# Как анализаторы кода ищут ошибки и дефекты безопасности

Сергей
Васильев

# WHOAMI

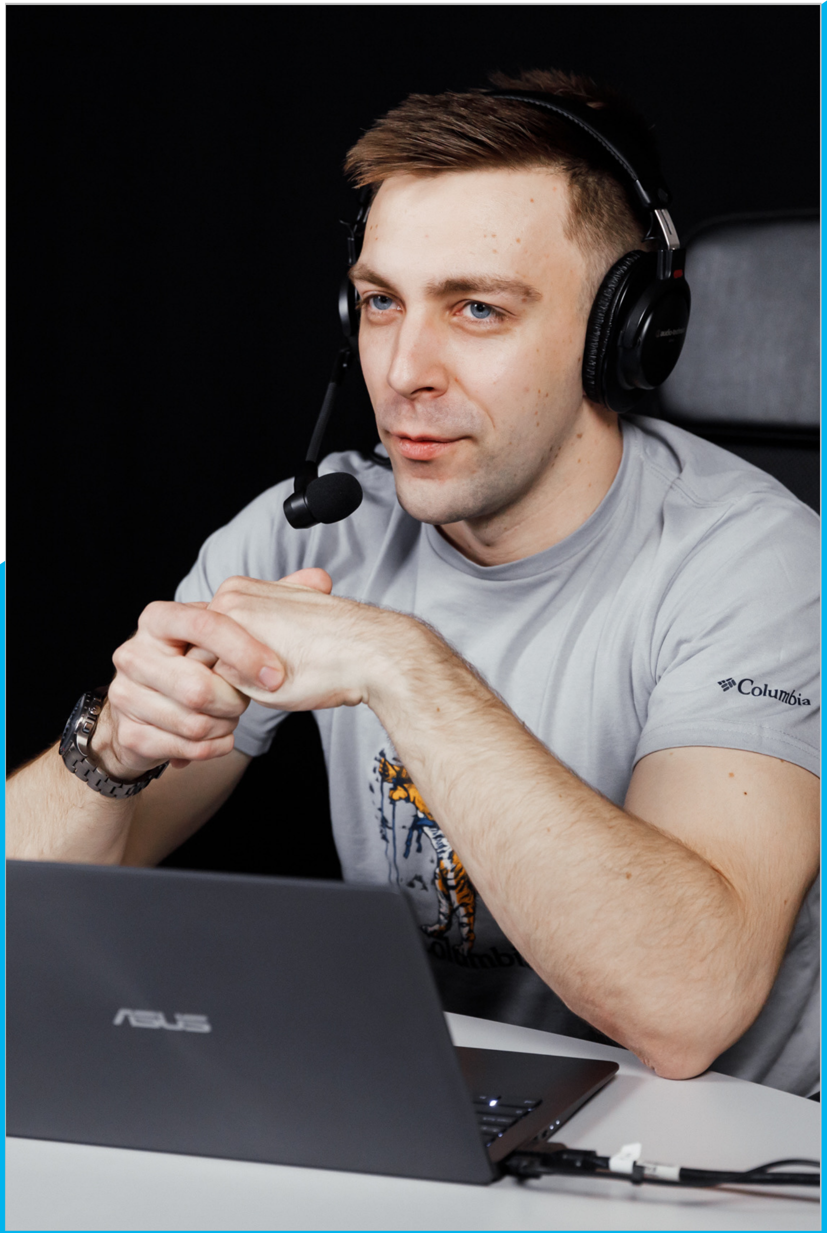
## Сергей Васильев

Head of DevRel в PVS-Studio LLC

8 лет в статическом анализе

В прошлом:
- C# Analyzer Team Leader
- Tools & DevOps Team Leader
- C++, C# developer

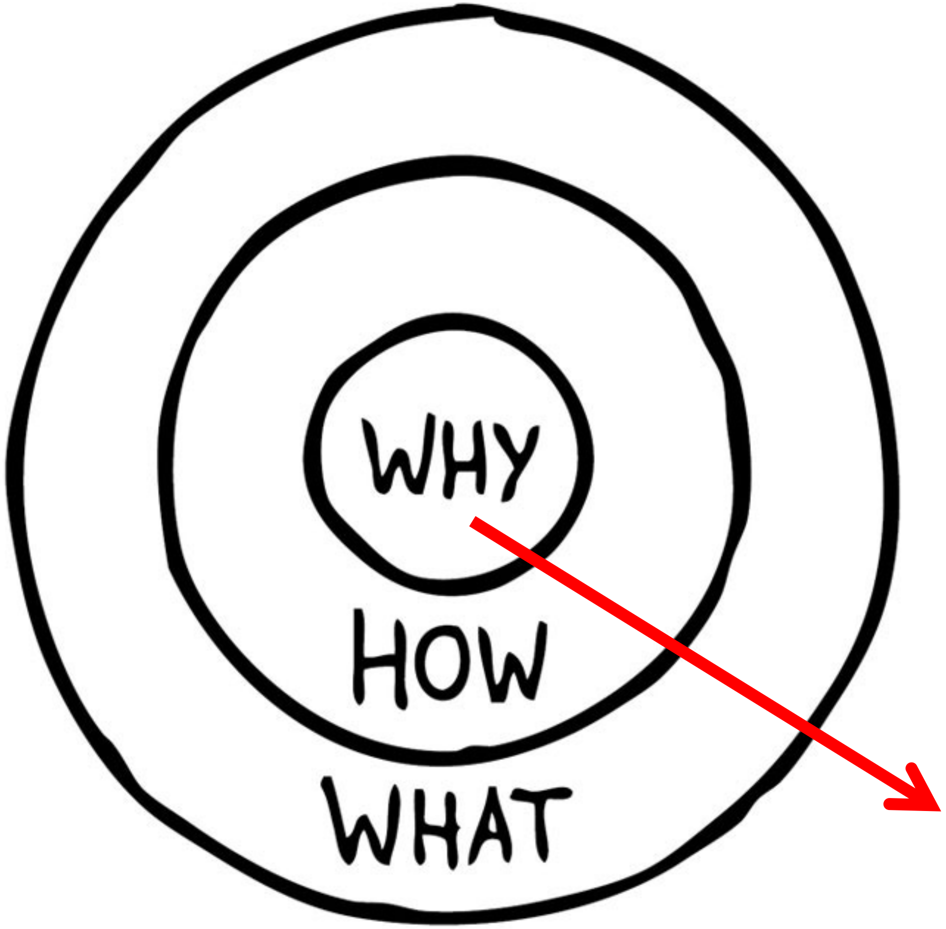
Пишу на habr, выступаю.



Sources

Static analysis

Report (issues)


WHY
HOW
WHAT

# Статический анализ: зачем?

# Статический анализ: зачем?

Рефакторинг

Поиск ошибок

Поиск дефектов безопасности

# Bouncy Castle

```
public static boolean isKeyRing(byte[] blob) throws IOException {
    BCPGInputStream bIn
      = new BCPGInputStream(new ByteArrayInputStream(blob));
    int tag = bIn.nextPacketTag();

    return     tag == PacketTags.PUBLIC_KEY
            || tag == PacketTags.PUBLIC_SUBKEY
            || tag == PacketTags.SECRET_KEY
            || tag == PacketTags.SECRET_KEY;
}
```

# Bouncy Castle

```
return    tag == PacketTags.PUBLIC_KEY
       || tag == PacketTags.PUBLIC_SUBKEY
       || tag == PacketTags.SECRET_KEY
       || tag == PacketTags.SECRET_KEY;
```

- Condition 'tag == PacketTags.SECRET_KEY' covered by subsequent condition 'tag == PacketTags.SECRET_KEY'
- Duplicate condition 'tag == PacketTags.SECRET_KEY'
- Condition 'tag == PacketTags.SECRET_KEY' is always 'false'
- Duplicate condition 'tag == PacketTags.SECRET_KEY'

V6001 There are identical sub-expressions 'tag == PacketTags.SECRET_KEY' to the left and to the right of the '||' operator.

# Hibernate

```
private static final Map<Class, String> annotationToXml;

static {
    annotationToXml = new HashMap<Class, String>();
    annotationToXml.put( Entity.class,
                         "entity" );
    annotationToXml.put( MappedSuperclass.class,
                         "mapped-superclass" );
    ....
    annotationToXml.put( ConstructorResult.class,
                         "constructor-result" );
}
```

# Hibernate

```
private static final Map<Class, String> annotationToXml;

static {
    annotationToXml = new HashMap<Class, String>();
    annotationToXml.put( Entity.class,
                         "entity" );
    annotationToXml.put( MappedSuperclass.class,
                         "mapped-superclass" );
    annotationToXml.put( Embeddable.class,
                         "embeddable" );
    annotationToXml.put( Table.class,
                         "table" );
    annotationToXml.put( SecondaryTable.class,
                         "secondary-table" );
    annotationToXml.put( SecondaryTables.class,
                         "secondary-table" );
    annotationToXml.put( PrimaryKeyJoinColumn.class,
                         "primary-key-join-column" );
    annotationToXml.put( PrimaryKeyJoinColumns.class,
                         "primary-key-join-column" );
    annotationToXml.put( IdClass.class,
                         "id-class" );
    annotationToXml.put( Inheritance.class,
                         "inheritance" );
    annotationToXml.put( DiscriminatorValue.class,
                         "discriminator-value" );
    annotationToXml.put( DiscriminatorColumn.class,
                         "discriminator-column" );
    annotationToXml.put( SequenceGenerator.class,
                         "sequence-generator" );
    annotationToXml.put( TableGenerator.class,
                         "table-generator" );
    annotationToXml.put( NamedEntityGraph.class,
                         "named-entity-graph" );
    annotationToXml.put( NamedEntityGraphs.class,
                         "named-entity-graph" );
    annotationToXml.put( NamedQuery.class,
                         "named-query" );
    annotationToXml.put( NamedQueries.class,
                         "named-query" );
    annotationToXml.put( NamedNativeQuery.class,
                         "named-native-query" );
    annotationToXml.put( NamedNativeQueries.class,
                         "named-native-query" );
    annotationToXml.put( NamedStoredProcedureQuery.class,
                         "named-stored-procedure-query" );
    annotationToXml.put( NamedStoredProcedureQueries.class,
                         "named-stored-procedure-query" );
    annotationToXml.put( SqlResultSetMapping.class,
                         "sql-result-set-mapping" );
    annotationToXml.put( SqlResultSetMappings.class,
                         "sql-result-set-mapping" );
    annotationToXml.put( ExcludeDefaultListeners.class,
                         "exclude-default-listeners" );
    annotationToXml.put( ExcludeSuperclassListeners.class,
                         "exclude-superclass-listeners" );
    annotationToXml.put( AccessType.class,
                         "access" );
    annotationToXml.put( AttributeOverride.class,
                         "attribute-override" );
    annotationToXml.put( AttributeOverrides.class,
                         "attribute-override" );
    annotationToXml.put( AttributeOverride.class,
                         "association-override" );
    annotationToXml.put( AttributeOverrides.class,
                         "association-override" );
    annotationToXml.put( AttributeOverride.class,
                         "map-key-attribute-override" );
    annotationToXml.put( AttributeOverrides.class,
                         "map-key-attribute-override" );
    annotationToXml.put( Id.class,
                         "id" );
    annotationToXml.put( EmbeddedId.class,
                         "embedded-id" );
    annotationToXml.put( GeneratedValue.class,
                         "generated-value" );
    annotationToXml.put( Column.class,
                         "column" );
    annotationToXml.put( Columns.class,
                         "column" );
    annotationToXml.put( Temporal.class,
                         "temporal" );
    annotationToXml.put( Lob.class,
                         "lob" );
    annotationToXml.put( Enumerated.class,
                         "enumerated" );
    annotationToXml.put( Version.class,
                         "version" );
    annotationToXml.put( Transient.class,
                         "transient" );
    annotationToXml.put( Basic.class,
                         "basic" );
    annotationToXml.put( Embedded.class,
                         "embedded" );
    annotationToXml.put( ManyToOne.class,
                         "many-to-one" );
    annotationToXml.put( OneToOne.class,
                         "one-to-one" );
    annotationToXml.put( OneToMany.class,
                         "one-to-many" );
    annotationToXml.put( ManyToMany.class,
                         "many-to-many" );
    annotationToXml.put( Any.class,
                         "any" );
    annotationToXml.put( ManyToAny.class,
                         "many-to-any" );
    annotationToXml.put( JoinTable.class,
                         "join-table" );
    annotationToXml.put( JoinColumn.class,
                         "join-column" );
    annotationToXml.put( JoinColumns.class,
                         "join-column" );
    annotationToXml.put( MapKey.class,
                         "map-key" );
    annotationToXml.put( OrderBy.class,
                         "order-by" );
    annotationToXml.put( EntityListeners.class,
                         "entity-listeners" );
    annotationToXml.put( PrePersist.class,
                         "pre-persist" );
    annotationToXml.put( PreRemove.class,
                         "pre-remove" );
    annotationToXml.put( PreUpdate.class,
                         "pre-update" );
    annotationToXml.put( PostPersist.class,
                         "post-persist" );
    annotationToXml.put( PostRemove.class,
                         "post-remove" );
    annotationToXml.put( PostUpdate.class,
                         "post-update" );
    annotationToXml.put( PostLoad.class,
                         "post-load" );
    annotationToXml.put( CollectionTable.class,
                         "collection-table" );
    annotationToXml.put( MapKeyClass.class,
                         "map-key-class" );
    annotationToXml.put( MapKeyTemporal.class,
                         "map-key-temporal" );
    annotationToXml.put( MapKeyEnumerated.class,
                         "map-key-enumerated" );
    annotationToXml.put( MapKeyColumn.class,
                         "map-key-column" );
    annotationToXml.put( MapKeyJoinColumn.class,
                         "map-key-join-column" );
    annotationToXml.put( MapKeyJoinColumns.class,
                         "map-key-join-column" );
    annotationToXml.put( OrderColumn.class,
                         "order-column" );
    annotationToXml.put( Cacheable.class,
                         "cacheable" );
    annotationToXml.put( Index.class,
                         "index" );
    annotationToXml.put( ForeignKey.class,
                         "foreign-key" );
    annotationToXml.put( Convert.class,
                         "convert" );
    annotationToXml.put( Converts.class,
                         "convert" );
    annotationToXml.put( ConstructorResult.class,
                         "constructor-result" );
}
```

# Hibernate

```
private static final Map<Class, String> annotationToXml;

static {
    annotationToXml = new HashMap<Class, String>();
    annotationToXml.put( Entity.class,
                         "entity" );
    annotationToXml.put( MappedSuperclass.class,
                         "mapped-superclass" );
    annotationToXml.put( Embeddable.class,
                         "embeddable" );
    annotationToXml.put( Table.class,
                         "table" );
    annotationToXml.put( SecondaryTable.class,
                         "secondary-table" );
    annotationToXml.put( SecondaryTables.class,
                         "secondary-table" );
    annotationToXml.put( PrimaryKeyJoinColumn.class,
                         "primary-key-join-column" );
    annotationToXml.put( PrimaryKeyJoinColumns.class,
                         "primary-key-join-column" );
    annotationToXml.put( IdClass.class,
                         "id-class" );
    annotationToXml.put( Inheritance.class,
```

```
annotationToXml.put( ExcludeSuperclassListeners.class,
                     "exclude-superclass-listeners" );
annotationToXml.put( AccessType.class,
                     "access" );
annotationToXml.put( AttributeOverride.class,
                     "attribute-override" );
annotationToXml.put( AttributeOverrides.class,
                     "attribute-override" );
annotationToXml.put( AttributeOverride.class,
                     "association-override" );
annotationToXml.put( AttributeOverrides.class,
                     "association-override" );
annotationToXml.put( AttributeOverride.class,
                     "map-key-attribute-override" );
annotationToXml.put( AttributeOverrides.class,
                     "map-key-attribute-override" );
annotationToXml.put( Id.class,
                     "id" );
annotationToXml.put( EmbeddedId.class,
                     "embedded-id" );
annotationToXml.put( GeneratedValue.class,
                     "generated-value" );
annotationToXml.put( Column.class,
```

```
annotationToXml.put( JoinColumn.class,
                     "join-column" );
annotationToXml.put( JoinColumns.class,
                     "join-column" );
annotationToXml.put( MapKey.class,
                     "map-key" );
annotationToXml.put( OrderBy.class,
                     "order-by" );
annotationToXml.put( EntityListeners.class,
                     "entity-listeners" );
annotationToXml.put( PrePersist.class,
                     "pre-persist" );
annotationToXml.put( PreRemove.class,
                     "pre-remove" );
annotationToXml.put( PreUpdate.class,
                     "pre-update" );
annotationToXml.put( PostPersist.class,
                     "post-persist" );
annotationToXml.put( PostRemove.class,
                     "post-remove" );
annotationToXml.put( PostUpdate.class,
                     "post-update" );
annotationToXml.put( PostLoad.class,
```

```
annotationToXml.put( AttributeOverride.class,  "attribute-override" );
annotationToXml.put( AttributeOverrides.class, "attribute-override" );
annotationToXml.put( AttributeOverride.class,  "association-override" );
annotationToXml.put( AttributeOverrides.class, "association-override" );
annotationToXml.put( AttributeOverride.class,  "map-key-attribute-override" );
annotationToXml.put( AttributeOverrides.class, "map-key-attribute-override" );
```

# Hibernate

```
private static final Map<Class, String> annotationToXml;

static {
    annotationToXml = new HashMap<Class, String>();
    annotationToXml.put( Entity.class,
                         "entity" );
    annotationToXml.put( MappedSuperclass.class,
                         "mapped-superclass" );
    annotationToXml.put( Embeddable.class,
                         "embeddable" );
    annotationToXml.put( Table.class,
                         "table" );
    annotationToXml.put( SecondaryTable.class,
                         "secondary-table" );
    annotationToXml.put( SecondaryTables.class,
                         "secondary-table" );
    annotationToXml.put( PrimaryKeyJoinColumn.class,
                         "primary-key-join-column" );
    annotationToXml.put( PrimaryKeyJoinColumns.class,
                         "primary-key-join-column" );
    annotationToXml.put( IdClass.class,
                         "id-class" );
    annotationToXml.put( Inheritance.class,
                         "inheritance" );
    annotationToXml.put( DiscriminatorValue.class,
                         "discriminator-value" );
    annotationToXml.put( DiscriminatorColumn.class,
                         "discriminator-column" );
    annotationToXml.put( SequenceGenerator.class,
                         "sequence-generator" );
    annotationToXml.put( TableGenerator.class,
                         "table-generator" );
    annotationToXml.put( NamedEntityGraph.class,
                         "named-entity-graph" );
    annotationToXml.put( NamedEntityGraphs.class,
                         "named-entity-graph" );
    annotationToXml.put( NamedQuery.class,
                         "named-query" );
    annotationToXml.put( NamedQueries.class,
                         "named-query" );
    annotationToXml.put( NamedNativeQuery.class,
                         "named-native-query" );
    annotationToXml.put( NamedNativeQueries.class,
                         "named-native-query" );
    annotationToXml.put( NamedStoredProcedureQuery.class,
                         "named-stored-procedure-query" );
    annotationToXml.put( NamedStoredProcedureQueries.class,
                         "named-stored-procedure-query" );
    annotationToXml.put( SqlResultSetMapping.class,
                         "sql-result-set-mapping" );
    annotationToXml.put( SqlResultSetMappings.class,
                         "sql-result-set-mapping" );
    annotationToXml.put( ExcludeDefaultListeners.class,
                         "exclude-default-listeners" );
```

