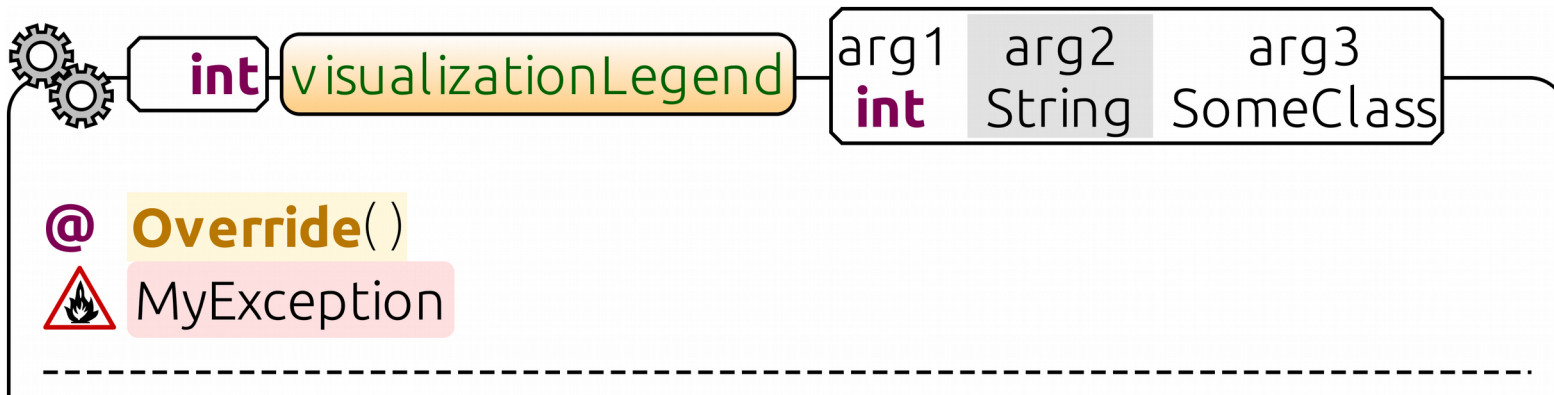
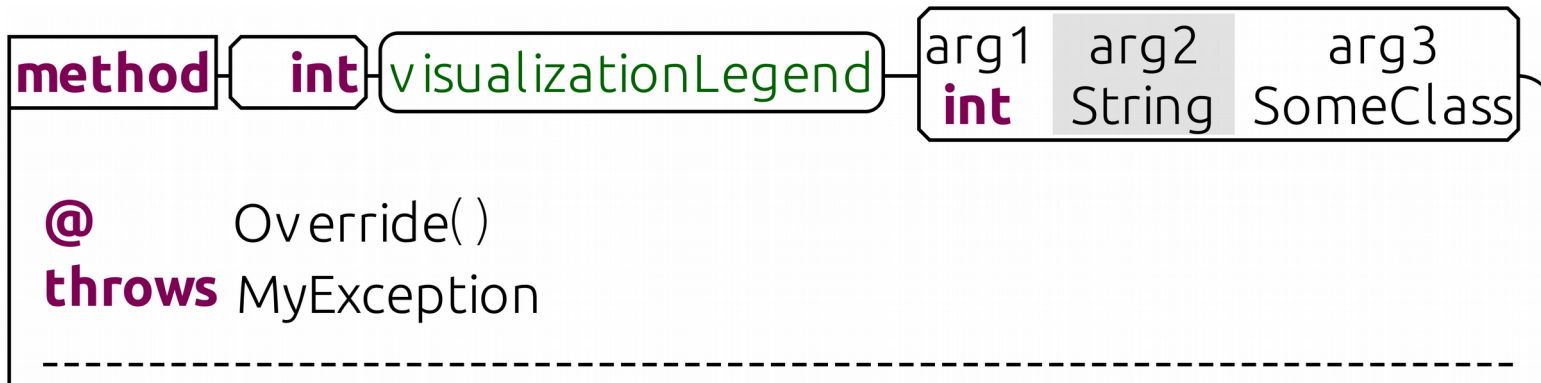


