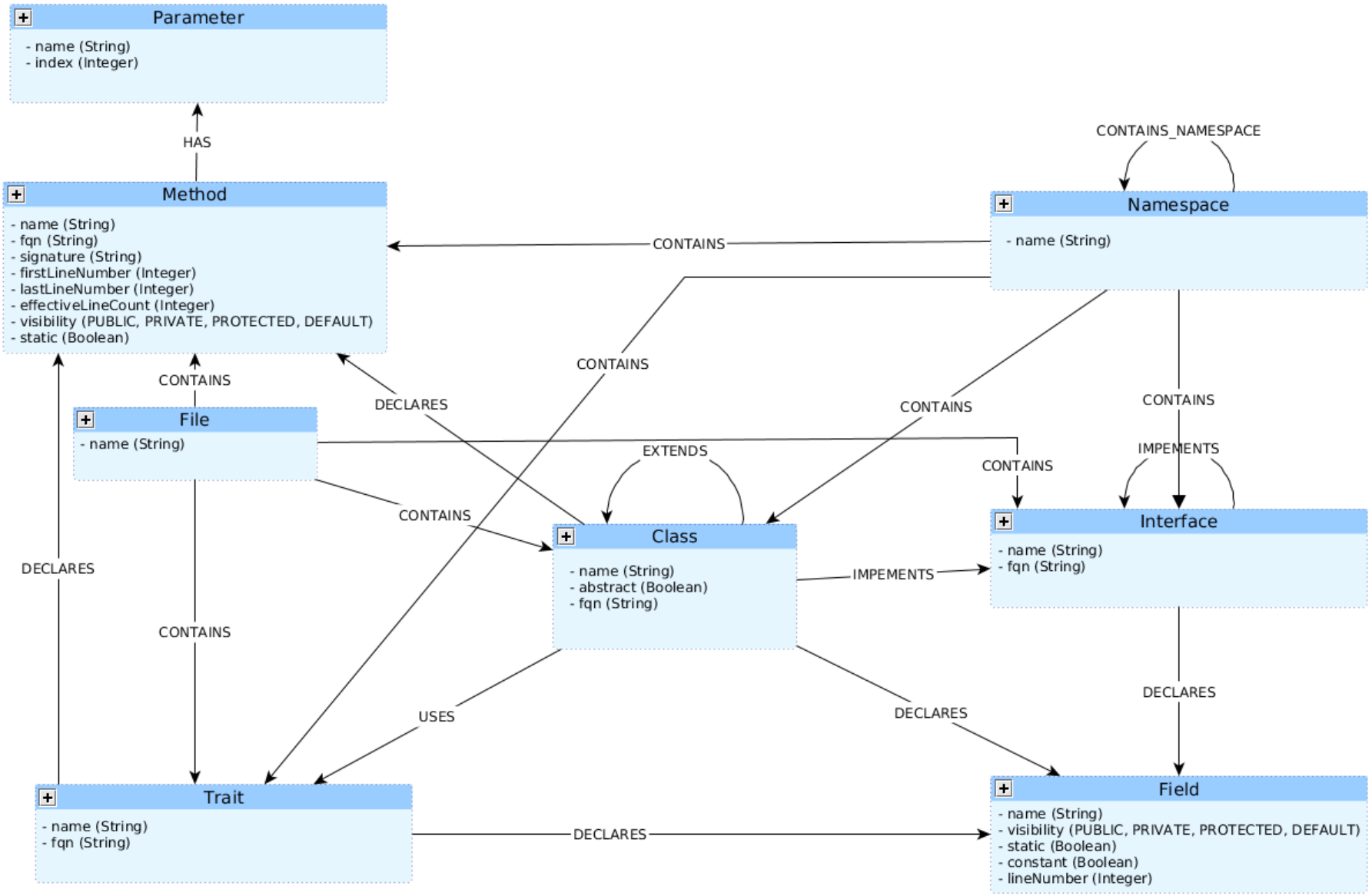
3.1. Struktur


Parameter
- name (String)
- index (Integer)
HAS
Method
- name (String)
- fqn (String)
- signature (String)
- firstLineNumber (Integer)
- lastLineNumber (Integer)
- effectiveLineCount (Integer)
- visibility (PUBLIC, PRIVATE, PROTECTED, DEFAULT)
- static (Boolean)
EXTENDS
DECLARES
Class
- name (String)
- abstract (Boolean)
- fqn (String)