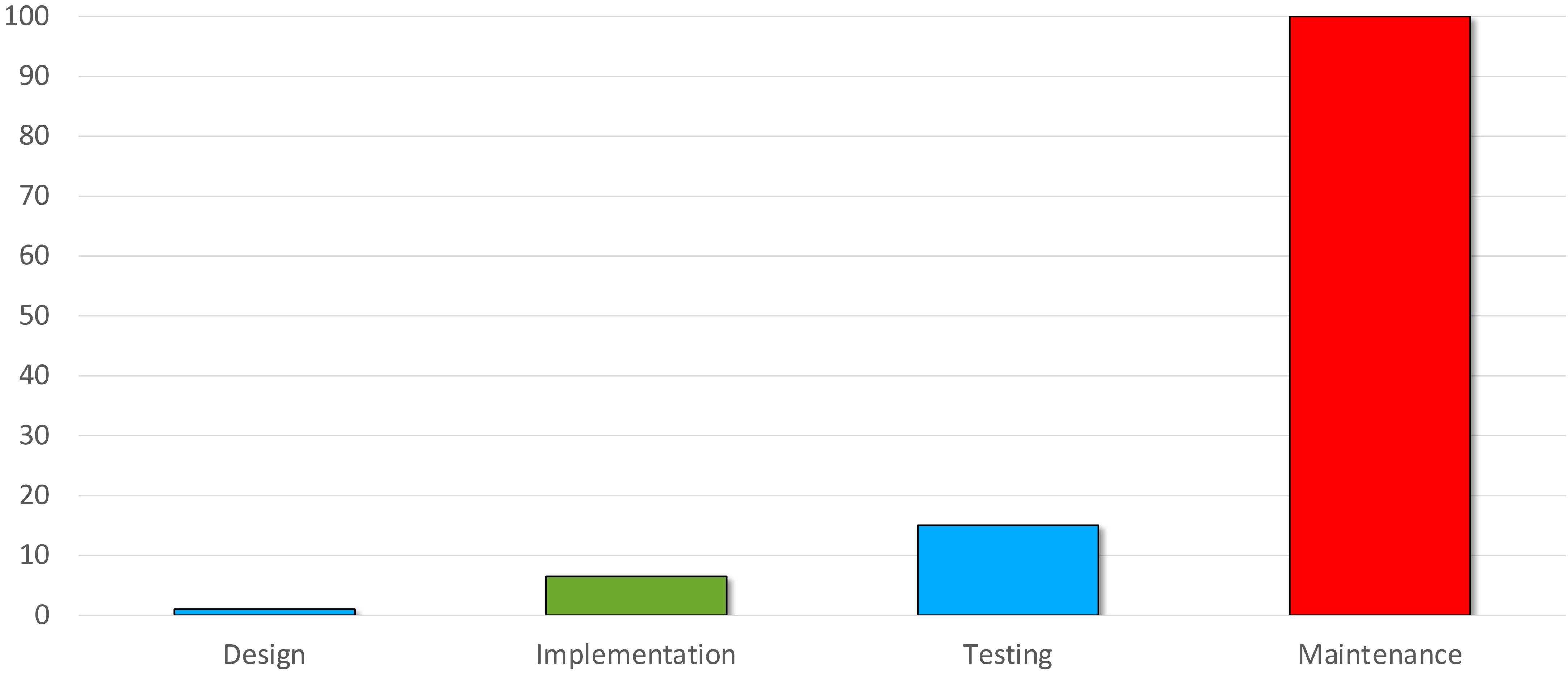
```
annotationToXml.put( ExcludeSuperclassListeners.class,
                     "exclude-superclass-listeners" );
annotationToXml.put( AccessType.class,
                     "access" );
annotationToXml.put( AttributeOverride.class,
                     "attribute-override" );
annotationToXml.put( AttributeOverrides.class,
                     "attribute-override" );
annotationToXml.put( AttributeOverride.class,
                     "association-override" );
annotationToXml.put( AttributeOverrides.class,
                     "association-override" );
annotationToXml.put( AttributeOverride.class,
                     "map-key-attribute-override" );
annotationToXml.put( AttributeOverrides.class,
                     "map-key-attribute-override" );
annotationToXml.put( Id.class,
                     "id" );
annotationToXml.put( EmbeddedId.class,
                     "embedded-id" );
annotationToXml.put( GeneratedValue.class,
                     "generated-value" );
annotationToXml.put( Column.class,
                     "column" );
annotationToXml.put( Columns.class,
                     "column" );
annotationToXml.put( Temporal.class,
```

```
annotationToXml.put( JoinColumn.class,
                     "join-column" );
annotationToXml.put( JoinColumns.class,
                     "join-column" );
annotationToXml.put( MapKey.class,
                     "map-key" );
annotationToXml.put( OrderBy.class,
                     "order-by" );
annotationToXml.put( EntityListeners.class,
                     "entity-listeners" );
annotationToXml.put( PrePersist.class,
                     "pre-persist" );
annotationToXml.put( PreRemove.class,
                     "pre-remove" );
annotationToXml.put( PreUpdate.class,
                     "pre-update" );
annotationToXml.put( PostPersist.class,
                     "post-persist" );
annotationToXml.put( PostRemove.class,
                     "post-remove" );
annotationToXml.put( PostUpdate.class,
                     "post-update" );
annotationToXml.put( PostLoad.class,
                     "post-load" );
annotationToXml.put( CollectionTable.class,
                     "collection-table" );
annotationToXml.put( MapKeyClass.class,
```

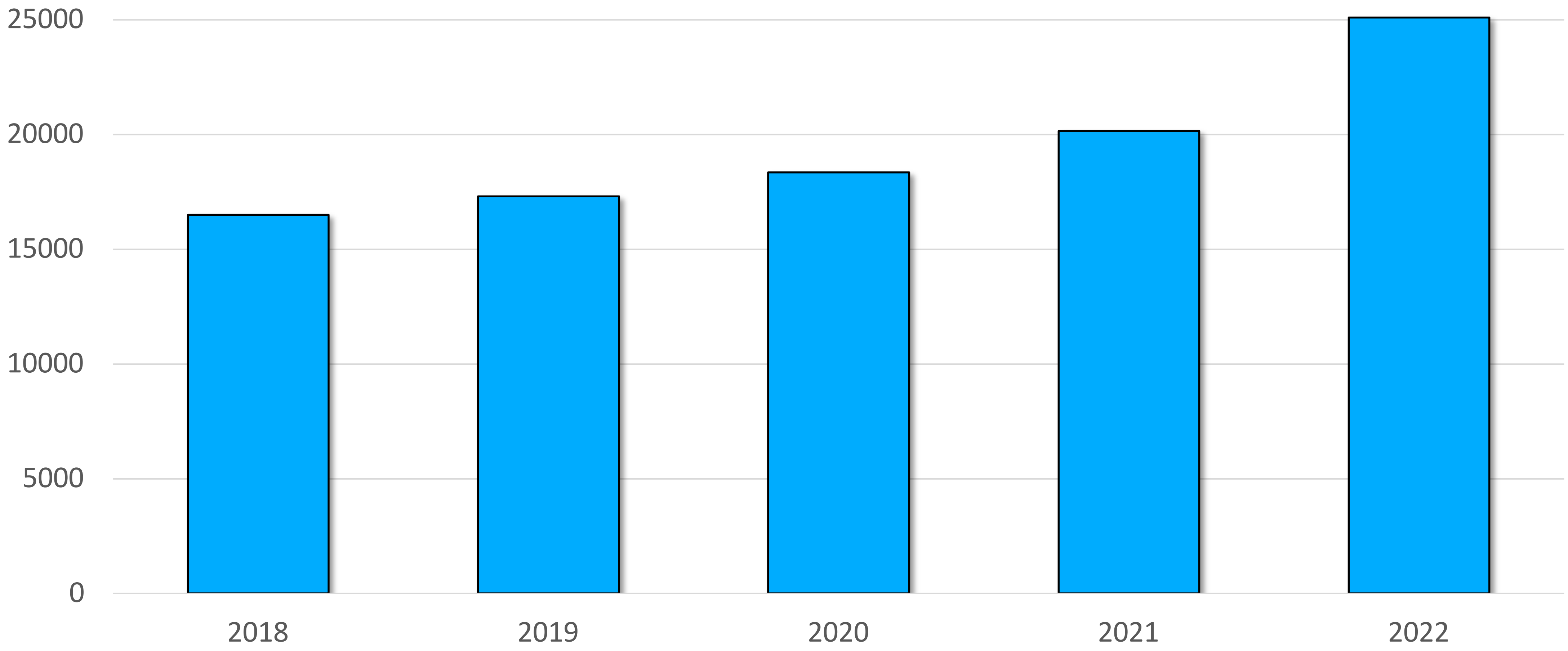
Duplicate Map key

V6033 An item with the same key 'AttributeOverrides.class' has already been added.

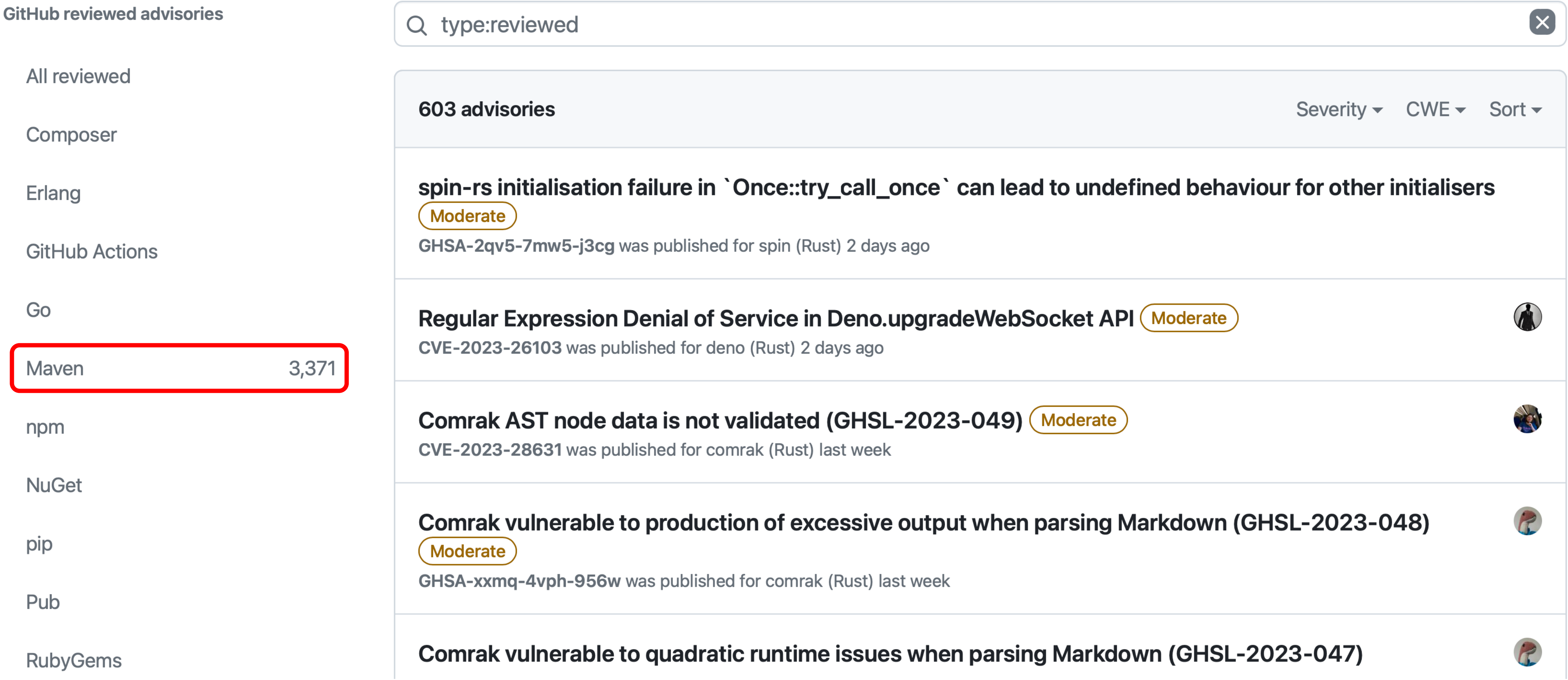
# Стоимость исправления уязвимостей



# Количество уязвимостей по годам



# GitHub Advisory Database



GitHub reviewed advisories

- All reviewed
- Composer
- Erlang
- GitHub Actions
- Go
- Maven 3,371
- npm
- NuGet
- pip
- Pub
- RubyGems

type:reviewed

**603 advisories**

**spin-rs initialisation failure in `Once::try_call_once` can lead to undefined behaviour for other initialisers** Moderate
GHSA-2qv5-7mw5-j3cg was published for spin (Rust) 2 days ago

**Regular Expression Denial of Service in Deno.upgradeWebSocket API** Moderate
CVE-2023-26103 was published for deno (Rust) 2 days ago

**Comrak AST node data is not validated (GHSL-2023-049)** Moderate
CVE-2023-28631 was published for comrak (Rust) last week

**Comrak vulnerable to production of excessive output when parsing Markdown (GHSL-2023-048)** Moderate
GHSA-xxmq-4vph-956w was published for comrak (Rust) last week

**Comrak vulnerable to quadratic runtime issues when parsing Markdown (GHSL-2023-047)**

# Статический анализ: как?

# Статический анализ: как?

Как анализаторы понимают код?

Какие есть виды анализа?

# Типы используемой информации

# Базовая информация для анализа

Синтаксис

Семантика

Аннотации

# Синтаксическая информация

# Токенизация

```
if (ioPending) {
  new
  MemoryAccessException(
    "Cyclic Access"
  );
}
```

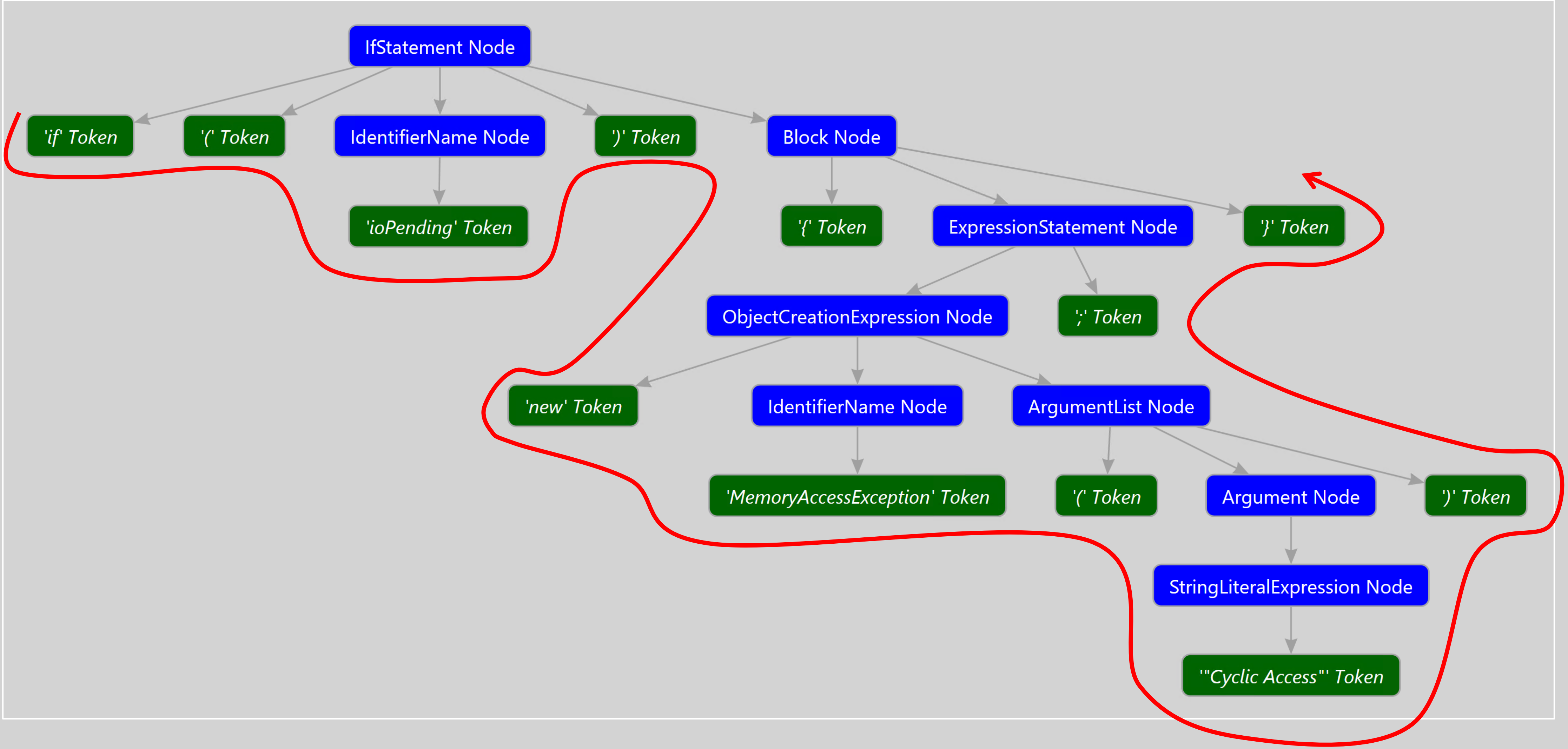
| **Лексема** | **Тип** |
|---|---|
| `if` | `If Keyword` |
| `(` | `Open Parenthesis` |
| `ioPending` | `Identifier` |
| `)` | `Close Parenthesis` |
| `{` | `Open Brace` |
| `new` | `New Keyword` |
| `MemoryAccessException` | `Identifier` |
| `(` | `Open Parenthesis` |
| `"Cyclic Access"` | `String Literal` |
| `)` | `Close Parenthesis` |
| `;` | `Semicolon` |
| `}` | `Close Brace` |

# Токенизация

```
if (ioPending) {
  new MemoryAccessException("Cyclic Access");
}

if (ioPending)
{
  new MemoryAccessException("Cyclic Access");
}

if (ioPending)
{ new MemoryAccessException("Cyclic Access"); }
```

