Рефлексируем: один баг, Java-проект, AI агент и много семантических анализаторов. Что может пойти не так?

Артём Симешин · Joker 2026

Apache Calcite · opencode · ripgrep · ast-grep · MCP IDEA · Serena + JDTLS · Graphify · Understand-Anything



Кто я

- Долго работаю в Сбере
- Люблю опенсорс
- Люблю иннерсорс
- Если вижу проблему, то решаю её

О чём поговорим

01 **Мы уже работаем с агентами** проектируем, кодим, тестим, пишем и формулируем требования

О чём поговорим

01 **Мы уже работаем с агентами** — проектируем, кодим, тестим, пишем и формулируем требования

02 **У каждого свой набор** — харнесы и агенты на разных моделях: токены и время тратим по-разному, но тратим все

О чём поговорим

01 **Мы уже работаем с агентами** — проектируем, кодим, тестим, пишем и формулируем требования

02 **У каждого свой набор** — харнесы и агенты на разных моделях: токены и время тратим по-разному, но тратим все

03 **Так ли хорош агент без инструментов** — на типовой задаче в большом опенсорс-проекте

О чём поговорим

01 **Мы уже работаем с агентами** — проектируем, кодим, тестим, пишем и формулируем требования

02 **У каждого свой набор** — харнесы и агенты на разных моделях: токены и время тратим по-разному, но тратим все

03 **Так ли хорош агент без инструментов** — на типовой задаче в большом опенсорс-проекте

04 **Каких инструментов ему не хватает** — чтобы решать задачи эффективнее

opencode

Популярный open-source агент:
легко кастомизируется, базовый
набор тулов в комплекте.
Модель подключается любая, у нас
DeepSeek V4 Flash через OpenRouter.

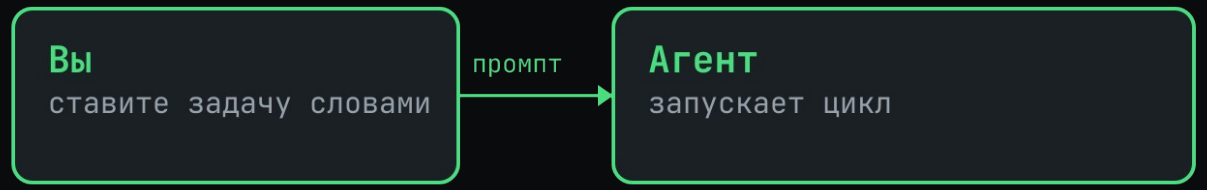


github.com/anomalyco/opencode

Агент, на котором прогоняли бенчмарк

Read	Grep	Glob	Edit	Bash	MCP
инструмент	инструмент	инструмент	инструмент	инструмент	инструмент