3.1. Struktur


EXTENDS
IMPEMENTS
Class
- name (String)
- abstract (Boolean)
- fqn (String)
IMPEMENTS
Interface
- name (String)
- fqn (String)
USES
Trait
- name (String)
- fqn (String)

3.1. Struktur


Parameter
- name (String)
- index (Integer)
HAS
Method
- name (String)
- fqn (String)
- signature (String)
- firstLineNumber (Integer)
- lastLineNumber (Integer)
- effectiveLineCount (Integer)
- visibility (PUBLIC, PRIVATE, PROTECTED, DEFAULT)
- static (Boolean)
CONTAINS_NAMESPACE
Namespace
- name (String)
CONTAINS
CONTAINS
CONTAINS
CONTAINS
DECLARES
File
- name (String)
EXTENDS
CONTAINS
CONTAINS
IMPEMENTS
CONTAINS
Class
- name (String)
- abstract (Boolean)
- fqn (String)
Interface
- name (String)
- fqn (String)
IMPEMENTS
DECLARES
CONTAINS
USES
DECLARES
DECLARES
Trait
- name (String)
- fqn (String)
DECLARES
Field
- name (String)
- visibility (PUBLIC, PRIVATE, PROTECTED, DEFAULT)
- static (Boolean)
- constant (Boolean)
- lineNumber (Integer)

3.2. Antr4

- Lexer

```
function test(){
    echo 1;
}
```

function	80	Function

3.2. Antr4

- Lexer

```
function test(){
    echo 1;
}
```

function	80	Function
	39	Whitespace
test	213	Label
(	200	OpenRoundBracket
)	201	CloseRoundBracket
{	204	OpenCurlyBracket

3.2. Antr4

- Lexer

```
function test(){
    echo 1;
}
```

function	80	Function
	39	Whitespace
test	213	Label
(	200	OpenRoundBracket
)	201	CloseRoundBracket
{	204	OpenCurlyBracket
	39	Whitespace
echo	62	Echo
	39	Whitespace
1	215	Decimal
;	208	SemiColon
	39	Whitespace