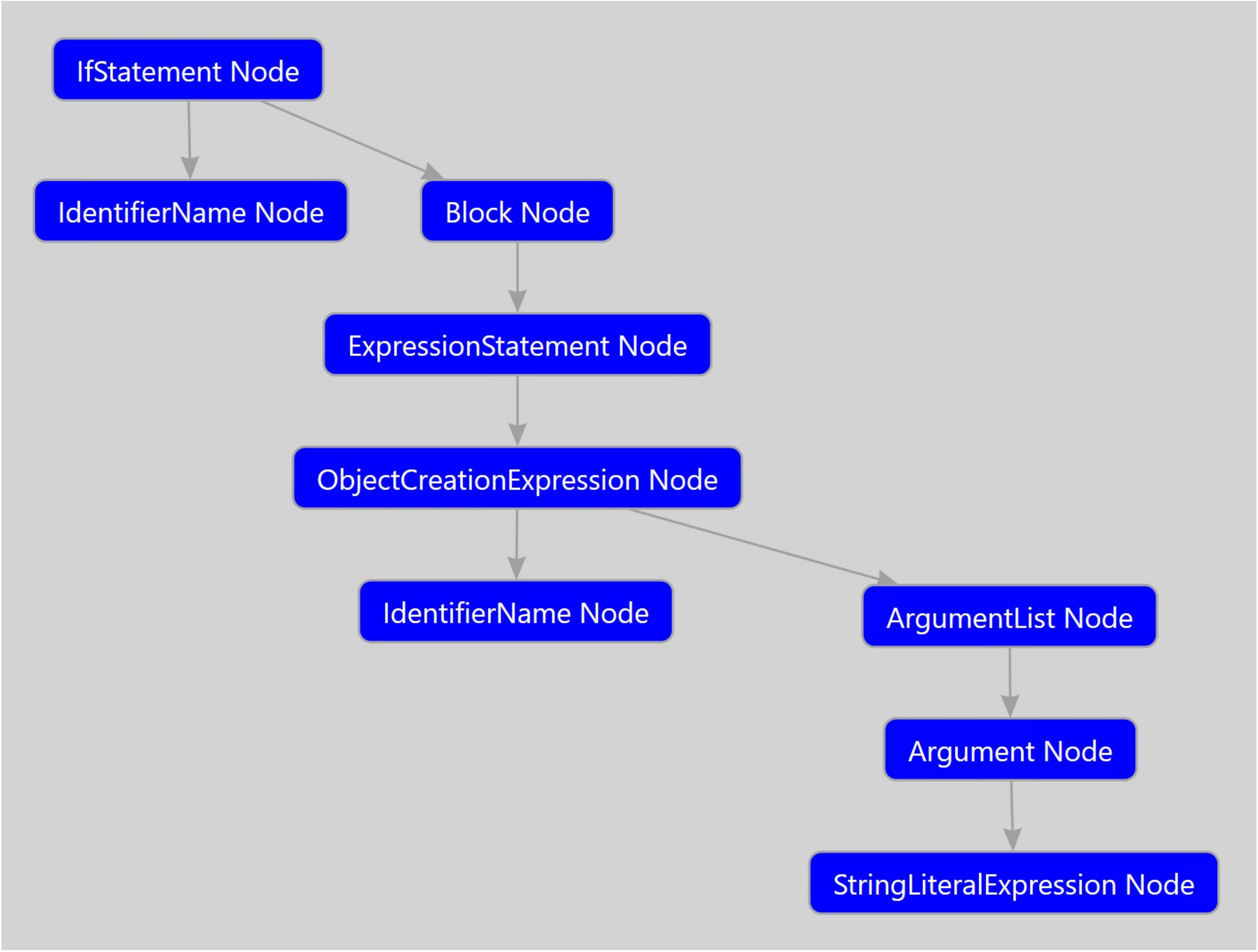
| **Лексема** | **Тип** |
|---|---|
| `if` | `If Keyword` |
| `(` | `Open Parenthesis` |
| `ioPending` | `Identifier` |
| `)` | `Close Parenthesis` |
| `{` | `Open Brace` |
| `new` | `New Keyword` |
| `MemoryAccessException` | `Identifier` |
| `(` | `Open Parenthesis` |
| `"Cyclic Access"` | `String Literal` |
| `)` | `Close Parenthesis` |
| `;` | `Semicolon` |
| `}` | `Close Brace` |

# Синтаксические деревья



# Синтаксические деревья

```
if (ioPending) {
  new
  MemoryAccessException(
    "Cyclic Access"
  );
}
```


IfStatement Node
IdentifierName Node
Block Node
ExpressionStatement Node
ObjectCreationExpression Node
IdentifierName Node
ArgumentList Node
Argument Node
StringLiteralExpression Node

# Пример: микс типов

```
private static boolean
checkSentenceCapitalization(@NotNull String value) {
    List<String> words = StringUtil.split(value, " ");
    ....
    // allow reasonable amount of capitalized words
    return capitalized / words.size() < 0.2;
}
```

# Пример: микс типов

```
private static boolean
checkSentenceCapitalization(@NotNull String value) {
    List<String> words = StringUtil.split(value, " ");
    ....
    // allow reasonable amount of capitalized words
    return capitalized / words.size() < 0.2;
}
```

| **Выражение** | **Значение** |
|---|---|
| `capitalized` | `3` |
| `words.size()` | `12` |
| `capitalized / words.size()` | `0` |
| `0 < 0.2` | `true` |

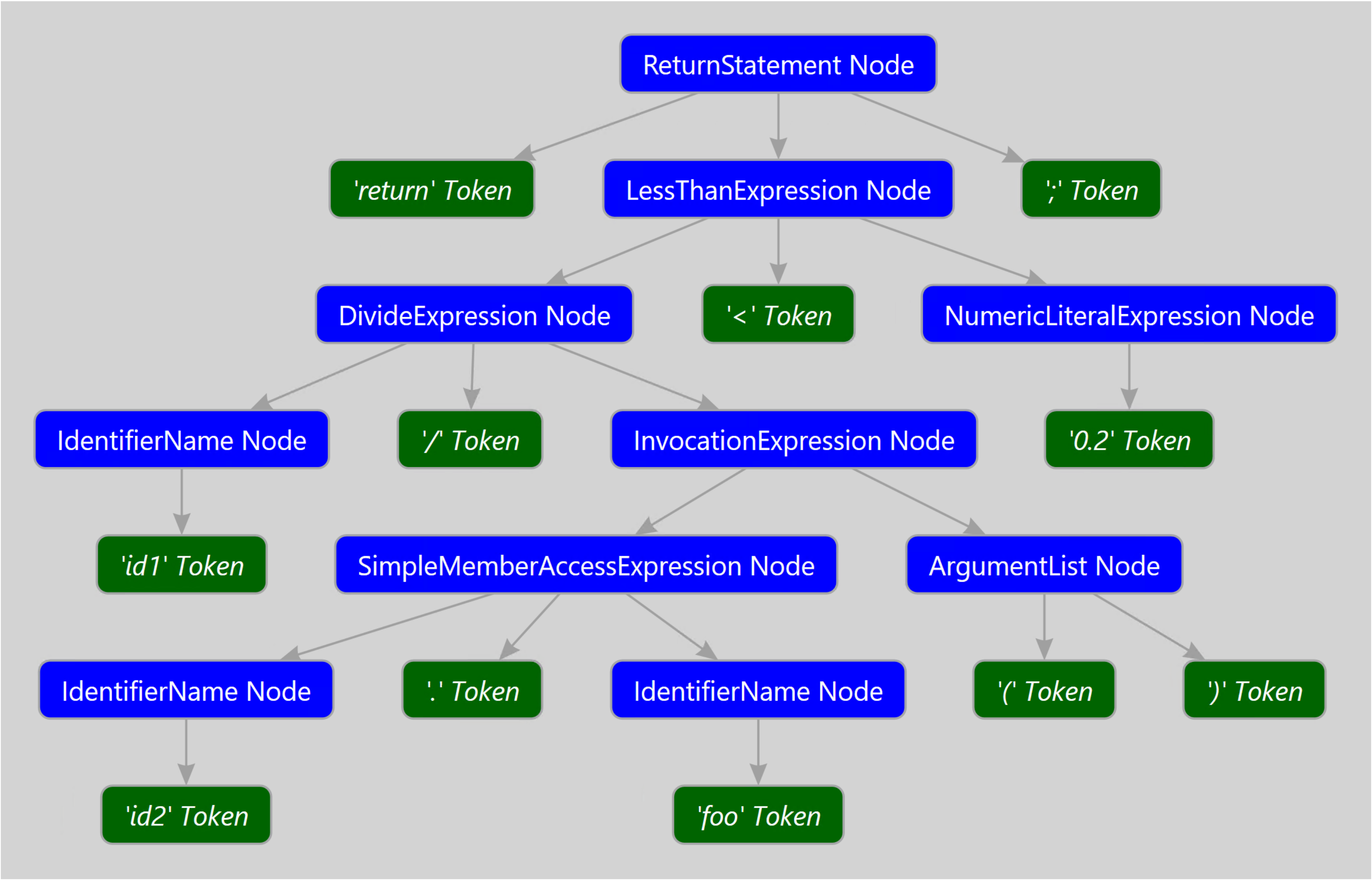
# Пример: микс типов

```
private static boolean
checkSentenceCapitalization(@NotNull String value) {
    List<String> words = StringUtil.split(value, " ");
    ....
    // allow reasonable amount of capitalized words
    return capitalized / words.size() < 0.2;
}
```

| **Выражение** | **Значение** |
|---|---|
| `capitalized` | `3` |
| `words.size()` | `12` |
| `(double)capitalized / words.size()` | `0.25` |
| `0.25 < 0.2` | `false` |

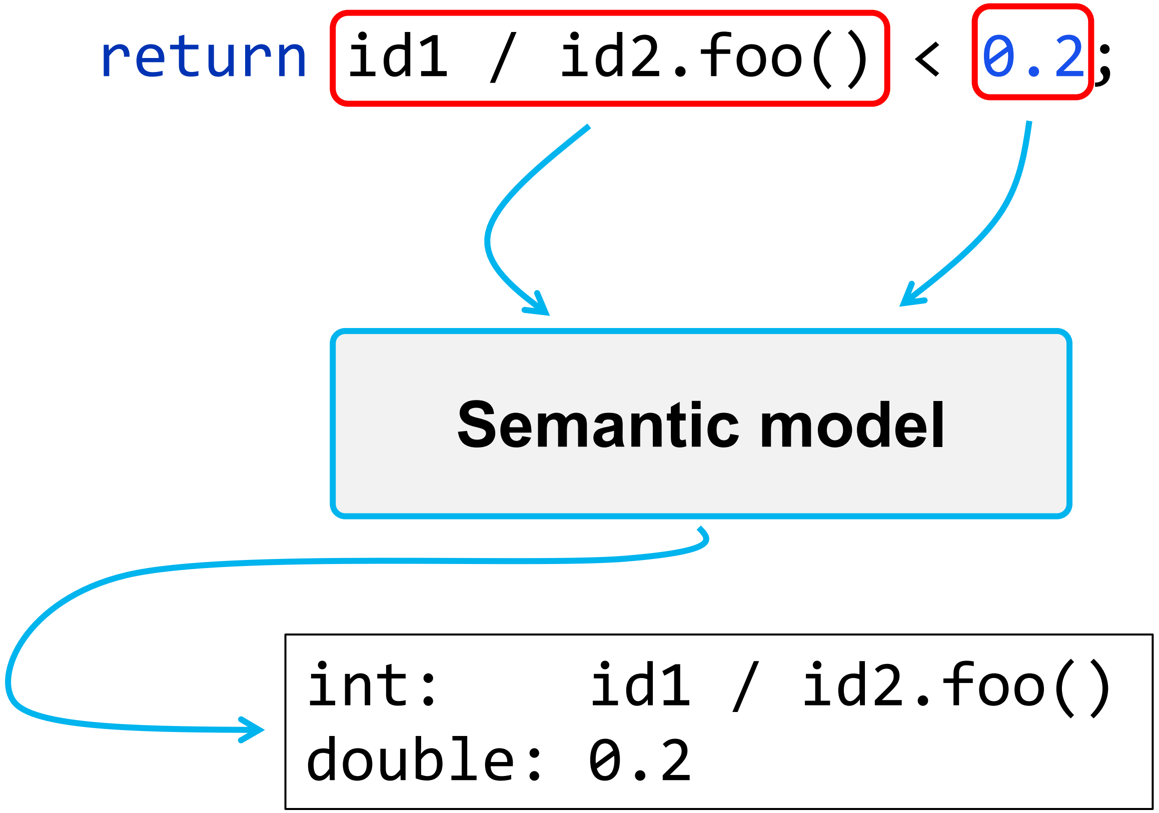
# Пример: микс типов

```
return capitalized / words.size() < 0.2;

return id1 / id2.foo() < 0.2;
return id1 / id2.foo() < 0.2;
```



# Семантическая информация

# Семантическая информация

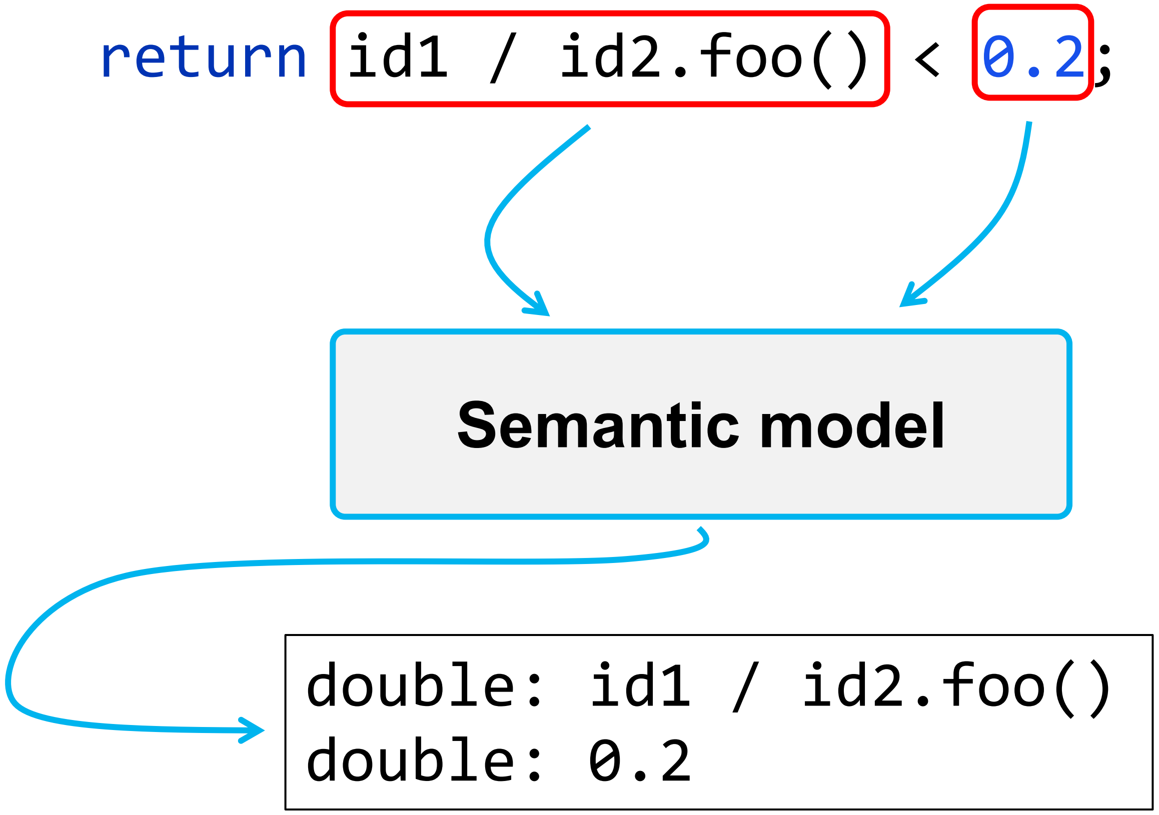


```
return id1 / id2.foo() < 0.2;
```

**Semantic model**

```
int:    id1 / id2.foo()
double: 0.2
```

# Семантическая информация



```
return id1 / id2.foo() < 0.2;
```

**Semantic model**

```
double: id1 / id2.foo()
double: 0.2
```

# Семантическая информация

- Информация о типах
- Сравнение сем. символов
- Определение сущностей
  (поля, локальные переменные и т. п.)

# Пример: potential NPE

```
public static void processFiles(String folderPath) {
  var folder = new File(folderPath);
  var files = folder.listFiles();

  for (int i = 0; i < files.length; ++i) {
    // process each file
  }
}

var folderPath = GetUserInput(); // "/Users/sergvasiliev/"
processFiles(folderPath);
```

# Пример: potential NPE

```
public static void processFiles(String folderPath) {
  var folder = new File(folderPath);
  var files = folder.listFiles(); // null

  for (int i = 0; i < files.length; ++i) { // NPE
    // process each file
  }
}

var folderPath = GetUserInput(); // "/Users/sergvasiliev/file"
processFiles(folderPath);
```

# Пример: potential NPR

```
void foo(File folder) {
  var files = folder.listFiles(); // potential null
  var filesCount = files.length; // potential NPE
}
```

# Пример: potential NPR

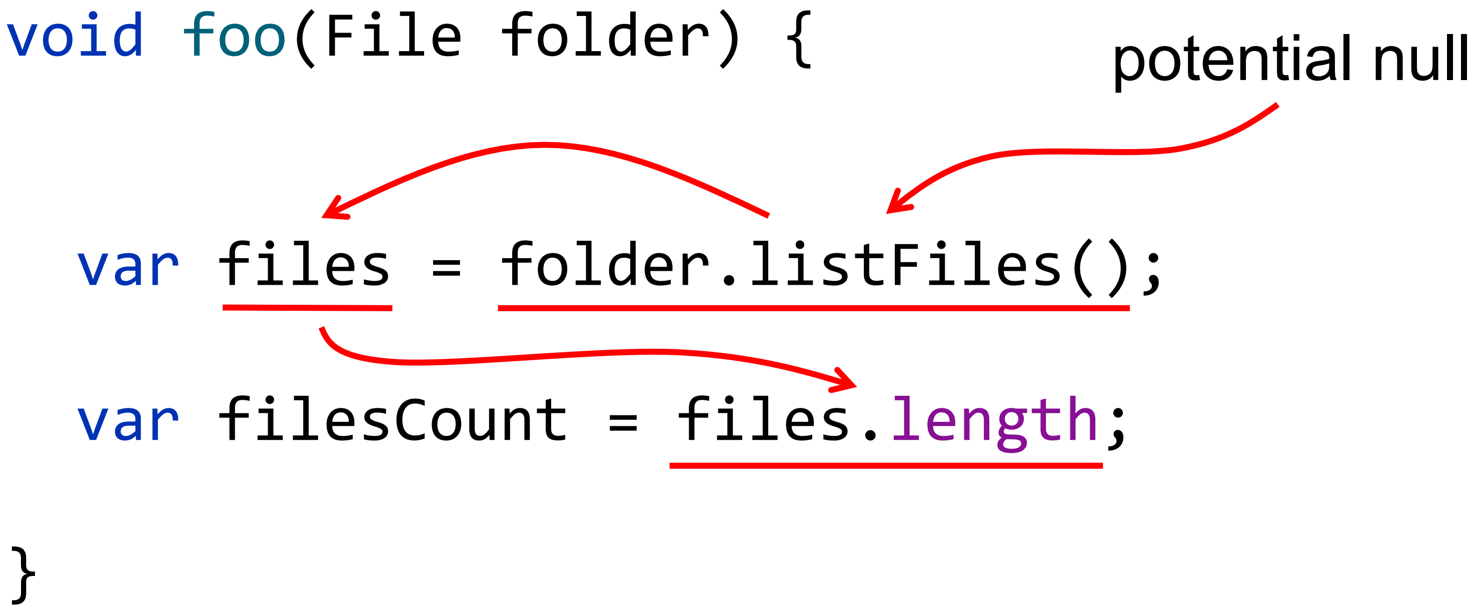
```
void foo(File folder) {

  var files = folder.listFiles();
  var filesCount = files.length;

}
```

potential null

# Аннотации анализатора

# Аннотации анализатора

```
Class("java.io.File")
....
- Function("listFiles")
   .Returns(Ptr(PotentialNullPointer))
....
```



```
void foo(File folder) {

  var files = folder.listFiles();

  var filesCount = files.length;

}
```

potential null

# Аннотации анализатора

```
public static String buildCanonicalHost(URL url) {
    String host = url.getHost();
    int port = url.getPort();

    if (port > -1) {
        host.concat(":" + Integer.toString(port));
    }

    return host;
}
```

# Аннотации анализатора

```
Class("java.lang.String",
         TypeClassification::String
       | TypeClassification::Immutable
       | TypeClassification::SuspiciousEqualsOperator)
....
 - Function("concat", Type::Pointer)
         .Pure()
         .Set(FunctionClassification::RetUse)
         .Returns(Ptr(NotNullPointer))
....
```

# Разновидности анализа

# Разновидности анализа

**Pattern-based analysis**

**Data flow analysis**

**Taint analysis**

# Pattern-based analysis

# Pattern-based analysis

Ищем ошибочные закономерности

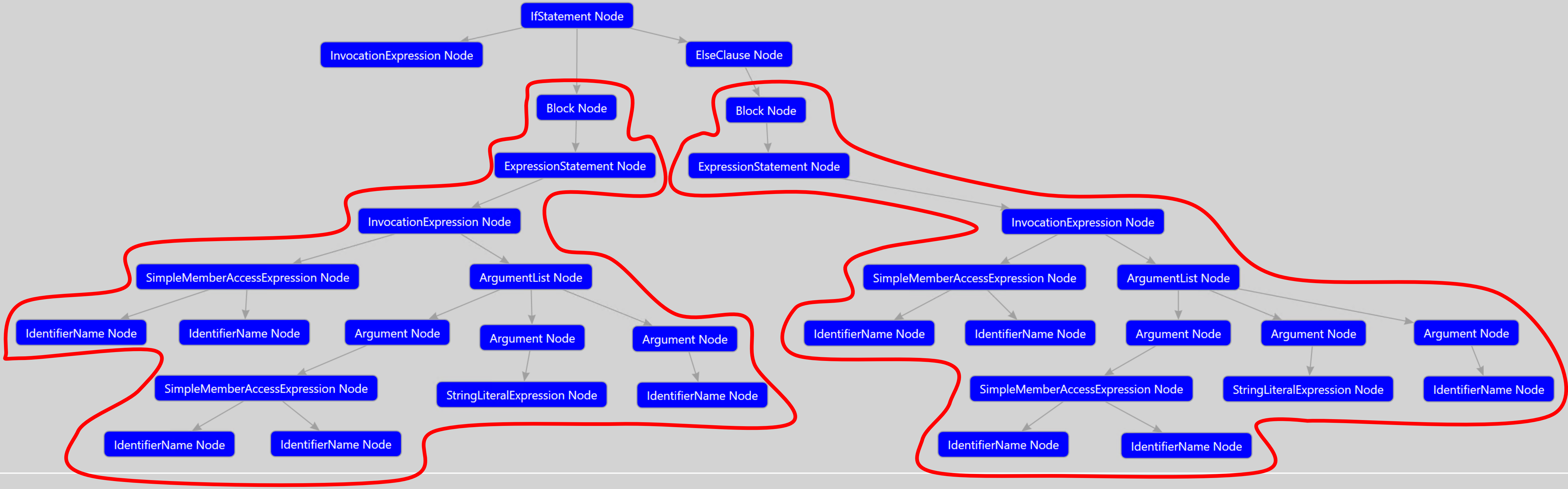
Используем:

- синтаксис
- семантика
- аннотации

# Pattern-based analysis: jMonkeyEngine

```
if (p.isConnected()) {
    log.log(Level.FINE, "Connection closed:{0}.", p);
}
else {
    log.log(Level.FINE, "Connection closed:{0}.", p);
}
```

V6004 The 'then' statement is equivalent to the 'else' statement.