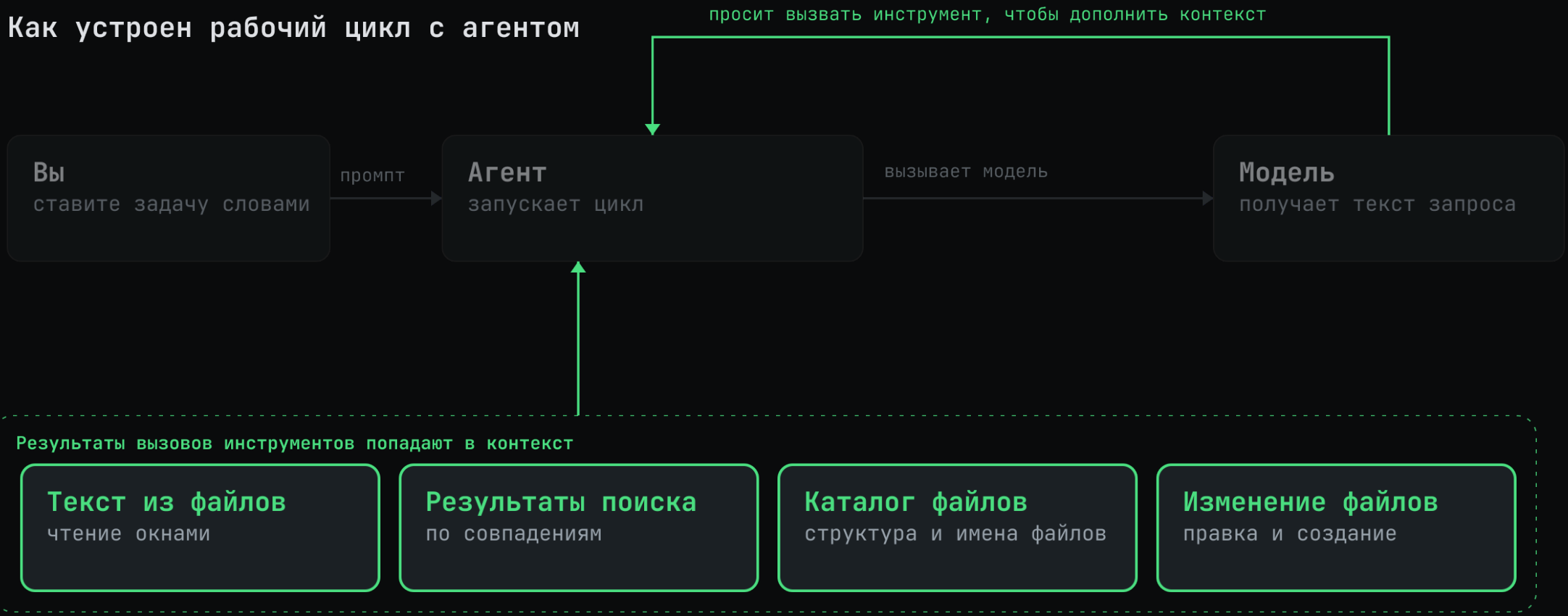
Как устроен рабочий цикл с агентом



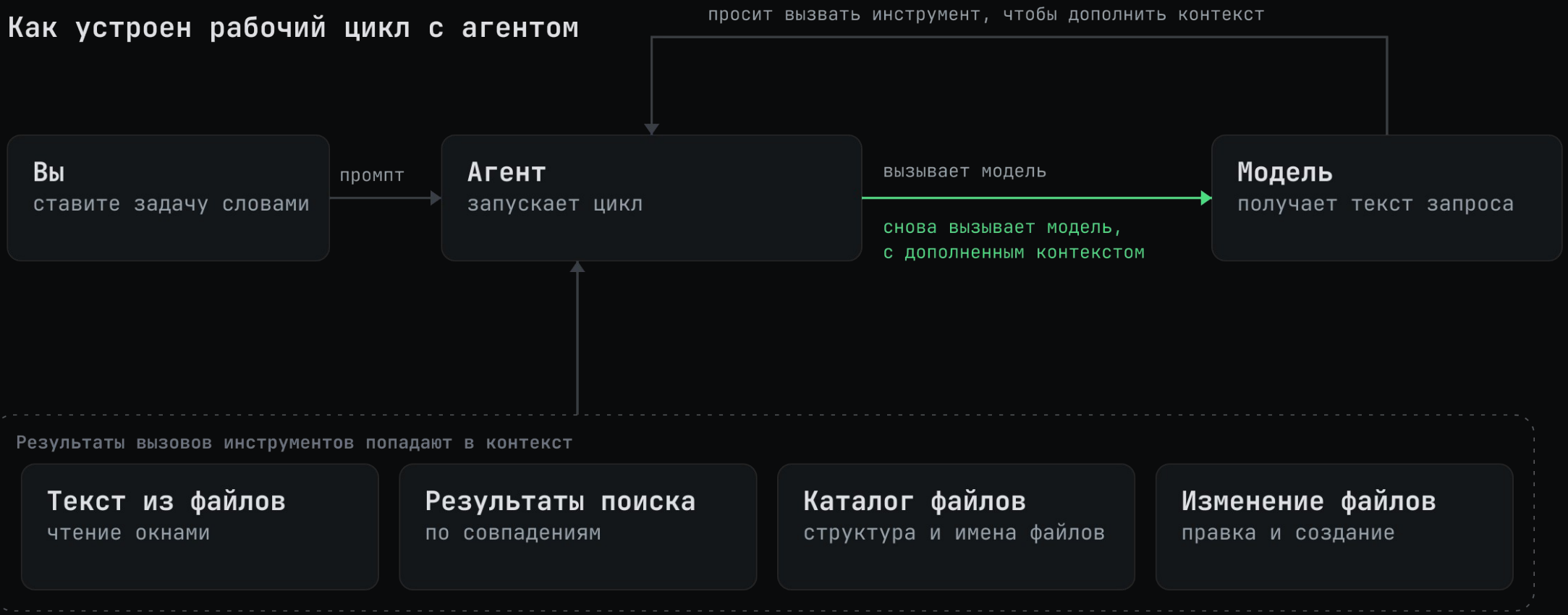
Как устроен рабочий цикл с агентом

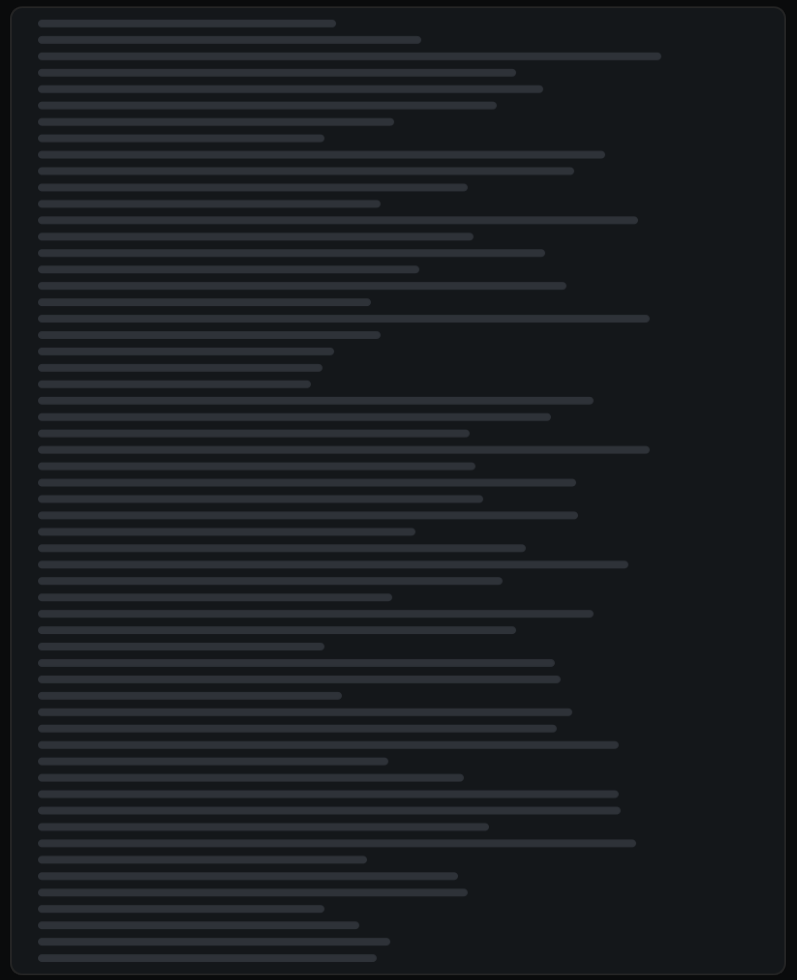


Как устроен рабочий цикл с агентом



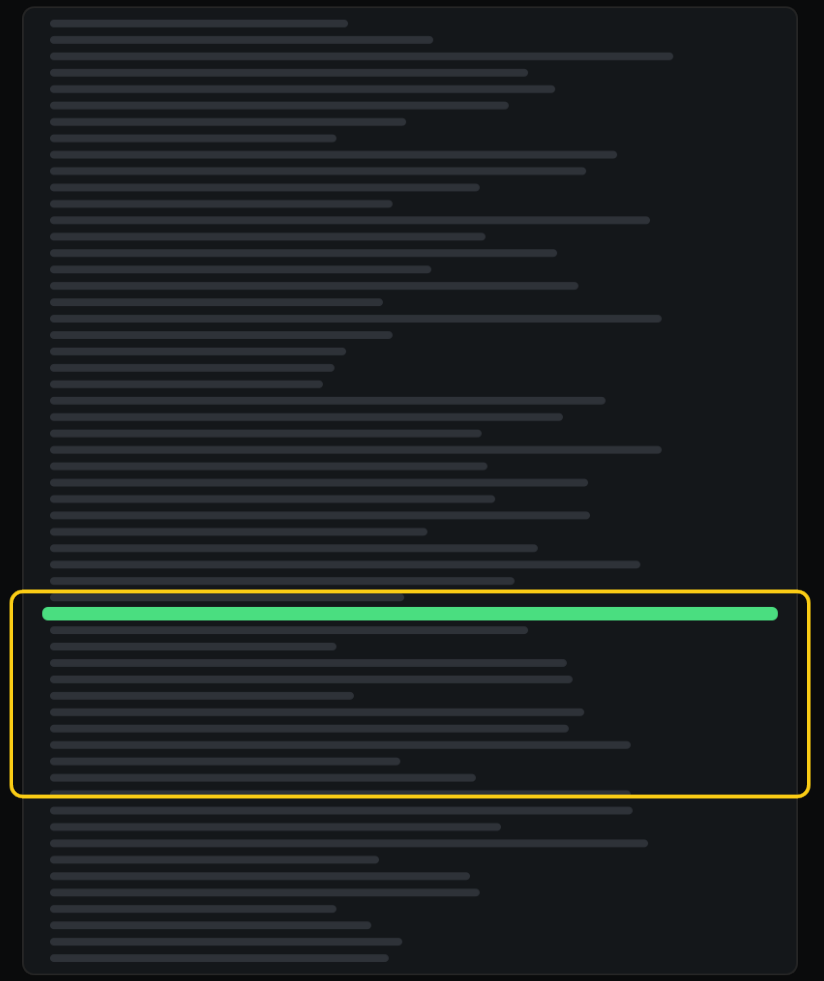
Как устроен рабочий цикл с агентом





Один большой класс

9 426 строк кода



Один большой класс

9 426 строк кода

Нужное место: 15 строк

меньше 0,2% файла

Окно одного чтения: 2 000 строк

это 21% файла

Чтобы прочитать эти 15 строк, надо угадать, с какой строки начинать

Что идёт не так

01 **Лишняя информация в контексте** вместе с нужным местом в контекст попадает всё окно чтения

Что идёт не так