3.2. Antr4

- Lexer

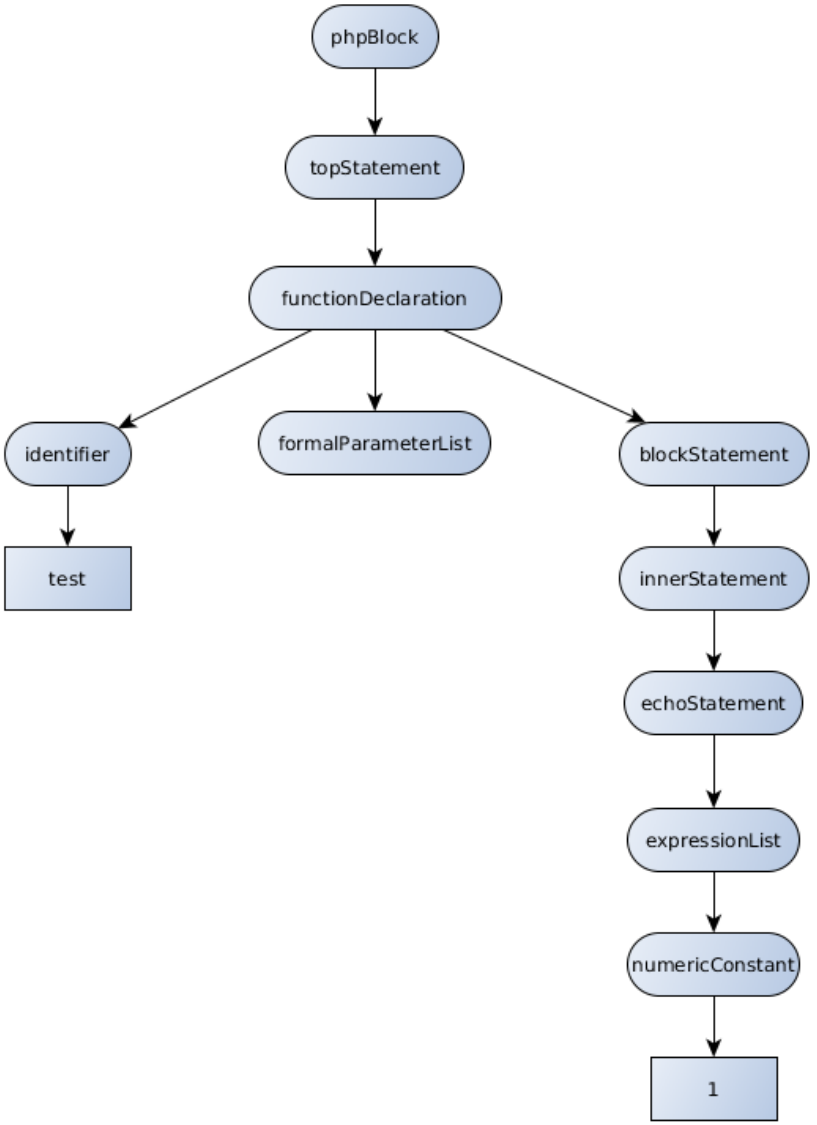
```
function test(){
    echo 1;
}
```

function	80	Function
	39	Whitespace
test	213	Label
(	200	OpenRoundBracket
)	201	CloseRoundBracket
{	204	OpenCurlyBracket
	39	Whitespace
echo	62	Echo
	39	Whitespace
1	215	Decimal
;	208	SemiColon
	39	Whitespace
}	205	CloseCurlyBracket