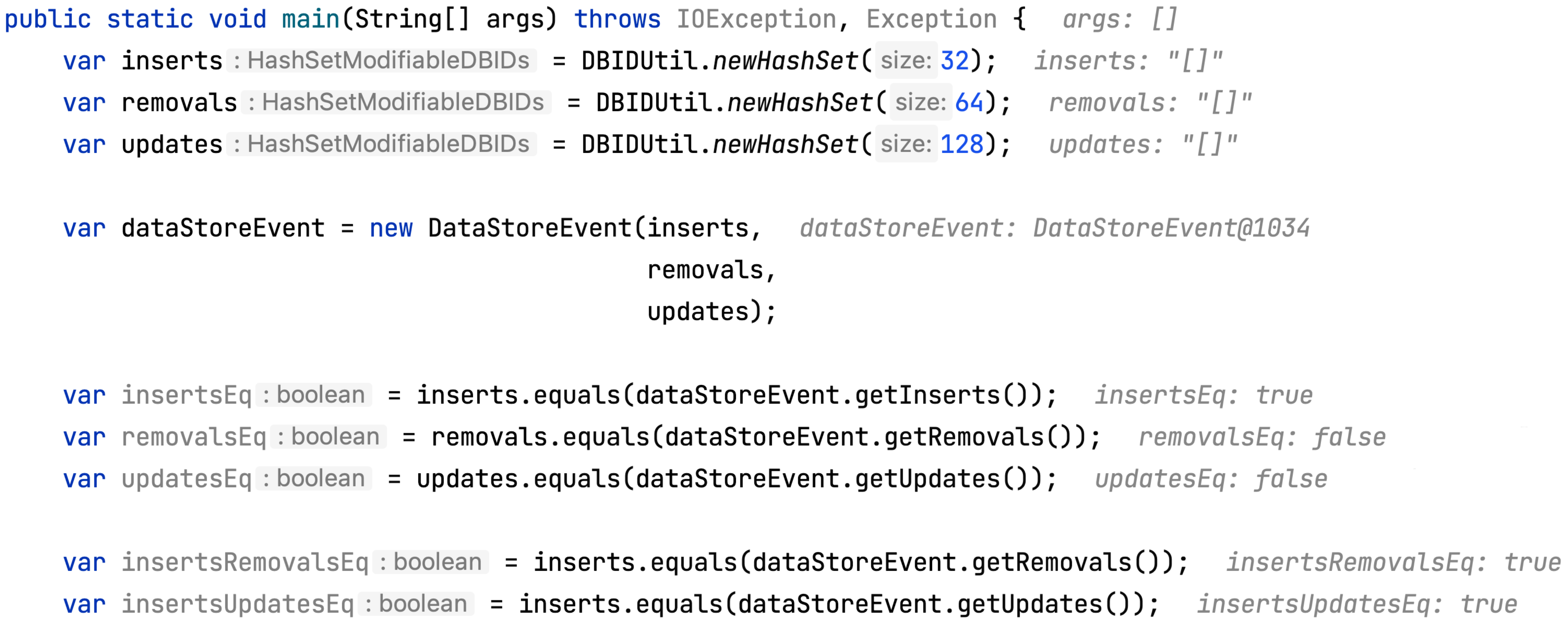

IfStatement Node
InvocationExpression Node
ElseClause Node
Block Node
Block Node
ExpressionStatement Node
ExpressionStatement Node
InvocationExpression Node
InvocationExpression Node
SimpleMemberAccessExpression Node
ArgumentList Node
SimpleMemberAccessExpression Node
ArgumentList Node
IdentifierName Node
IdentifierName Node
Argument Node
Argument Node
Argument Node
IdentifierName Node
IdentifierName Node
Argument Node
Argument Node
Argument Node
SimpleMemberAccessExpression Node
StringLiteralExpression Node
IdentifierName Node
SimpleMemberAccessExpression Node
StringLiteralExpression Node
IdentifierName Node
IdentifierName Node
IdentifierName Node
IdentifierName Node
IdentifierName Node

# ELKI

https://elki-project.github.io

```
<!-- https://mvnrepository.com/artifact/io.github.elki-project/elki -->
<dependency>
  <groupId>io.github.elki-project</groupId>
  <artifactId>elki</artifactId>
  <version>0.8.0</version>
</dependency>
```

# Пример: ELKI



```
public static void main(String[] args) throws IOException, Exception {   args: []
    var inserts : HashSetModifiableDBIDs = DBIDUtil.newHashSet( size: 32);   inserts: "[]"
    var removals : HashSetModifiableDBIDs = DBIDUtil.newHashSet( size: 64);   removals: "[]"
    var updates : HashSetModifiableDBIDs = DBIDUtil.newHashSet( size: 128);   updates: "[]"

    var dataStoreEvent = new DataStoreEvent(inserts,   dataStoreEvent: DataStoreEvent@1034
                                            removals,
                                            updates);

    var insertsEq : boolean = inserts.equals(dataStoreEvent.getInserts());   insertsEq: true
    var removalsEq : boolean = removals.equals(dataStoreEvent.getRemovals());   removalsEq: false
    var updatesEq : boolean = updates.equals(dataStoreEvent.getUpdates());   updatesEq: false

    var insertsRemovalsEq : boolean = inserts.equals(dataStoreEvent.getRemovals());   insertsRemovalsEq: true
    var insertsUpdatesEq : boolean = inserts.equals(dataStoreEvent.getUpdates());   insertsUpdatesEq: true
```

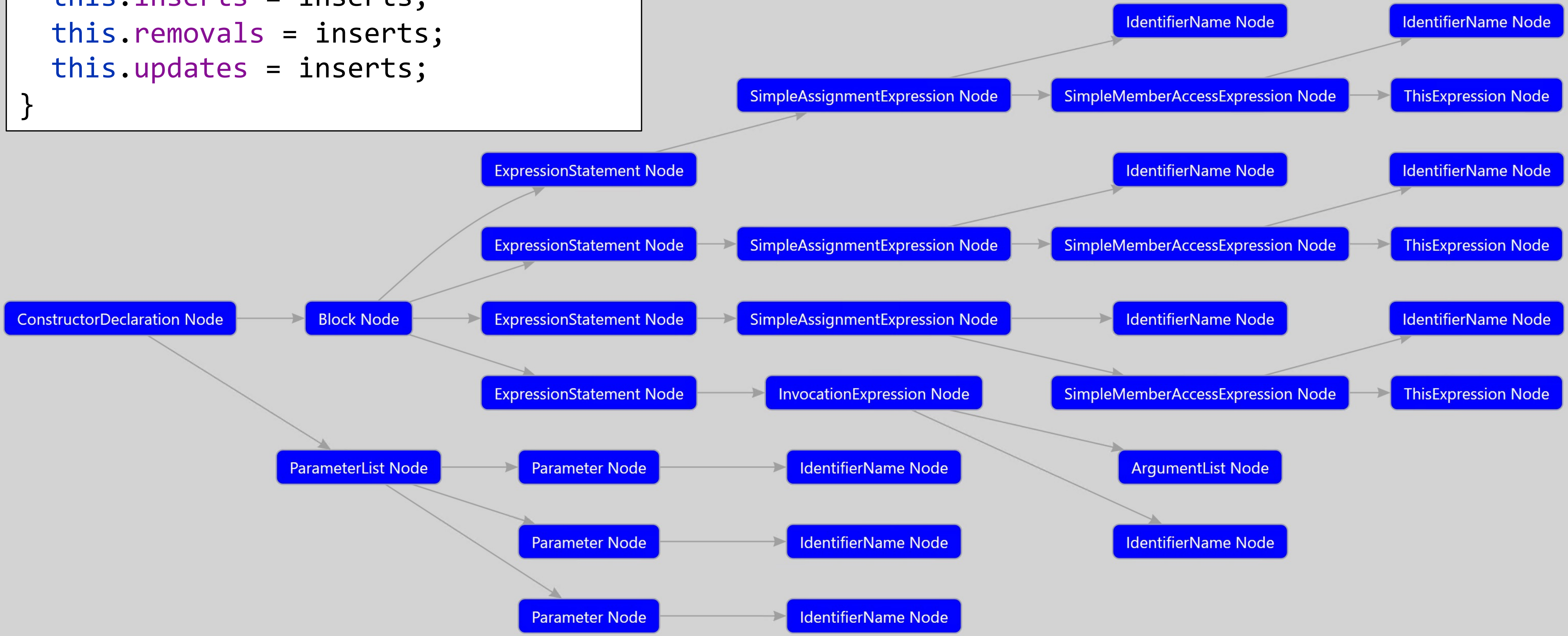
# Pattern-based analysis: ELKI

```
public DataStoreEvent(DBIDs inserts,
                      DBIDs removals,
                      DBIDs updates) {
  super();
  this.inserts = inserts;
  this.removals = inserts;
  this.updates = inserts;
}
```

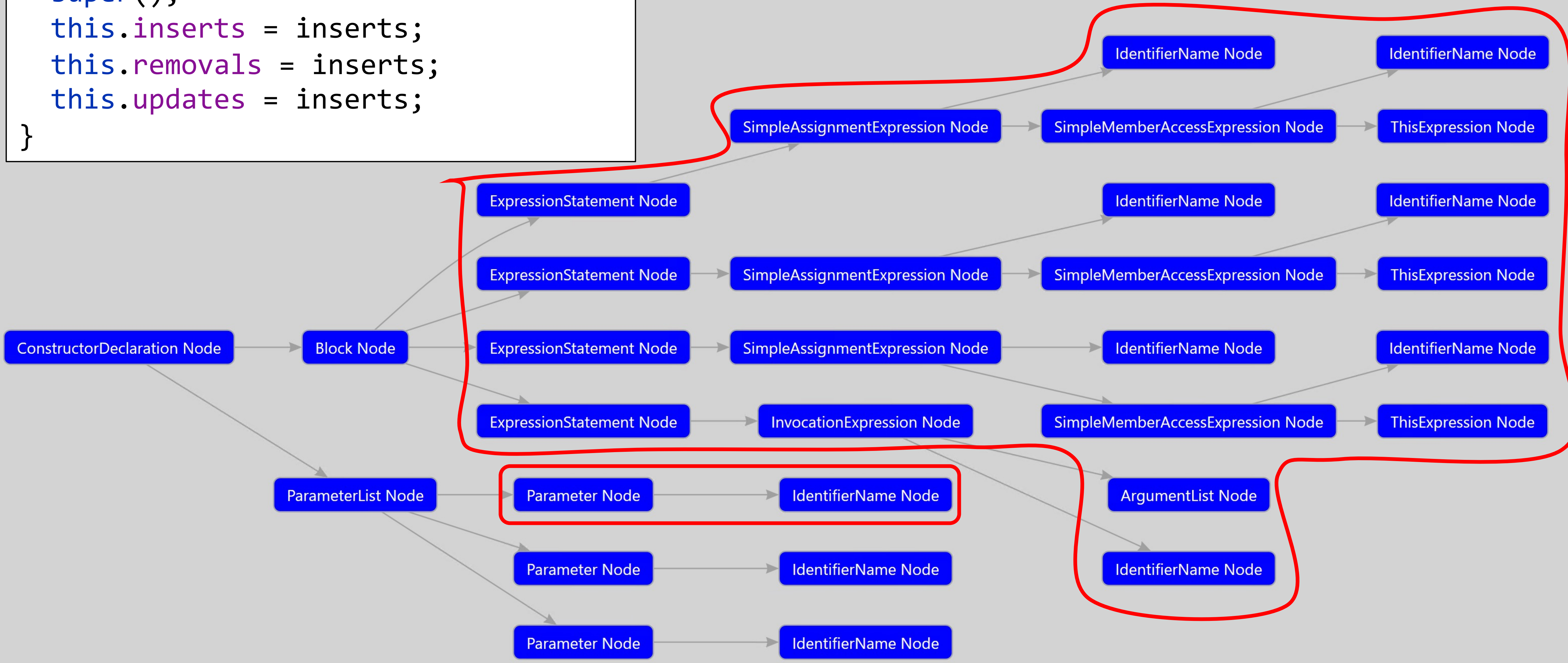
Parameter 'removals' is never used
Parameter 'updates' is never used

V6022 Parameter 'removals’
is not used inside constructor body.
V6022 Parameter 'updates'
is not used inside constructor body.

```
public DataStoreEvent(DBIDs inserts,
                      DBIDs removals,
                      DBIDs updates)
{
  super();
  this.inserts = inserts;
  this.removals = inserts;
  this.updates = inserts;
}
```

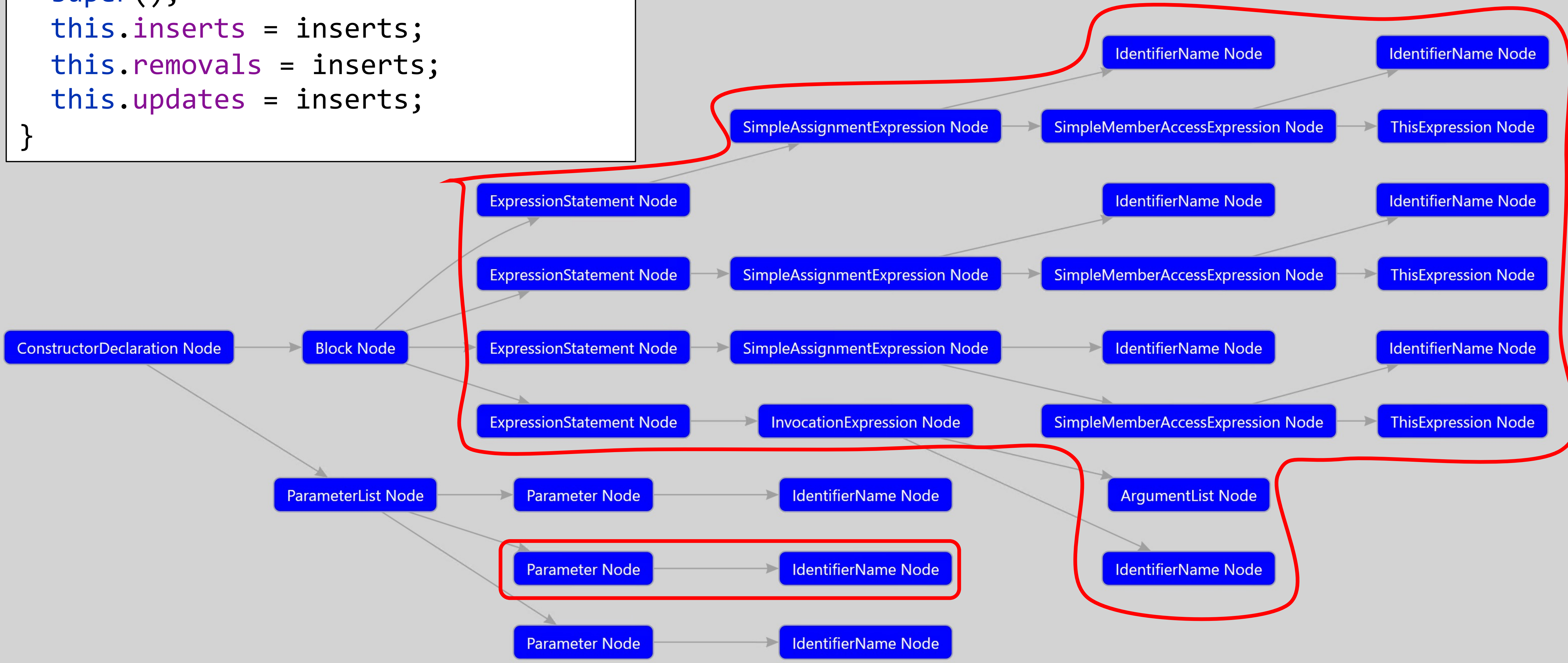


```
public DataStoreEvent(DBIDs inserts,
                      DBIDs removals,
                      DBIDs updates)
{
  super();
  this.inserts = inserts;
  this.removals = inserts;
  this.updates = inserts;
}
```