@Override

```
public int legend(int arg1, String arg2, SomeClass arg3)
throws MyException
{
```



```
int local1;  
local1 = 42;  
String local2 = arg2;
```

```
int local1  
local1 ← 42  
String local2 ← arg2
```

```
int local1  
local1 ← 42  
String local2 ← arg2
```


```
if (arg1 >= 5) {  
    System.out.println("Arg1 is greater or equal 4");  
    return 1;  
} else if (arg1 < 0) {  
    System.out.println("Arg1 is negative");  
}
```

```
if arg1 ≥ 5  
    System.out.println("Arg1 is greater or equal 4")  
    return 1  
else if arg1 < 0  
    System.out.println("Arg1 is negative")
```

```
if arg1 ≥ 5  
    System.out.println("Arg1 is greater or equal 4")  
    ↩ 1  
if arg1 < 0  
    System.out.println("Arg1 is negative")
```

```
else {  
    assert(arg1 >=0 && arg1 <5);  
    System.out.print("Arg1 is within the proper range.");  
}
```

```
-----  
assert (arg1 ≥ 0 ∧ arg1 < 5)  
System.out.print("Arg1 is within the proper range.")
```

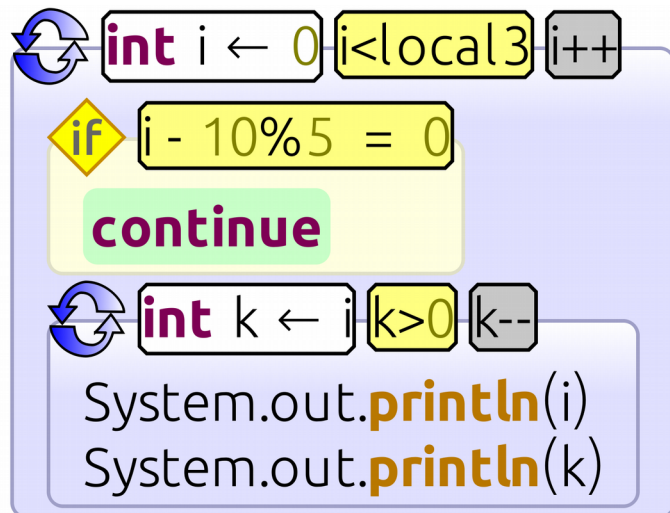
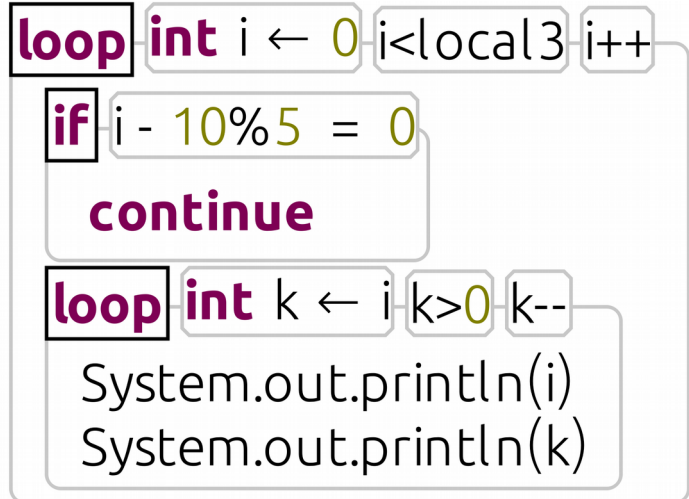
```
-----  
 (arg1 ≥ 0 ∧ arg1 < 5)  
System.out.print("Arg1 is within the proper range.")
```



```

for(int i = 0; i<local3; i++) {
    if (i - 10 % 5 == 0) continue;
    for (int k = i; k>0; k--) {
        System.out.println(i);
        System.out.println(k);
    }
}

```



```
try {  
    SomeClass sc = new SomeClass();  
    Object baz_result = sc.baz(local1, this, null, 0, arg2);  
    local1 = ((Cell)baz_result).value;  
}
```

try

```
SomeClass sc ← new SomeClass()  
Object baz_result ← sc.baz(local1 this null 0 arg2)  
local1 ← ((Cell)baz_result).value
```



```
SomeClass sc ← new SomeClass()  
Object baz_result ← sc.baz(local1 this null 0 arg2)  
local1 ← ((Cell)baz_result).value
```