

Java++;

Hugo Correia, 66995

Miguel Borges, 67041

Pedro Rijo, 67060

BASIC FUNCTIONALITY

Field Assertion - Example

```
@Assertion("field > 0")
```

```
int field = -2;
```

```
println(field);
```

```
> "The assertion field > 0 is false"
```

```
int field = 2;
```

```
println(field);
```

```
> 2
```

Field Initialization - Example

```
@Assertion("true")
```

```
int initField;
```

```
println(initField);
```

```
> "Error: initField was not initialized"
```

```
initField = 3;
```

```
println(initField);
```

```
> 3
```

Method Assertion - Example

```
@Assertion("$1>0 && $_ > $1")  
public int m1(int x) {  
    return x + 3;  
}
```

```
println( m1(1) );  
    > 4
```

```
m1(-10);  
    > "Method assertion ($1>0 && $_ > $1) failed"
```

Extensions

- Method Assertions on entry and on method return
- Constructor Assertion
- Array Assertion
- Cast Assertion
- Exception Assertion

Method entry - Goal

Verify if an assertion is valid before the execution of a method.

```
@Assertion(before="$1>0")  
public int m1(int x) {  
    ...  
}
```

```
//Should be an Invalid Access  
m1(-10);
```

Method entry - Solution

Added a new field in `@Assertion` annotation.

```
@Assertion(before="$1>0", value="$>0")
```

Note : `before` can't contain "\$_"

Method entry - Example

```
@Assertion(before="$1>0")  
public int m1(int x) {  
    return x - 3;  
}
```

```
println(m1(1));  
> -2
```

```
println(m1(0));  
> "Method assertion ( $1>0 ) failed"
```

Constructor Assertion - Goal

Checks if arguments verify the defined assertion

Inherits assertions from super class constructors

Assertion is verified before the construction of the object.

Constructor Assertion - Example

```
public class Foo {  
    int field;  
    @Assertion("$1 > 0")  
    public Foo(int field) {  
        this.field = field;  
    }  
}
```

```
new TestConstructor(10);  
> Ok
```

```
new TestConstructor(-3);  
> "Constructor assertion not verified"
```

Array Assertion - Goal

Avoid access to a non-initialized position of an array.

```
int[] array = new int[10];
```

```
//Should be an Invalid Access
```

```
int x = array[1];
```

Array Assertion - Solution

Created a new annotation for classes.

```
@ArrayInitializationAssertion  
public class Foo() {...}
```

Array Assertion - Solution

If the class is annotated, we replace all accesses to arrays by a call to new defined static method.

```
Foo foo = new Foo();
```

```
//Before interception:  
foo.ints[1] = 2;
```

```
//After interception:  
ArrayInterceptor.arrayWriteInt(foo.array, 1, 2)
```

Array Assertion - Solution

For each array position we have a Boolean value that marks the position as initialized or not.

On Initialization:

Initialize the **Boolean** array to **false**

Array Assertion - Solution

On Write access:

Initialize the index we are writing to with **true**

On Read access:

```
if index is not initialized  
    throw new RuntimeException();
```


Array Assertion – Example

```
Integer[] integer = new Integer[20];
```

```
integer[1] = 1;
```

```
println(integer[1]);
```

```
> 1
```

```
println(integer[10]);
```

```
> "Array not initialized"
```

Cast Assertion - Goal

Limit the possible casts allowed to a asserted class

```
@CastAssertion({"Foo", "Bar"})  
public class Baz{...}
```

Cast Assertion - Solution

Created a new annotation with an array of allowed classes to be casted.

When someone makes a cast of the annotated class, we verify if the cast is valid.

Note : Cast to own class are always allowed

Cast Assertion - Example

```
@CastAssertion({"Foo", "Bar"})  
public class Baz{...}
```

```
Foo t = (Foo) new Baz();  
> Ok
```

```
Object t = (Object) new Baz();  
> "Cast not allowed"
```

Exception Assertion - Goal

Allows a programmer to call a specified method on chosen exceptions.

Exception Assertion - Solution

Created a new annotation that receives a list of defined Exceptions and the method it should call.

```
@ExceptionAssertion( exception = {"Exception"},  
                    method = "catchMethod" )
```

Global behavior to all exceptions thrown in the class

Exception Assertion - Example

```
@ExceptionAssertion(exception = {"NullPointerException"},
                    method = "catchMethod")

public class Bar {}

try{
    //throws NullPointerException
    new Bar().m1();
} catch (NullPointerException e) {
    //will call catchMethod()
}

//throws FileNotFoundException
new TestExceptionHandler().m2();
```

Questions ?