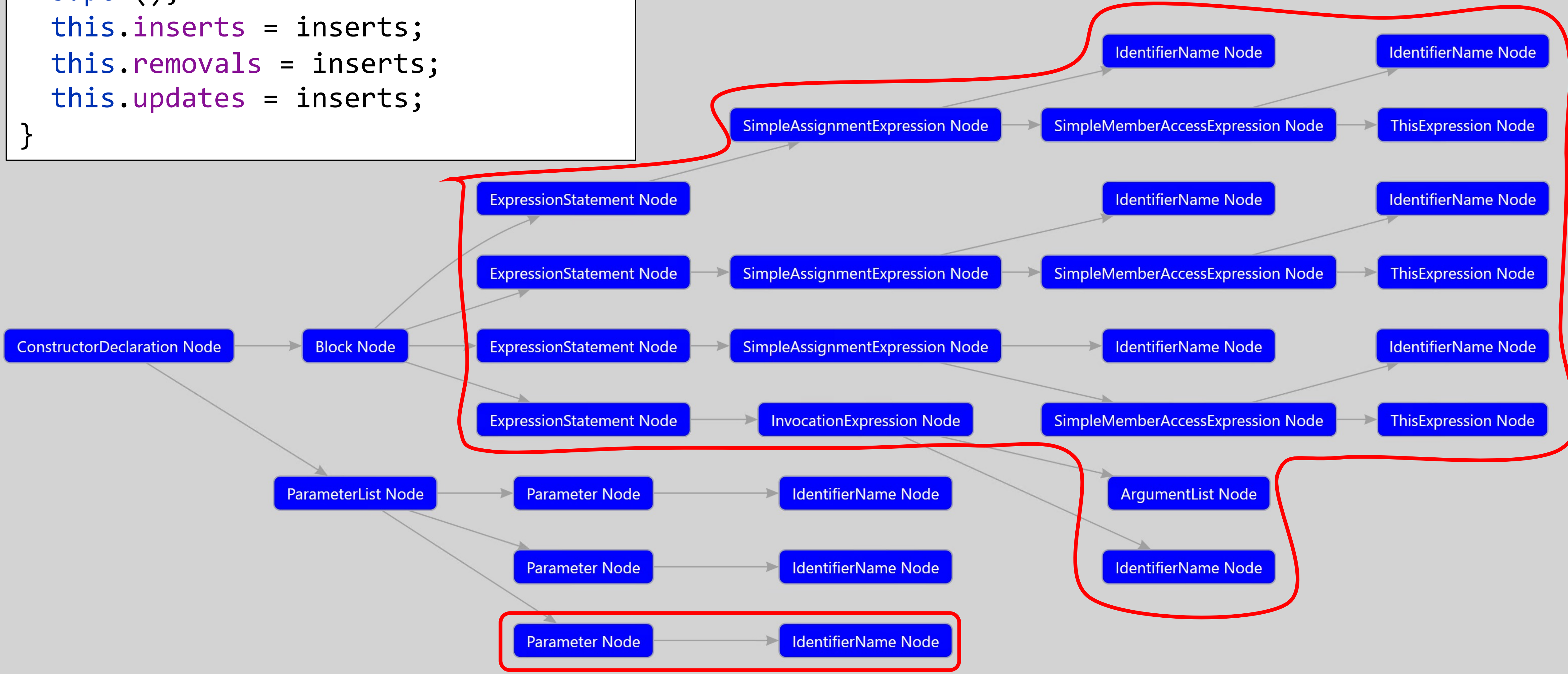


IdentifierName Node
IdentifierName Node
SimpleAssignmentExpression Node
SimpleMemberAccessExpression Node
ThisExpression Node
ExpressionStatement Node
IdentifierName Node
IdentifierName Node
ExpressionStatement Node
SimpleAssignmentExpression Node
SimpleMemberAccessExpression Node
ThisExpression Node
ConstructorDeclaration Node
Block Node
ExpressionStatement Node
SimpleAssignmentExpression Node
IdentifierName Node
IdentifierName Node
ExpressionStatement Node
InvocationExpression Node
SimpleMemberAccessExpression Node
ThisExpression Node
ParameterList Node
Parameter Node
IdentifierName Node
ArgumentList Node
Parameter Node
IdentifierName Node
IdentifierName Node
Parameter Node
IdentifierName Node

```
public DataStoreEvent(DBIDs inserts,
                      DBIDs removals,
                      DBIDs updates)
{
  super();
  this.inserts = inserts;
  this.removals = inserts;
  this.updates = inserts;
}
```



```
public DataStoreEvent(DBIDs inserts,
                      DBIDs removals,
                      DBIDs updates)
{
  super();
  this.inserts = inserts;
  this.removals = inserts;
  this.updates = inserts;
}
```

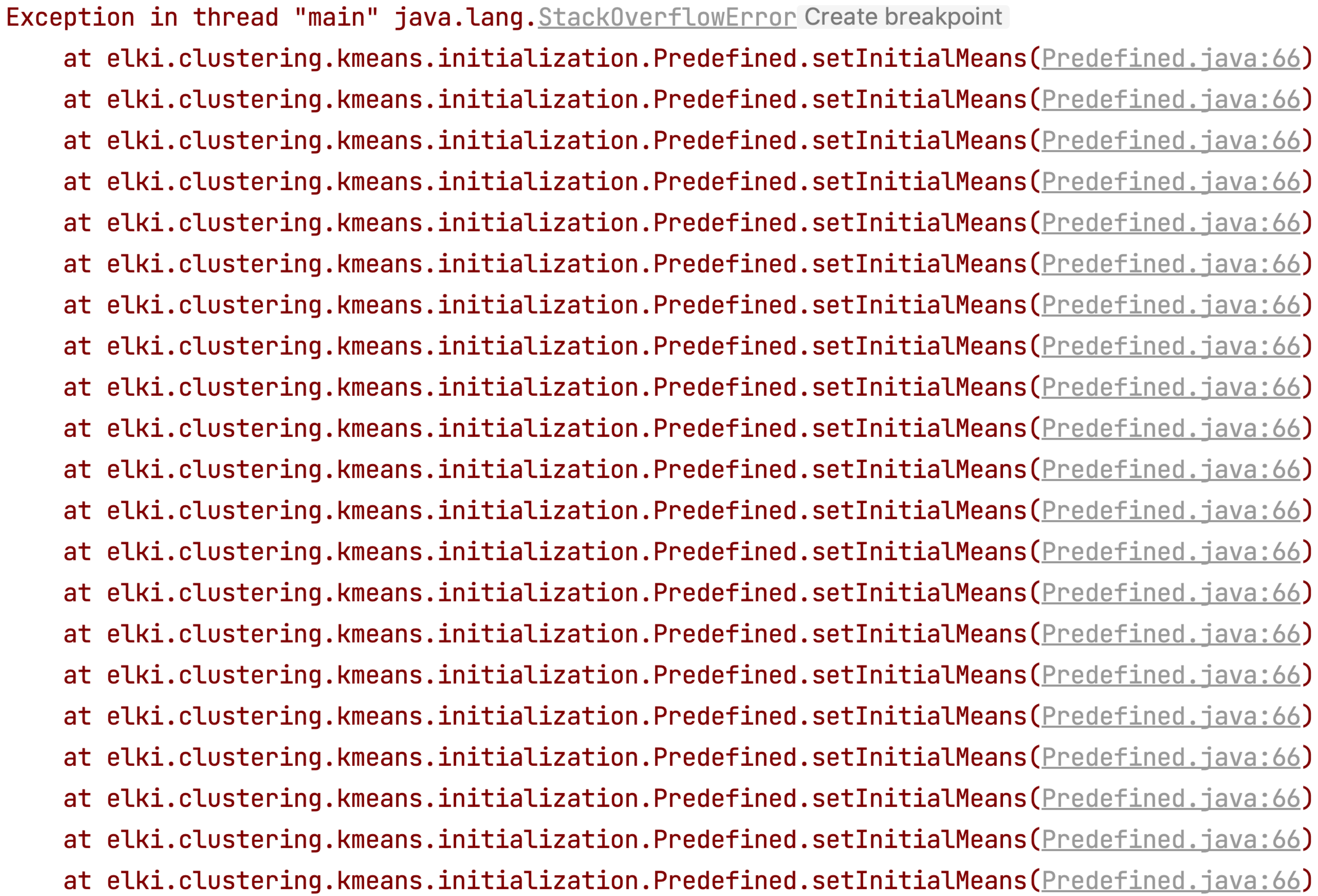


# Пример: ELKI

```java
public class Predefined extends AbstractKMeansInitialization {
    double[][] initialMeans;

    public Predefined(double[][] initialMeans) {
        super((RandomFactory) null);
        this.initialMeans = initialMeans;
    }

    public void setInitialMeans(List<double[]> initialMeans) {
        this.setInitialMeans(initialMeans);
    }

    public void setInitialMeans(double[][] initialMeans) {
        double[][] vecs = initialMeans.clone();
        this.initialMeans = vecs;
    }
    ....
}
```

# Пример: ELKI

```
public static void main(String[] args) throws .... {

    var predefined = new Predefined(new double[3][3]);

    var newInitialMeans = new ArrayList<double[]>();
    predefined.setInitialMeans(newInitialMeans);
}
```



```
Exception in thread "main" java.lang.StackOverflowError
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
	at elki.clustering.kmeans.initialization.Predefined.setInitialMeans(Predefined.java:66)
```

# Пример: ELKI

```
public static void main(String[] args) throws .... {

    var predefined = new Predefined(new double[3][3]);

    var newInitialMeans = new ArrayList<double[]>();
    predefined.setInitialMeans(newInitialMeans);
}
```

# Pattern-based analysis: ELKI

```java
public class Predefined extends AbstractKMeansInitialization {
    double[][] initialMeans;

    public Predefined(double[][] initialMeans) {
        super((RandomFactory) null);
        this.initialMeans = initialMeans;
    }

    public void setInitialMeans(List<double[]> initialMeans) {
        this.setInitialMeans(initialMeans);
    }

    public void setInitialMeans(double[][] initialMeans) {
        double[][] vecs = initialMeans.clone();
        this.initialMeans = vecs;
    }
    ....
}
```

# Pattern-based analysis: ELKI

```
public class Predefined extends AbstractKMeansInitialization {
    ....

    public void setInitialMeans(List<double[]> initialMeans) {
        this.setInitialMeans(initialMeans);
    }
    ....
}
```

Method 'setInitialMeans()' recurses
infinitely, and can only end
by throwing an exception

V6062 Possible infinite recursion
inside the 'setInitialMeans' method.

# Пример: potential NPE

```
void ProcessUserName(String userName) {
    if (userName == null && userName.length() == 0) {
        //  Show error
    }

    // Process
}
```

# Пример: potential NPE

```
static void ProcessConfiguration(Configuration configuration) {
    if (configuration == null) {
       ....
       logger.logError("Unknown configuration: {0}",
                       configuration.getName())
       ....
    }

    // Process configuration
}
```

# Data flow analysis

# Data flow analysis

Отслеживаем значения выражений

Храним доп. информацию о выражениях (аннотации)

Pointers / references: null state analysis

Booleans: true / false

Integers: ranges

# Data flow analysis

```
void DataFlowTest(int x) {
  if (x > 10) {
    var y = x - 10;
    if (y < 0) {
      //....
    }

    if (y <= 1) {
      //....
    }
  }
}
```

# Data flow analysis

```
void DataFlowTest(int x) {
  if (x > 10) {
    var y = x - 10;
    if (y < 0) {
      //....
    }

    if (y <= 1) {
      //....
    }
  }
}
```

# Data flow analysis

```
void DataFlowTest(int x) { [-2147483648; 2147483647]
  if (x > 10) {
    var y = x - 10;
    if (y < 0) {
      //....
    }

    if (y <= 1) {
      //....
    }
  }
}
```

# Data flow analysis

```
void DataFlowTest(int x) { [-2147483648; 2147483647]
  if (x > 10) {
    var y = x - 10;
    if (y < 0) {
      //....
    }

    if (y <= 1) {
      //....
    }
  }
}
```

# Data flow analysis

```
void DataFlowTest(int x) { [-2147483648; 2147483647]
  if (x > 10) {            [11; 2147483647]
    var y = x - 10;
    if (y < 0) {
      //....
    }

    if (y <= 1) {
      //....
    }
  }
}
```

# Data flow analysis

```
void DataFlowTest(int x) {
  if (x > 10) {                 [11; 2147483647]
    var y = x - 10;
    if (y < 0) {
      //....
    }

    if (y <= 1) {
      //....
    }
  }
}
```

# Data flow analysis

```
void DataFlowTest(int x) {
  if (x > 10) {                [11; 2147483647]
    var y = x - 10;            [1; 2147483637]
    if (y < 0) {
      //....
    }

    if (y <= 1) {
      //....
    }
  }
}
```

# Data flow analysis

```
void DataFlowTest(int x) {
  if (x > 10) {
    var y = x - 10;
    if (y < 0) {
      //....
    }

    if (y <= 1) {
      //....
    }
  }
}
```

[1; 2147483637]

# Data flow analysis

```
void DataFlowTest(int x) {
  if (x > 10) {
    var y = x - 10;
    if (y < 0) {
      //....
    }

    if (y <= 1) {
      //....
    }
  }
}
```

[1; 2147483637]
[-2147483648; -1]

Condition is always false

# Data flow analysis

```
void DataFlowTest(int x) {
  if (x > 10) {
    var y = x - 10;
    if (y < 0) {
      //....
    }

    if (y <= 1) {
      //....
    }
  }
}
```

[1; 2147483637]

[-2147483648; 1]

OK: [1]

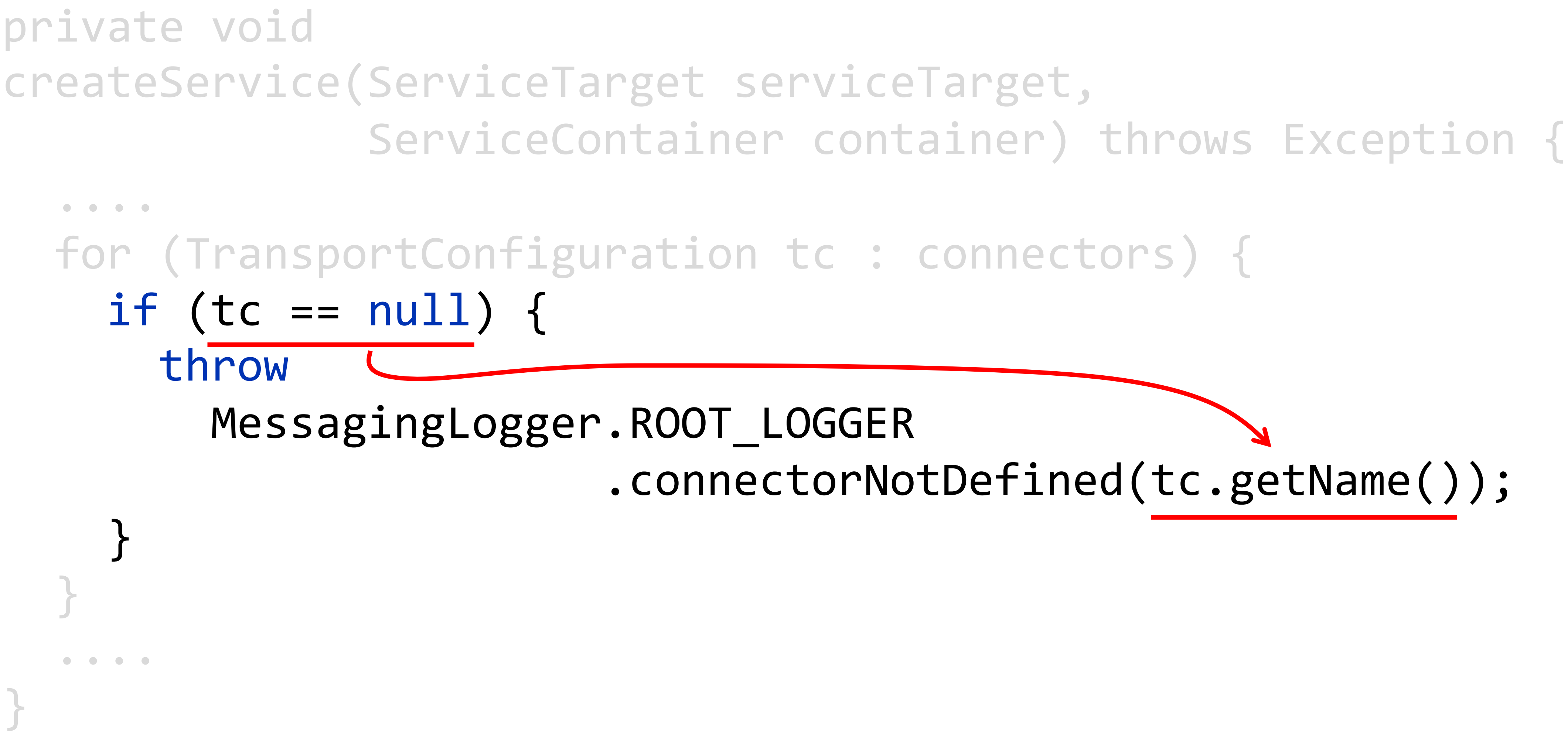
# Пример: WildFly

```
private void
createService(ServiceTarget serviceTarget,
              ServiceContainer container) throws Exception {
  ....
  for (TransportConfiguration tc : connectors) {
    if (tc == null) {
      throw
        MessagingLogger.ROOT_LOGGER
                       .connectorNotDefined(tc.getName());
    }
  }
  ....
}
```

# Пример: WildFly

```
private void
createService(ServiceTarget serviceTarget,
              ServiceContainer container) throws Exception {

  ....
  for (TransportConfiguration tc : connectors) {
    if (tc == null) {
      throw
        MessagingLogger.ROOT_LOGGER
                       .connectorNotDefined(tc.getName());
    }
  }
  ....
}
```

# Data flow analysis: WildFly



```
private void
createService(ServiceTarget serviceTarget,
              ServiceContainer container) throws Exception {
  ....
  for (TransportConfiguration tc : connectors) {
    if (tc == null) {
      throw
        MessagingLogger.ROOT_LOGGER
                       .connectorNotDefined(tc.getName());
    }
  }
  ....
}
```

# Data flow analysis: WildFly

```
if (tc == null) {
  throw
    MessagingLogger.ROOT_LOGGER
                  .connectorNotDefined(tc.getName());
}
```

Method invocation 'getName'
will produce 'NullPointerException'

V6008 Null dereference of 'tc'.

# Пример: Ghidra

```
private static boolean isOK(long[] data) {
  for (int i = 0; i < data.length; i++) {
    if (
           (data[i] != 0)
        ||
           (data[i] != 0xFFFFFFFFL)
       )
    {
      return true;
    }
  }
  return false;
}
```

# Data flow analysis: Ghidra

```
private static boolean isOK(long[] data) {
  for (int i = 0; i < data.length; i++) {
    if (
           (data[i] != 0)
        ||
           (data[i] != 0xFFFFFFFFL)
       )
    {
      return true;
    }
  }
  return false;
}
```

# Data flow analysis: Ghidra

```
private static boolean isOK(long[] data) {
  for (int i = 0; i < data.length; i++) {
    if (
           (data[i] != 0)            [Long.MIN; -1][1; Long.MAX]
        ||
           (data[i] != 0xFFFFFFFFL)
       )
    {
      return true;
    }
  }
  return false;
}
```

# Data flow analysis: Ghidra

```
private static boolean isOK(long[] data) {
  for (int i = 0; i < data.length; i++) {
    if (
           (data[i] != 0)           [Long.MIN; -1][1; Long.MAX]
        ||                          [0]
           (data[i] != 0xFFFFFFFFL)
       )
    {
      return true;
    }
  }
  return false;
}
```

# Data flow analysis: Ghidra

```
private static boolean isOK(long[] data) {
  for (int i = 0; i < data.length; i++) {
    if (
           (data[i] != 0)                     [Long.MIN; -1][1; Long.MAX]
        ||                                    [0]
           (data[i] != 0xFFFFFFFFL)           [0] != [0xFFFFFFFFL]
       )
    {
      return true;
    }
  }
  return false;
}
```

Always true

# Data flow analysis: Ghidra

```
private static boolean isOK(long[] data) {
  for (int i = 0; i < data.length; i++) {
    if (
           (data[i] != 0)                      [Long.MIN; -1][1; Long.MAX]
        ||                                     [0]
           (data[i] != 0xFFFFFFFFL) [0] != [0xFFFFFFFFL]
       )
    {
      return
    }
  }
  return fals
}
```

Always true

Condition '(data[i] != 0) || (data[i] != 0xFFFFFFFFL)' is always 'true'

Condition 'data[i] != 0xFFFFFFFFL' is always 'true' when reached

V6007 Expression 'data[i] != 0xFFFFFFFFL' is always true.