3.2. Antr4

- Parser

```
function test(){
    echo 1;
}
```



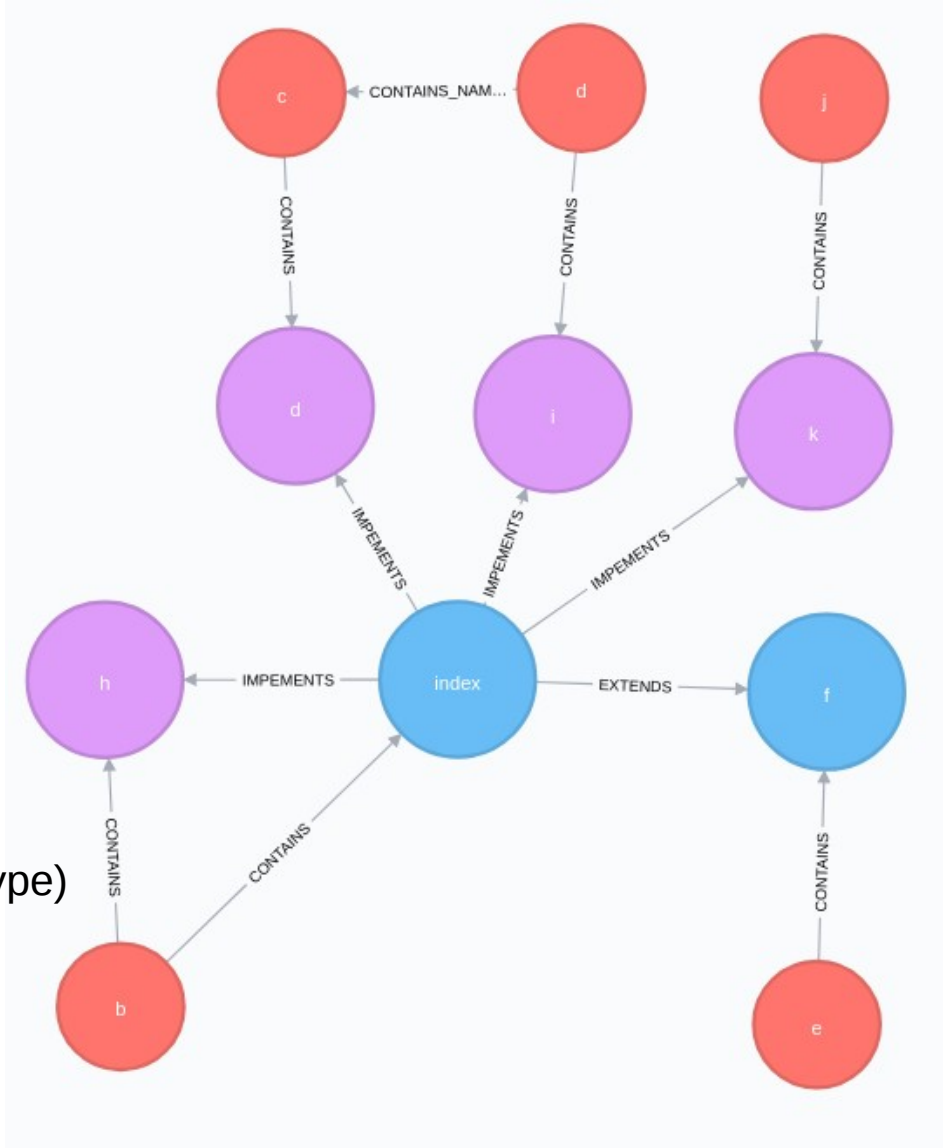
3.3. Umsetzung

- Überführung von Baum zu Modell-Objekten
- Startet bei file-Object
- Sub Parser für Typen (Class, Interface, Trait), Namespace, Field, Constant, Function, Use

```
switch (tree.getClass().getSimpleName()) {
    case "UseDeclarationContentListContext":
        PHPUse u = (new PHPUseParser()).parse(tree);
        useContext.put(u.alias, u);
    case "QualifiedNamespaceNameContext":
        namespace = (new PHPNameSpaceParser(helper)).parse(tree);
    case "ClassDeclarationContext":
        fileDescriptor.getClasses().add(
                (new PHPTypeParser(helper, namespace, useContext)).parse(tree));
    case "FunctionDeclarationContext":
        fileDescriptor.getFunctions().add(
                (new PHPFunctionParser(helper, namespace, useContext)).parse(tree));
}
```

4. Beispiel

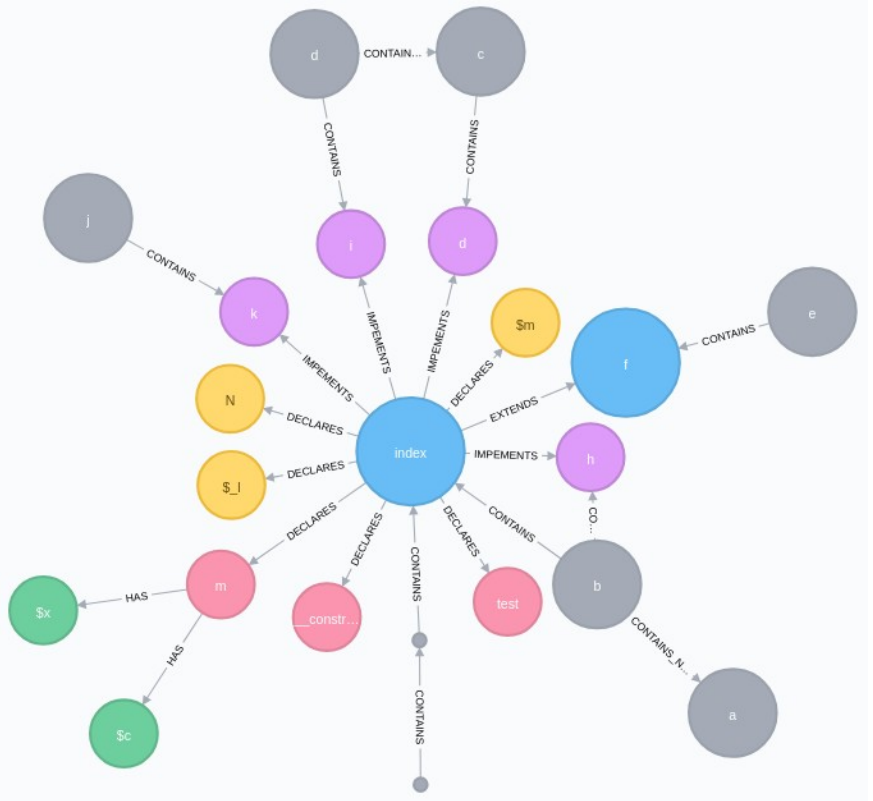
```php
<?php

namespace a\b;

use c\d;
use e\f as g;

class index extends g
implements h, d, d\i, \j\k {}
```



MATCH
(namespace:Namespace)-[:CONTAINS]->(entity:Type)
RETURN namespace, entity

4. Beispiel

```php
<?php

namespace a\b;

use c\d;
use e\f as g;

class index extends g implements h, d, d\i, \j\k {

    private $_l = 1;
    public static $m;
    const N = 'Konstanter Wert';

    public function __construct(){
        echo "text index contructor";
    }

    public static function m($c, $x = 1){
        $a = new o\p();
        $b = a::test();
    }

    public function test(){
        echo "test";
    }
}
```



Quellen

- 80% of the web powered by PHP: https://haydenjames.io/80-percent-web-powered-by-php/
- jQAssistant: https://jqassistant.org/
- jQAssistant Tutorial: https://101.jqassistant.org/scanner-plugin/readme.html
- Php: http://php.net

Vielen Dank für Ihre Aufmerksamkeit

Fragen?