# Data flow analysis

```
public void ShellChromeGallery(....)
{
    if (appShell == null) {
        flyoutBehavior.setSelectedIndex(1);
        flyoutHeaderBehavior.setSelectedIndex(0);
    } else {
        flyoutBehavior.setSelectedIndex(....);
        flyoutHeaderBehavior.setSelectedIndex(....);
    }

    appShell.setFlyoutBackdrop(SolidColorBrush.Pink);
}
```

# Data flow analysis

```
public void ShellChromeGallery(....)
{
    if (appShell == null) { appShell -> null
        flyoutBehavior.setSelectedIndex(1);
        flyoutHeaderBehavior.setSelectedIndex(0);
    } else {                appShell -> not null
        flyoutBehavior.setSelectedIndex(....);
        flyoutHeaderBehavior.setSelectedIndex(....);
    }
    // appShell -> Unknown [null checked]
    appShell.setFlyoutBackdrop(SolidColorBrush.Pink);
}
```

# SQL injection

```
processUserData(HttpServletRequest req, Connection con) .... {
  var email = req.getParameter("email");
  var password = req.getParameter("password");

  String sqlQuery =   "SELECT * FROM users "
                    + "WHERE email='" + email + "' "
                    + "AND password='" + password + "'";

  Statement statement = con.createStatement();
  ResultSet rs = statement.executeQuery(sqlQuery);
  return rs.next();
}
```

# SQL injection

```
SELECT * FROM users
WHERE email='{0}'
AND password='{1}'

email:    jpoint23@gmail.com
password: lolStr0ngPWD

SELECT * FROM users
WHERE email='jpoint23@gmail.com'
AND password='lolStr0ngPWD'
```

# SQL injection

```
SELECT * FROM users
WHERE email='{0}'
AND password='{1}'

email:     ' OR '1'='1
password: lolNoMatter' OR '1'='1

SELECT * FROM users
WHERE email='' OR '1'='1'
AND password='lolNoMatter' OR '1'='1'
```

# Taint analysis

# Taint analysis

- Используется в SAST-решениях (static application security testing)

- Основан на трассах передачи "размеченных" данных

- Подходит для ловли:
  - SQL injection
  - Path traversal
  - XXE
  - ....

# SQL injection

```
processUserData(HttpServletRequest req, Connection con) .... {

  var email = req.getParameter("email");

  var password = req.getParameter("password");

  String sqlQuery =   "SELECT * FROM users "
                    + "WHERE email='" + email + "' "
                    + "AND password='" + password + "'";

  Statement statement = con.createStatement();
  ResultSet rs = statement.executeQuery(sqlQuery);
  return rs.next();
}
```

# SQL injection

```
processUserData(HttpServletRequest req, Connection con) .... {

  var email = req.getParameter("email");

  var password = req.getParameter("password");

  String sqlQuery =   "SELECT * FROM users "
                    + "WHERE email='" + email + "' "
                    + "AND password='" + password + "'";

  Statement statement = con.createStatement();
  ResultSet rs = statement.executeQuery(sqlQuery);
  return rs.next();
}
```

# Taint analysis

**Источники**

Откуда данные приходят?

**Передатчики**

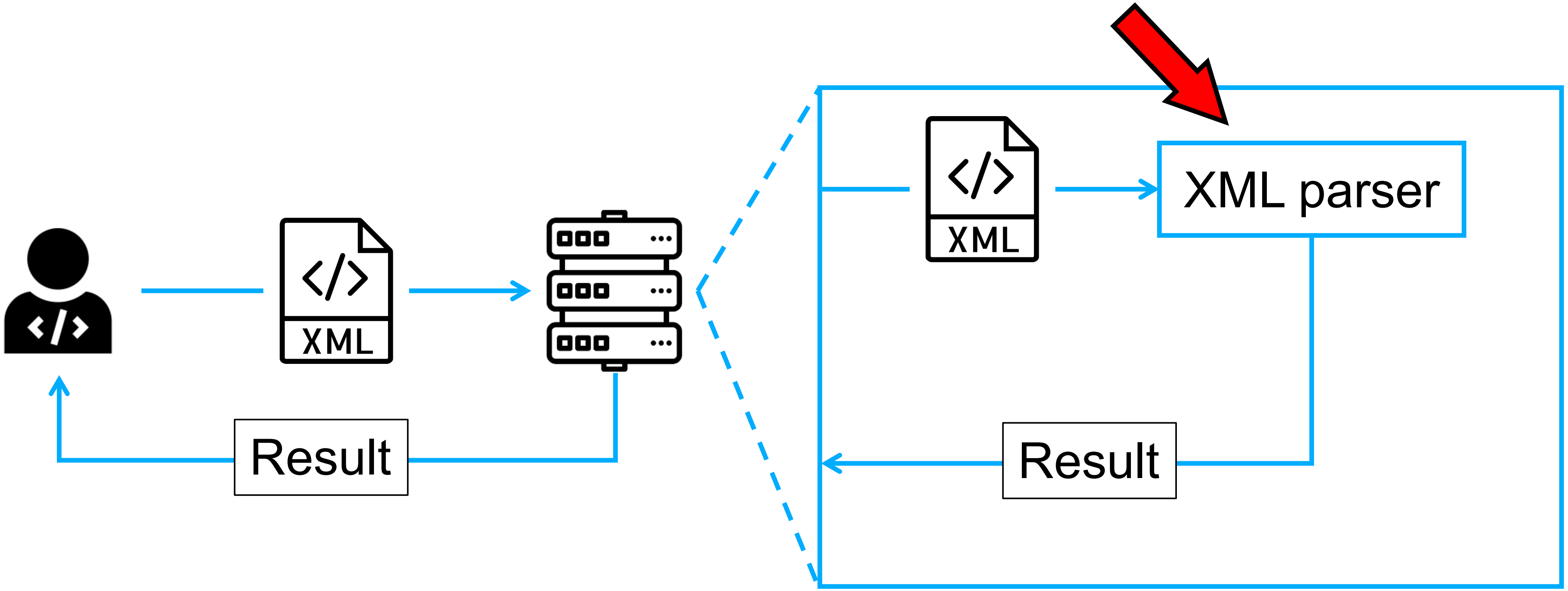
Как данные передаются?

**Санитайзеры**

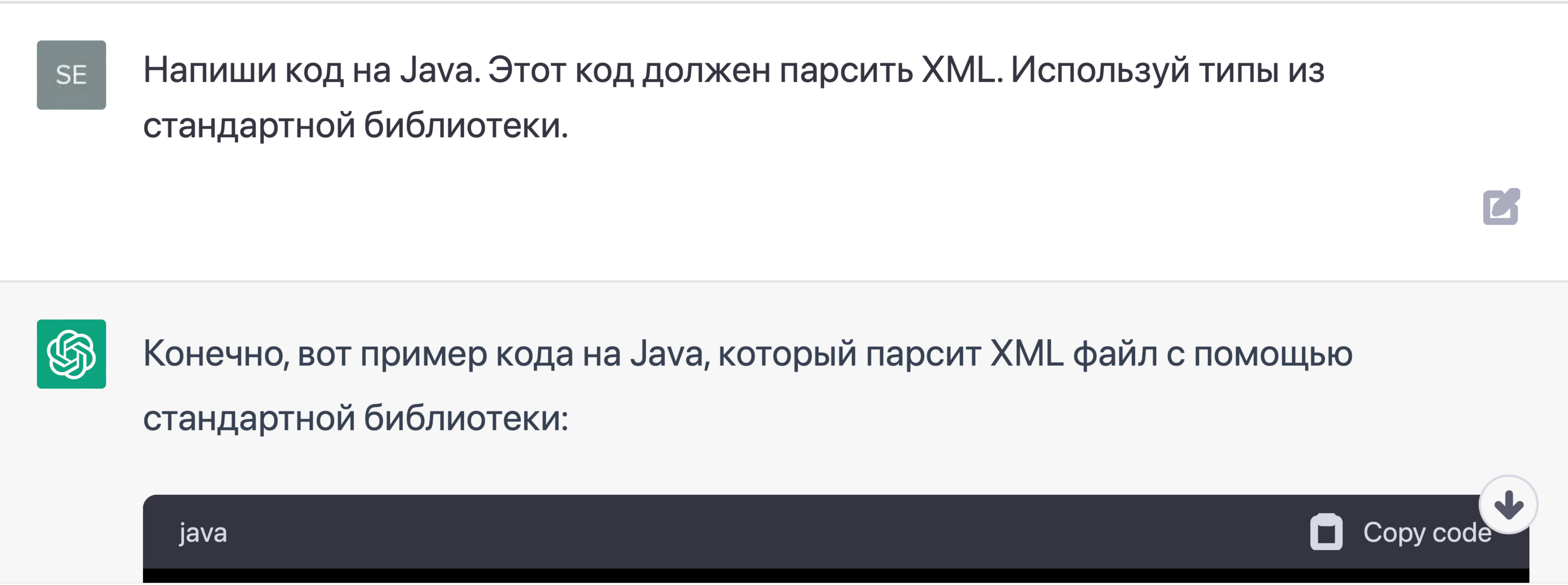
Как данные проверяются?

**Приёмники**

Куда данные не должны попасть?


XML
Result
XML
XML parser
Result

# Пишем XML-парсер с помощью ChatGPT

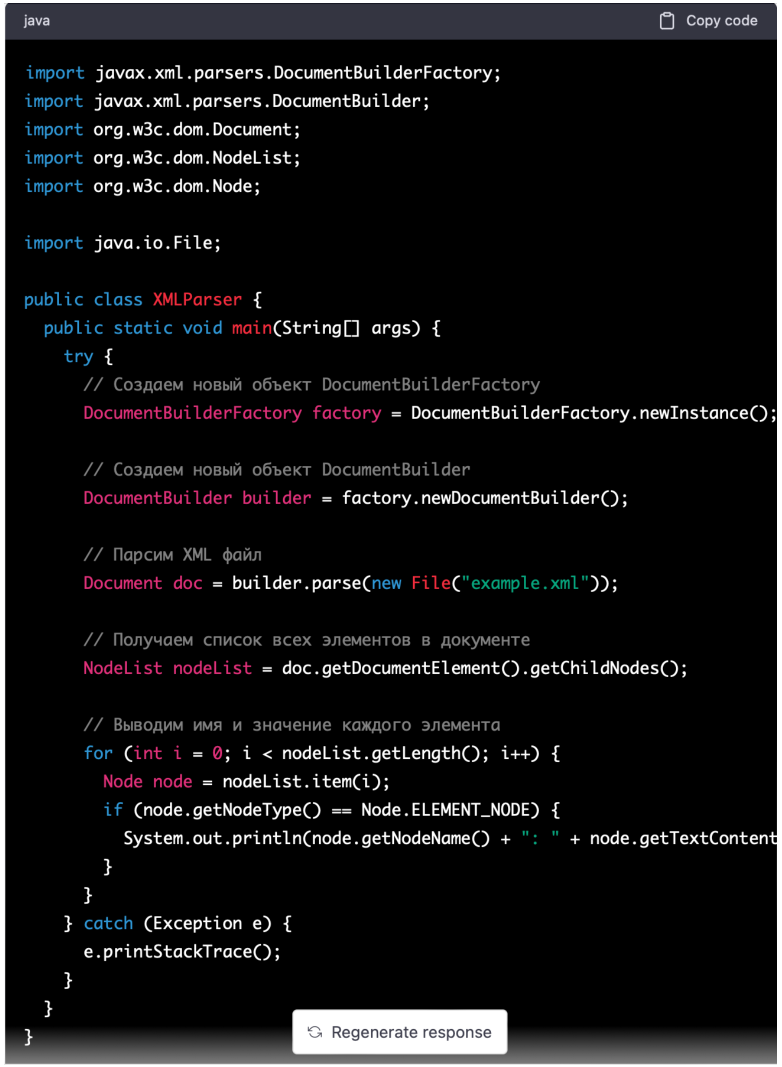


Напиши код на Java. Этот код должен парсить XML. Используй типы из стандартной библиотеки.

Конечно, вот пример кода на Java, который парсит XML файл с помощью стандартной библиотеки:

# Пишем XML-парсер с помощью ChatGPT

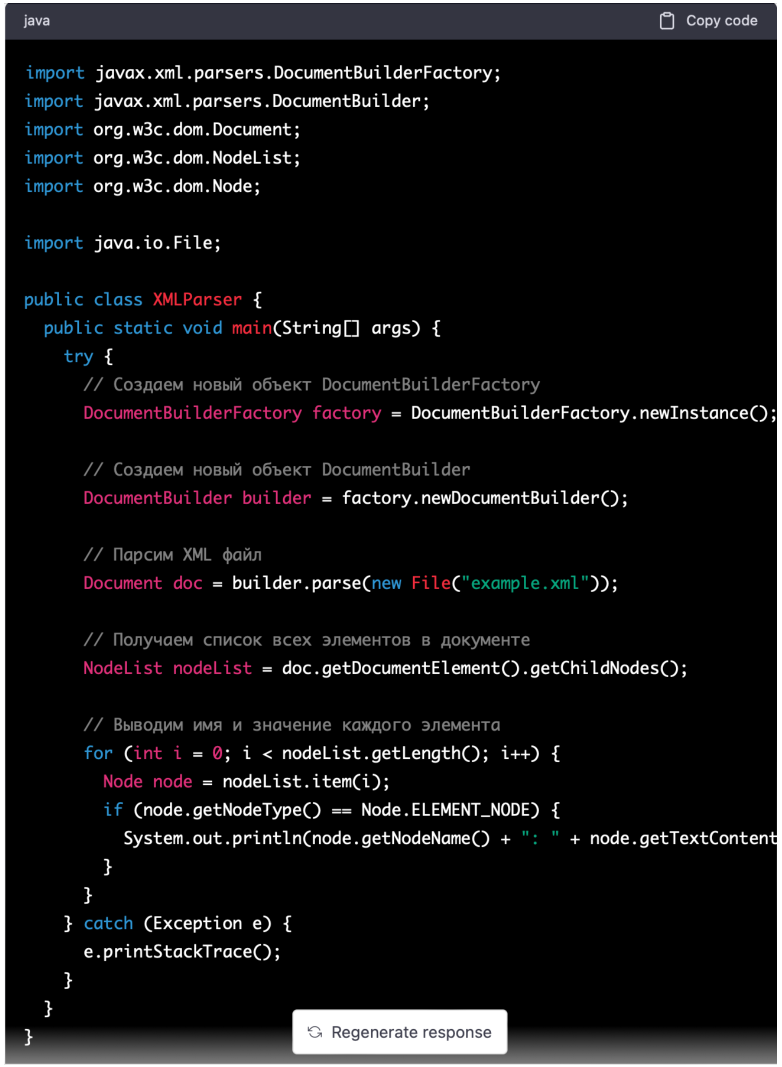
```java
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;

import java.io.File;

public class XMLParser {
    public static void main(String[] args) {
        try {
            // Создаем новый объект DocumentBuilderFactory
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

            // Создаем новый объект DocumentBuilder
            DocumentBuilder builder = factory.newDocumentBuilder();

            // Парсим XML файл
            Document doc = builder.parse(new File("example.xml"));

            // Получаем список всех элементов в документе
            NodeList nodeList = doc.getDocumentElement().getChildNodes();

            // Выводим имя и значение каждого элемента
            for (int i = 0; i < nodeList.getLength(); i++) {
                Node node = nodeList.item(i);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    System.out.println(node.getNodeName() + ": " + node.getTextContent());
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



```java
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;

import java.io.File;

public class XMLParser {
  public static void main(String[] args) {
    try {
      // Создаем новый объект DocumentBuilderFactory
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

      // Создаем новый объект DocumentBuilder
      DocumentBuilder builder = factory.newDocumentBuilder();

      // Парсим XML файл
      Document doc = builder.parse(new File("example.xml"));

      // Получаем список всех элементов в документе
      NodeList nodeList = doc.getDocumentElement().getChildNodes();

      // Выводим имя и значение каждого элемента
      for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
          System.out.println(node.getNodeName() + ": " + node.getTextContent
        }
      }
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

# Пишем XML-парсер с помощью ChatGPT

```
try {
  DocumentBuilderFactory factory
    = DocumentBuilderFactory.newInstance();
  DocumentBuilder builder = factory.newDocumentBuilder();
  Document doc = builder.parse(new File("example.xml"));

  NodeList nodeList = doc.getDocumentElement()
                         .getChildNodes();

    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            System.out.println(node.getNodeName() + ": "
           + node.getTextContent());
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}
```



```java
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;

import java.io.File;

public class XMLParser {
  public static void main(String[] args) {
    try {
      // Создаем новый объект DocumentBuilderFactory
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

      // Создаем новый объект DocumentBuilder
      DocumentBuilder builder = factory.newDocumentBuilder();

      // Парсим XML файл
      Document doc = builder.parse(new File("example.xml"));

      // Получаем список всех элементов в документе
      NodeList nodeList = doc.getDocumentElement().getChildNodes();

      // Выводим имя и значение каждого элемента
      for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
          System.out.println(node.getNodeName() + ": " + node.getTextContent
        }
      }
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

# Пишем XML-парсер с помощью ChatGPT

```
File xmlFromUser = getXmlFromUser();

DocumentBuilderFactory factory
    = DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();

Document doc = builder.parse(xmlFromUser);

// XML processing
....

sentResultToUser();
```

# Парсинг XML в проекете c3p0

```
public static C3P0Config
extractXmlConfigFromInputStream(InputStream is) throws ....
{
  DocumentBuilderFactory fact
    =  DocumentBuilderFactory.newInstance();
  DocumentBuilder db = fact.newDocumentBuilder();
  Document doc = db.parse(is);

  return extractConfigFromXmlDoc(doc);
}
```

# Пишем XML-парсер с помощью ChatGPT

```
File xmlFromUser = getXmlFromUser();

DocumentBuilderFactory factory
    = DocumentBuilderFactor
DocumentBuilder bui                                         ();

Docu                                    ser);

// XM
....

sentResultToUser();
```

VULNERABLE

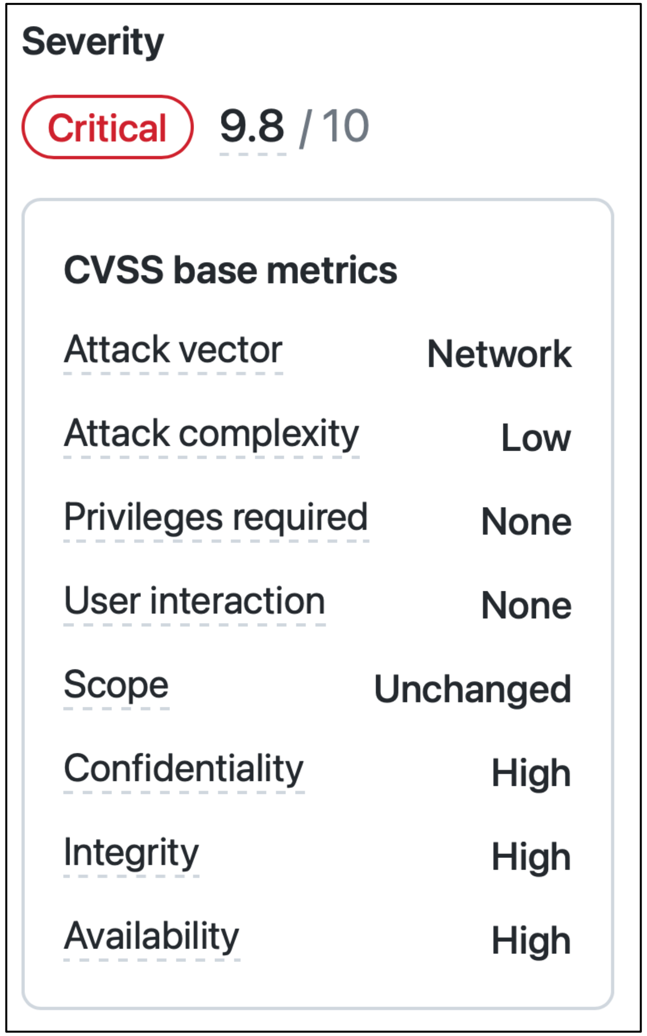
# Парсинг XML в проекете c3p0

```
public static C3P0Config
extractXmlConfigFromInputStream(InputStream i                 ....
{
  DocumentBuilderFactory fa
     =  DocumentBuil
  Docume                                     r();
  Do

  retu                   igFromXmlDoc(doc);
}
```

VULNERABLE

# CVE-2018-20433

```
public static C3P0Config
extractXmlConfigFromInputStream(InputStream is) t
{
  DocumentBuilderFactory fact
     =  DocumentBuilderFactory.newInstance();
  DocumentBuilder db = fact.newDocumentBuilder();
  Document doc = db.parse(is);

  return extractConfigFromXmlDoc(doc);
}
```



**Severity**

Critical 9.8 / 10

**CVSS base metrics**

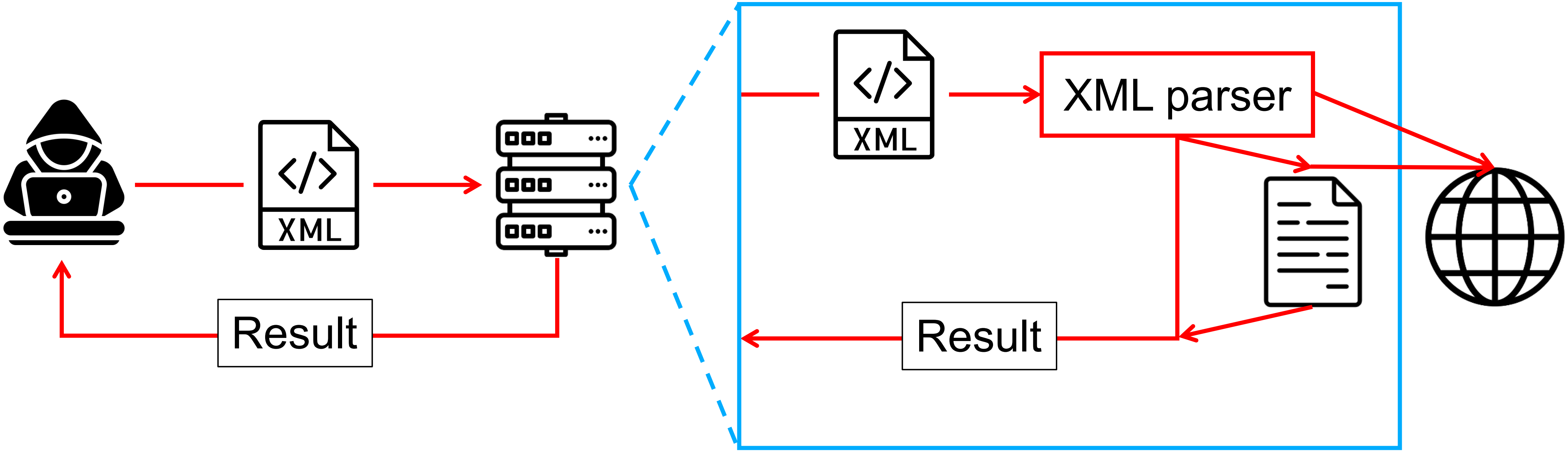
| | |
|---|---|
| Attack vector | Network |
| Attack complexity | Low |
| Privileges required | None |
| User interaction | None |
| Scope | Unchanged |
| Confidentiality | High |
| Integrity | High |
| Availability | High |

# Insecure XML processing

Semgrep

**java.lang.security.audit.xxe.documentbuilderfactory-disallow-doctype-decl-missing.documentbuilderfactory-disallow-doctype-decl-missing**

DOCTYPE declarations are enabled for this DocumentBuilderFactory. This is vulnerable to XML external entity attacks. Disable this by setting the feature "http://apache.org/xml/features/disallow-doctype-decl" to true. Alternatively, allow DOCTYPE declarations and only prohibit external entities declarations. This can be done by setting the features "http://xml.org/sax/features/external-general-entities" and "http://xml.org/sax/features/external-parameter-entities" to false.

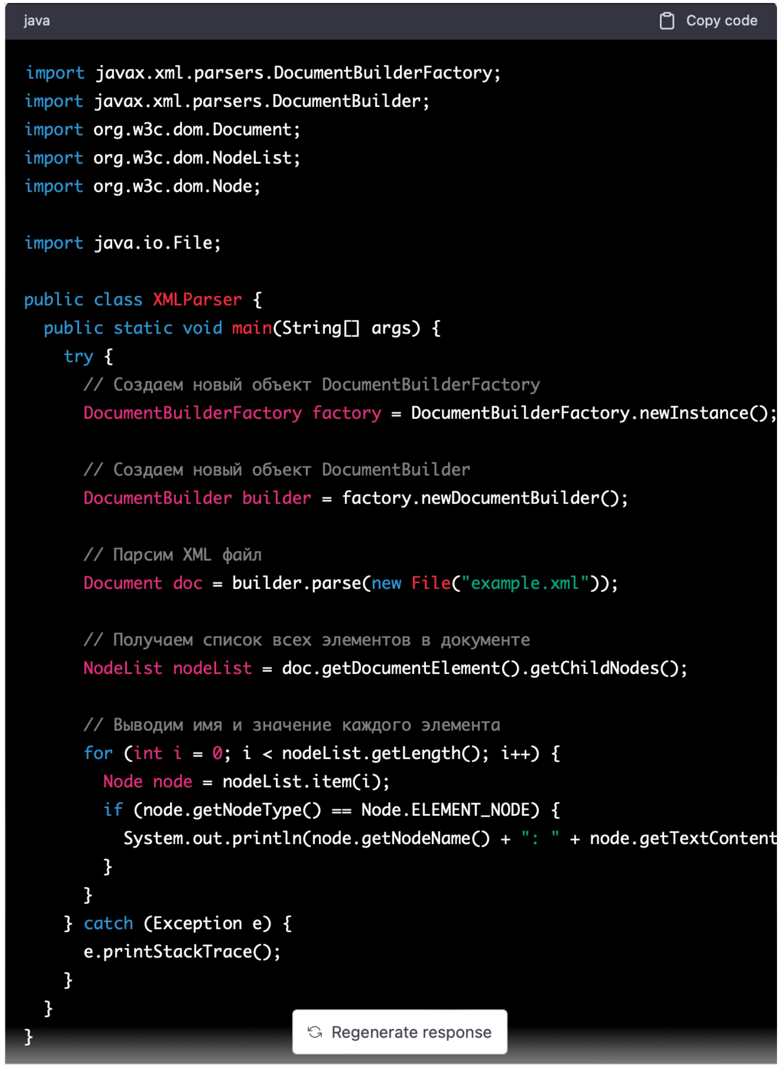

XML
Result
XML
XML parser
Result

```
<!ENTITY hostsEntity SYSTEM  "file:///C:/Windows/System32/drivers/etc/hosts">

<!ENTITY hackEntity SYSTEM  "https://myEvilSite.com/xxe">
```

# XXE
# (XML eXternal entities)

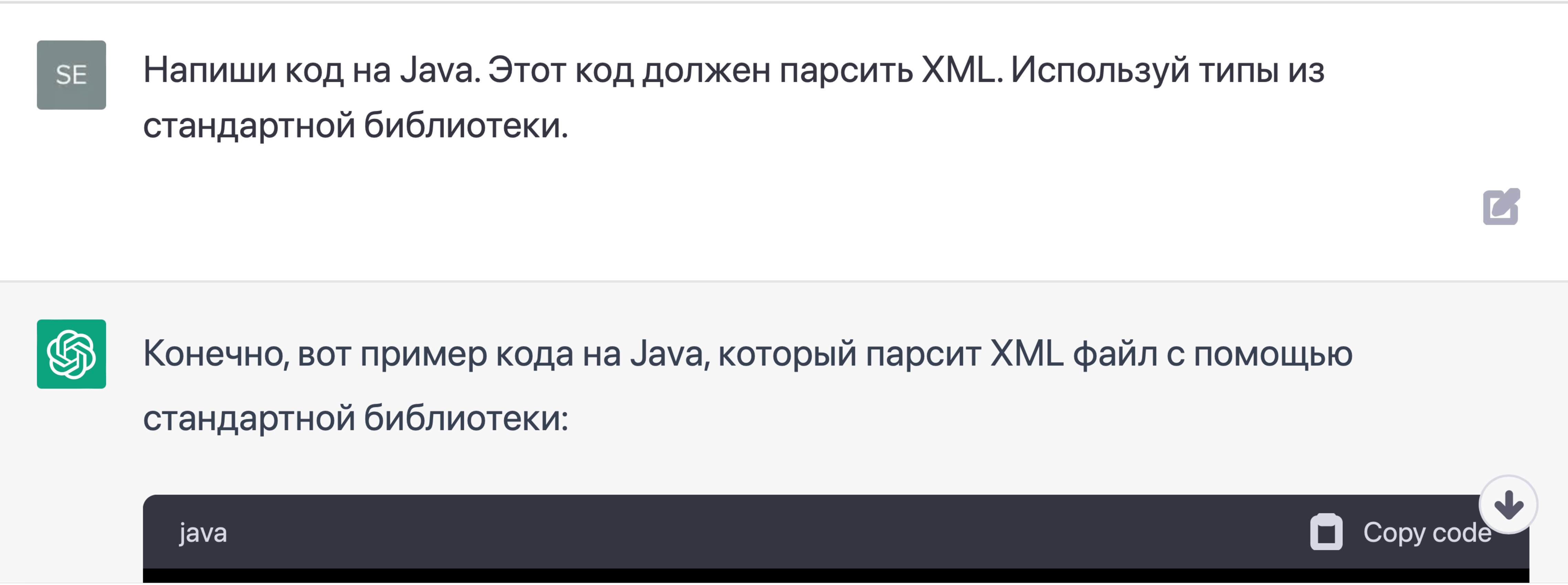
# Пишем XML-парсер с помощью ChatGPT

```java
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;

import java.io.File;

public class XMLParser {
    public static void main(String[] args) {
        try {
            // Создаем новый объект DocumentBuilderFactory
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

            // Создаем новый объект DocumentBuilder
            DocumentBuilder builder = factory.newDocumentBuilder();

            // Парсим XML файл
            Document doc = builder.parse(new File("example.xml"));

            // Получаем список всех элементов в документе
            NodeList nodeList = doc.getDocumentElement().getChildNodes();

            // Выводим имя и значение каждого элемента
            for (int i = 0; i < nodeList.getLength(); i++) {
                Node node = nodeList.item(i);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    System.out.println(node.getNodeName() + ": " + node.getTextContent());
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



```java
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;

import java.io.File;

public class XMLParser {
  public static void main(String[] args) {
    try {
      // Создаем новый объект DocumentBuilderFactory
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

      // Создаем новый объект DocumentBuilder
      DocumentBuilder builder = factory.newDocumentBuilder();

      // Парсим XML файл
      Document doc = builder.parse(new File("example.xml"));

      // Получаем список всех элементов в документе
      NodeList nodeList = doc.getDocumentElement().getChildNodes();

      // Выводим имя и значение каждого элемента
      for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
          System.out.println(node.getNodeName() + ": " + node.getTextContent
        }
      }
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

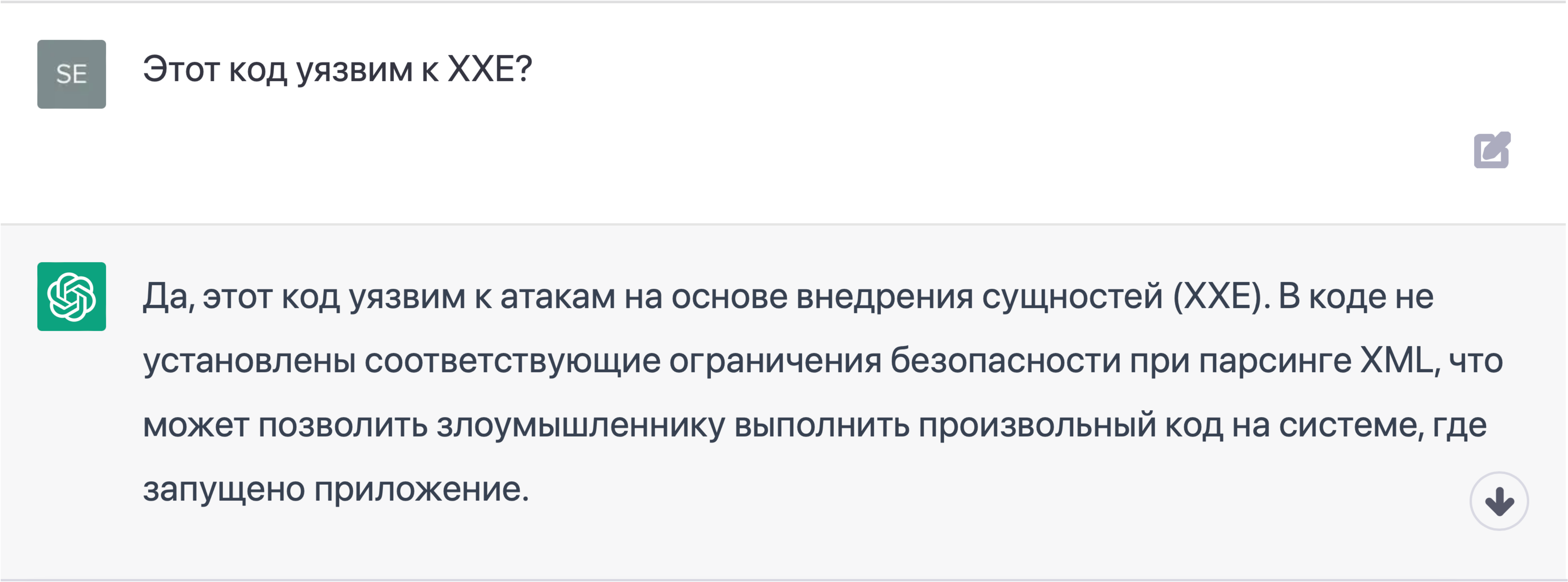
# Пишем XML-парсер с помощью ChatGPT



Напиши код на Java. Этот код должен парсить XML. Используй типы из стандартной библиотеки.

Конечно, вот пример кода на Java, который парсит XML файл с помощью стандартной библиотеки:

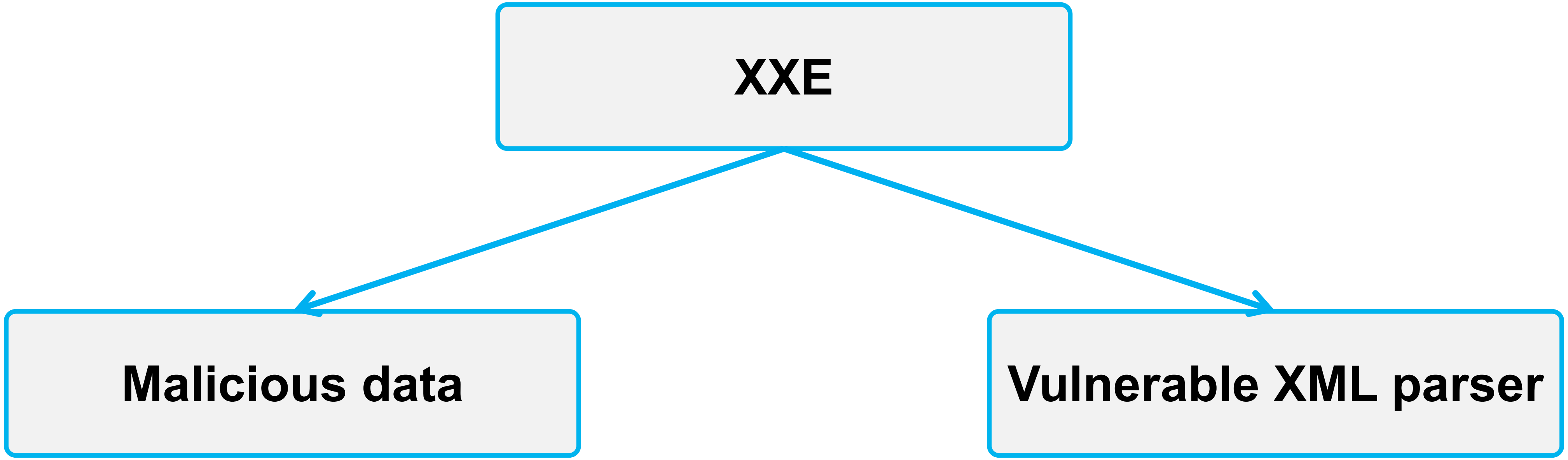
# Пишем XML-парсер с помощью ChatGPT



Этот код уязвим к XXE?

Да, этот код уязвим к атакам на основе внедрения сущностей (XXE). В коде не установлены соответствующие ограничения безопасности при парсинге XML, что может позволить злоумышленнику выполнить произвольный код на системе, где запущено приложение.

# XML eXternal entities



XXE

Malicious data

Vulnerable XML parser

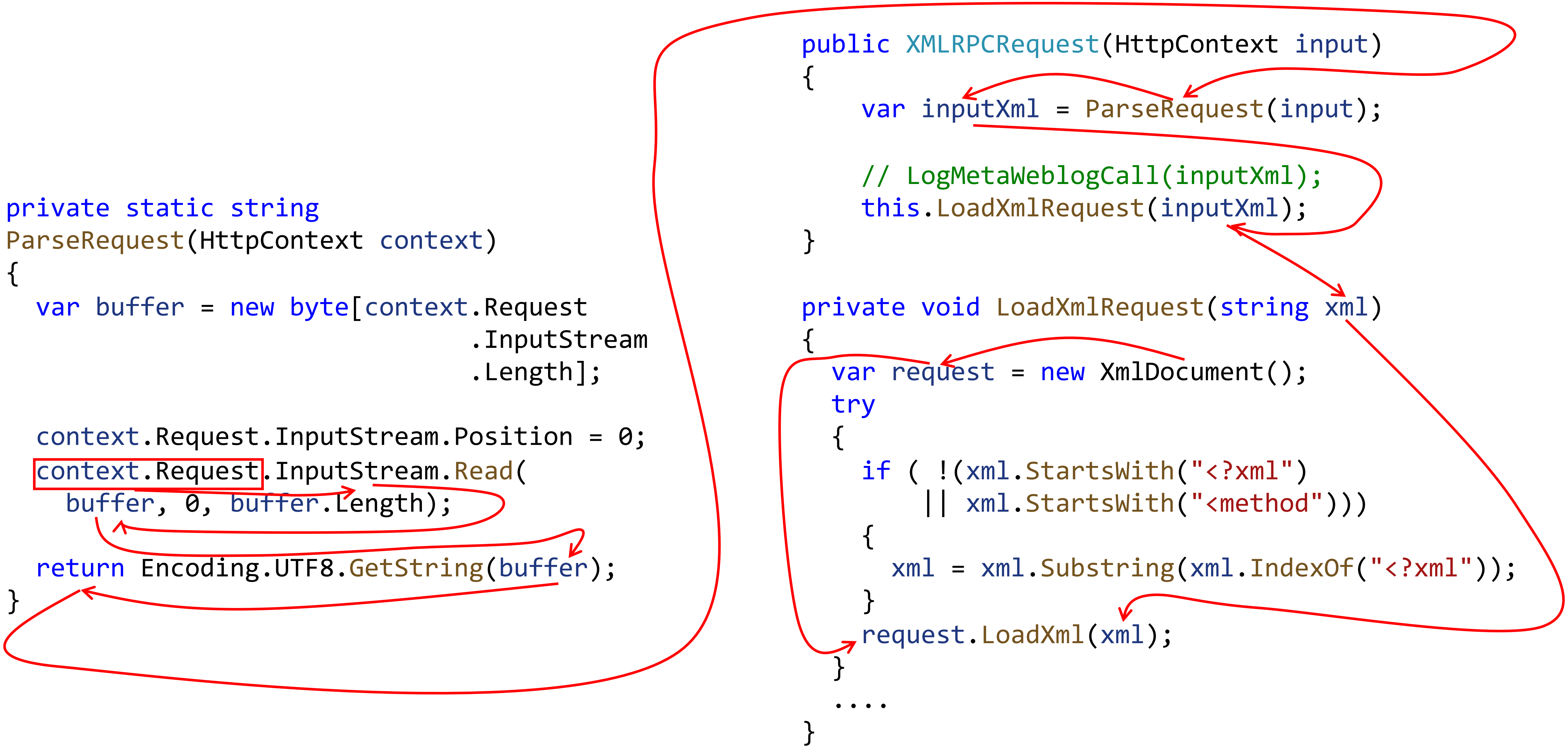
# XXE

```
public static C3P0Config
extractXmlConfigFromInputStream(InputStream is) throws ....
{
  DocumentBuilderFactory fact
     =  DocumentBuilderFactory.newInstance();
  DocumentBuilder db = fact.newDocumentBuilder();

  Document doc = db.parse(is);

  return extractConfigFromXmlDoc(doc);
}
```

# XXE

```
public static C3P0Config
extractXmlConfigFromInputStream(InputStream is) throws ....
{
  DocumentBuilderFactory fact
    =  DocumentBuilderFactory.newInstance();

  fact.setFeature(
    "http://apache.org/xml/features/disallow-doctype-decl",
    true);

  DocumentBuilder db = fact.newDocumentBuilder();
  Document doc = db.parse(is);

  return extractConfigFromXmlDoc(doc);
}
```

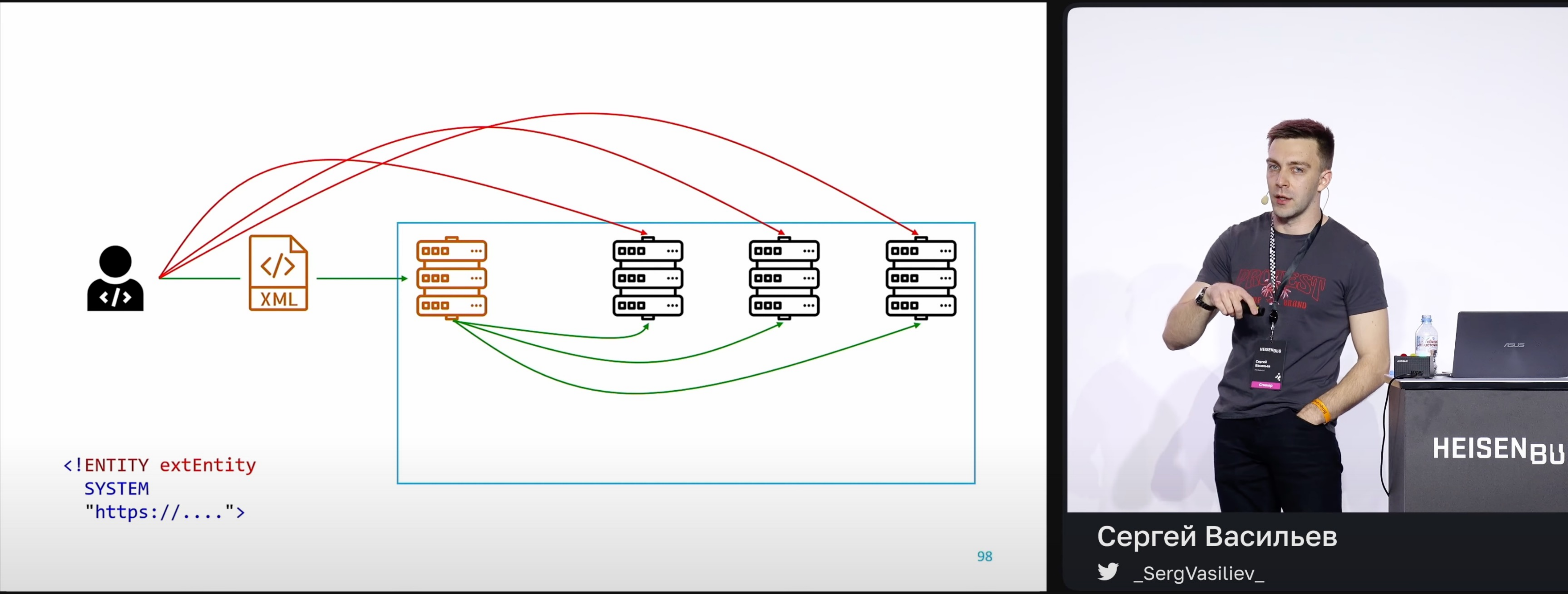
# BlogEngine.NET: CVE-2018-14485



```
private static string
ParseRequest(HttpContext context)
{
  var buffer = new byte[context.Request
                               .InputStream
                               .Length];

  context.Request.InputStream.Position = 0;
  context.Request.InputStream.Read(
    buffer, 0, buffer.Length);

  return Encoding.UTF8.GetString(buffer);
}
```

```
public XMLRPCRequest(HttpContext input)
{
    var inputXml = ParseRequest(input);

    // LogMetaWeblogCall(inputXml);
    this.LoadXmlRequest(inputXml);
}

private void LoadXmlRequest(string xml)
{
  var request = new XmlDocument();
  try
  {
    if ( !(xml.StartsWith("<?xml")
        || xml.StartsWith("<method")))
    {
      xml = xml.Substring(xml.IndexOf("<?xml"));
    }
    request.LoadXml(xml);
  }
  ....
}
```

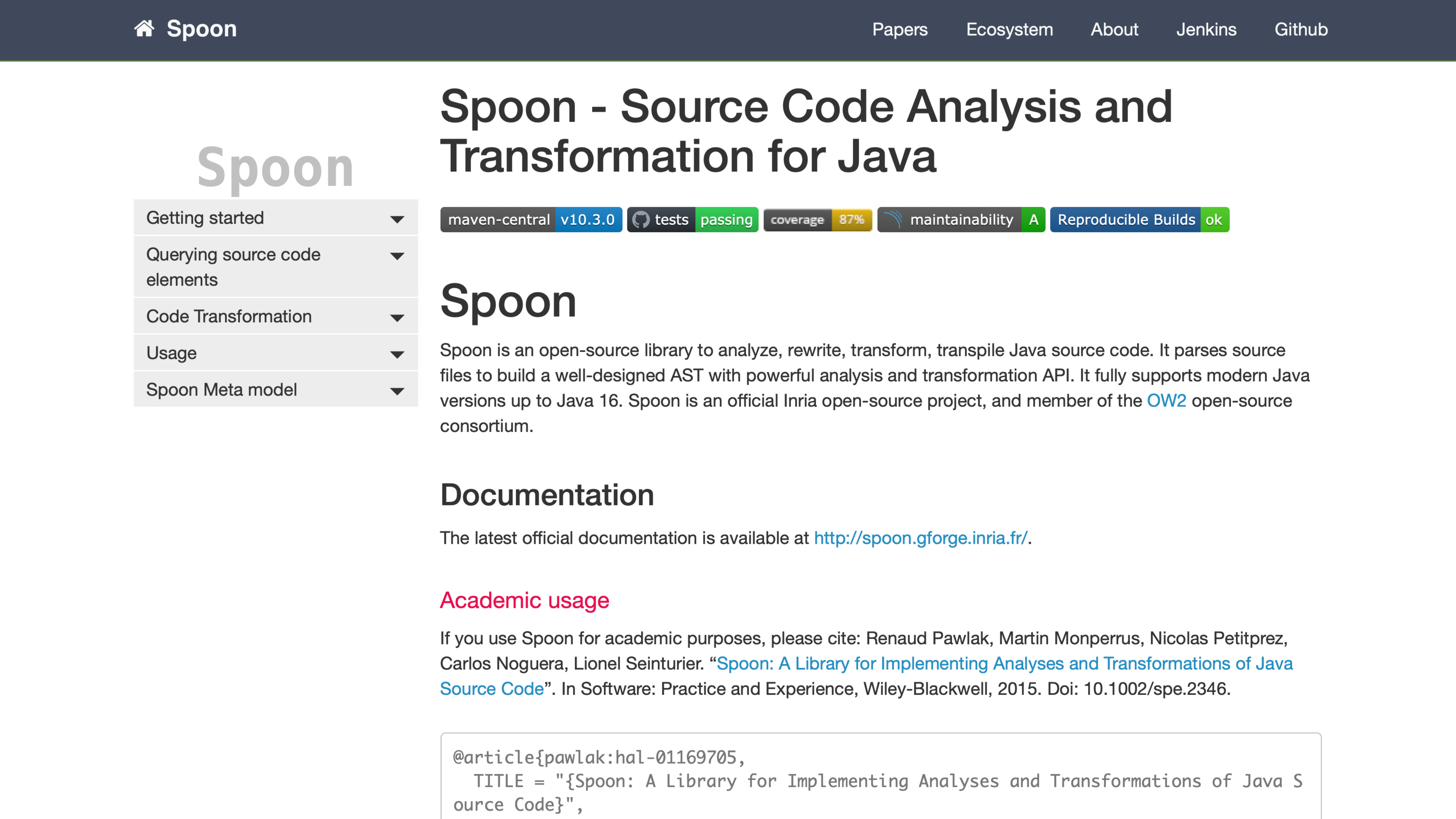
# Правильно ли вы парсите XML?



```
<!ENTITY extEntity
  SYSTEM
  "https://....">
```

[https://youtu.be/1OuJkOH8aP4](https://youtu.be/1OuJkOH8aP4)

# Что дальше?



# Spoon - Source Code Analysis and Transformation for Java

## Spoon

Spoon is an open-source library to analyze, rewrite, transform, transpile Java source code. It parses source files to build a well-designed AST with powerful analysis and transformation API. It fully supports modern Java versions up to Java 16. Spoon is an official Inria open-source project, and member of the OW2 open-source consortium.

## Documentation

The latest official documentation is available at http://spoon.gforge.inria.fr/.

### Academic usage

If you use Spoon for academic purposes, please cite: Renaud Pawlak, Martin Monperrus, Nicolas Petitprez, Carlos Noguera, Lionel Seinturier. "Spoon: A Library for Implementing Analyses and Transformations of Java Source Code". In Software: Practice and Experience, Wiley-Blackwell, 2015. Doi: 10.1002/spe.2346.

```
@article{pawlak:hal-01169705,
  TITLE = "{Spoon: A Library for Implementing Analyses and Transformations of Java S
ource Code}",
```

# Spoon

```
public class
CatchProcessor extends AbstractProcessor<CtCatch> {
  public void process(CtCatch element) {
    // we get all statements and if there isn't statement,
    // it means the block catch is empty!
    if (element.getBody().getStatements().size() == 0) {

      getFactory().getEnvironment()
                  .report(this,
                          Level.WARN,
                          element,
                          "empty catch clause");
    }
  }
}
```

# Spoon

- Сайт: https://spoon.gforge.inria.fr/index.html
- Примеры правил: https://github.com/SpoonLabs/spoon-examples

- Подключение через Maven:

```
<dependency>
    <groupId>fr.inria.gforge.spoon</groupId>
    <artifactId>spoon-core</artifactId>
    <version>10.2.0</version>
</dependency>
```

POWERED BY SEMGREP OSS AND PRO ENGINE

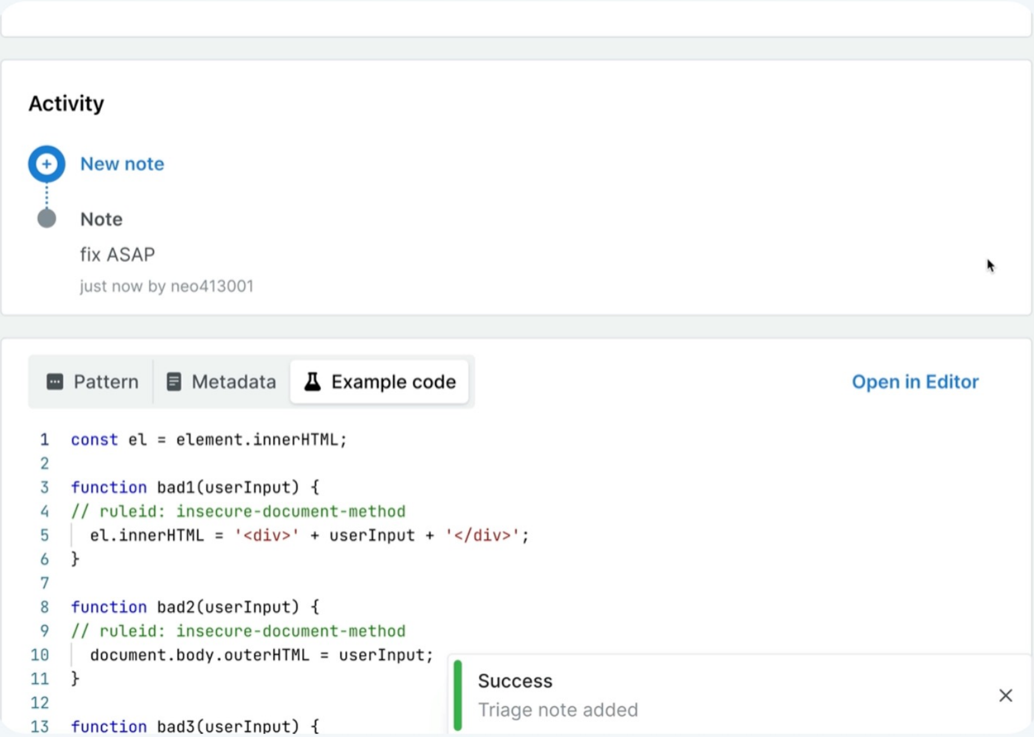
# Semgrep Code

Deploy, manage, and monitor Semgrep at scale

*Scan 30+ languages with 2,750+ Community and Pro rules*

Talk to a product advisor

Documentation



**Activity**

New note

**Note**

fix ASAP

just now by neo413001

```
const el = element.innerHTML;

function bad1(userInput) {
// ruleid: insecure-document-method
  el.innerHTML = '<div>' + userInput + '</div>';
}

function bad2(userInput) {
// ruleid: insecure-document-method
  document.body.outerHTML = userInput;
}

function bad3(userInput) {
```

Success

Triage note added

# Semgrep

- Сайт: https://semgrep.dev
- SAST
- Open source
- Можно писать свои правила

# PVS-Studio выбирают за...

**Экспертную техническую поддержку**

**Возможность офлайн-использования**

**Простоту использования**

**Диагностические возможности**

Мы разработали более 900 диагностических правил и добавляем новые каждый месяц.

**Кроссплатформенность**

**Удобные отчёты анализатора**

**Удобный CLI**

**Плагины**

**Режим проверки Legacy-кода**

**Работу с ложными срабатываниями**

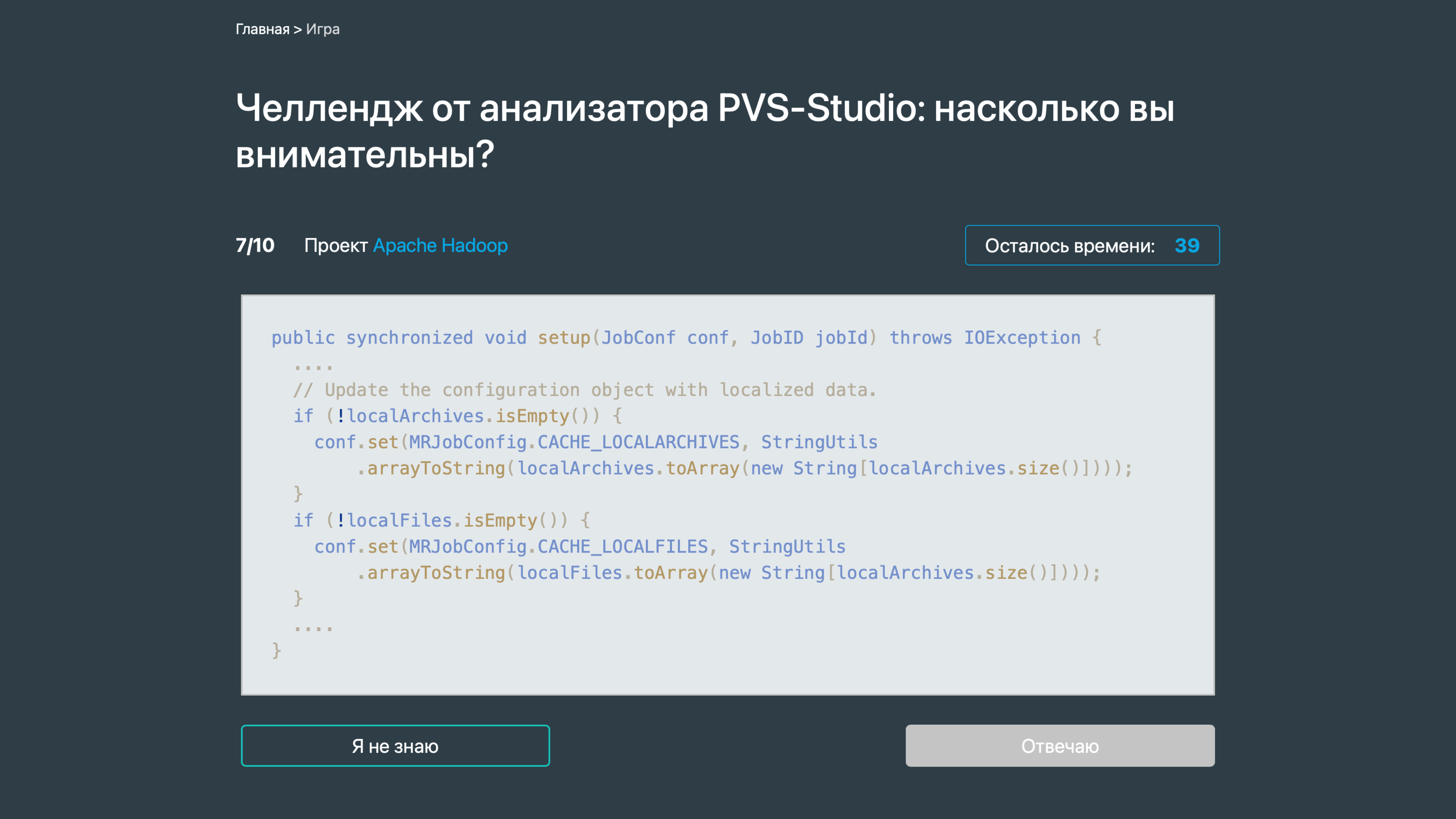
Скачать дистрибутив

# PVS-Studio

- Попробовать:
  https://pvs-studio.ru/jpoint_2023

- Промокод
  (триал на 30 дней): **jpoint_23**

- Интеграция:
  - IntelliJ IDEA
  - Maven
  - Gradle
  - SonarQube



Главная > Игра

# Челлендж от анализатора PVS-Studio: насколько вы внимательны?

**7/10** Проект Apache Hadoop

Осталось времени: **39**

```
public synchronized void setup(JobConf conf, JobID jobId) throws IOException {
  ....
  // Update the configuration object with localized data.
  if (!localArchives.isEmpty()) {
    conf.set(MRJobConfig.CACHE_LOCALARCHIVES, StringUtils
        .arrayToString(localArchives.toArray(new String[localArchives.size()])));
  }
  if (!localFiles.isEmpty()) {
    conf.set(MRJobConfig.CACHE_LOCALFILES, StringUtils
        .arrayToString(localFiles.toArray(new String[localArchives.size()])));
  }
  ....
}
```

# Квизы на поиск ошибок

- 10 примеров с ошибками
- 60 секунд на каждую ошибку

- Языки:
  - Java https://quiz.pvs-studio.com/ru/java/
  - C# https://quiz.pvs-studio.com/ru/csharp/
  - C++ https://quiz.pvs-studio.com/ru/cpp/

Промокод PVS-Studio

**jpoint_23**

**Сергей Васильев**

pvs-studio.com

vasiliev@viva64.com