01 **Лишняя информация в контексте** вместе с нужным местом в контекст попадает всё окно чтения

02 **Переполнение и автокомпакт** контекст переполняется, автоматическое сжатие теряет важные данные

Что идёт не так

01 **Лишняя информация в контексте** вместе с нужным местом в контекст попадает всё окно чтения

02 **Переполнение и автокомпакт** контекст переполняется, автоматическое сжатие теряет важные данные

03 **Некорректные решения** лишний контекст уводит модель от правильного ответа

Что идёт не так

01 **Лишняя информация в контексте** вместе с нужным местом в контекст попадает всё окно чтения

02 **Переполнение и автокомпакт** контекст переполняется, автоматическое сжатие теряет важные данные

03 **Некорректные решения** лишний контекст уводит модель от правильного ответа

04 **Нужный файл не прочитан** нужной информации может просто не оказаться в контексте

Что идёт не так

01 **Лишняя информация в контексте** вместе с нужным местом в контекст попадает всё окно чтения

02 **Переполнение и автокомпакт** контекст переполняется, автоматическое сжатие теряет важные данные

03 **Некорректные решения** лишний контекст уводит модель от правильного ответа

04 **Нужный файл не прочитан** нужной информации может просто не оказаться в контексте

05 **Модель гадает объём чтения** не знает, сколько строк читать из файла, и делает много вызовов инструментов

Что идёт не так

01 **Лишняя информация в контексте** вместе с нужным местом в контекст попадает всё окно чтения

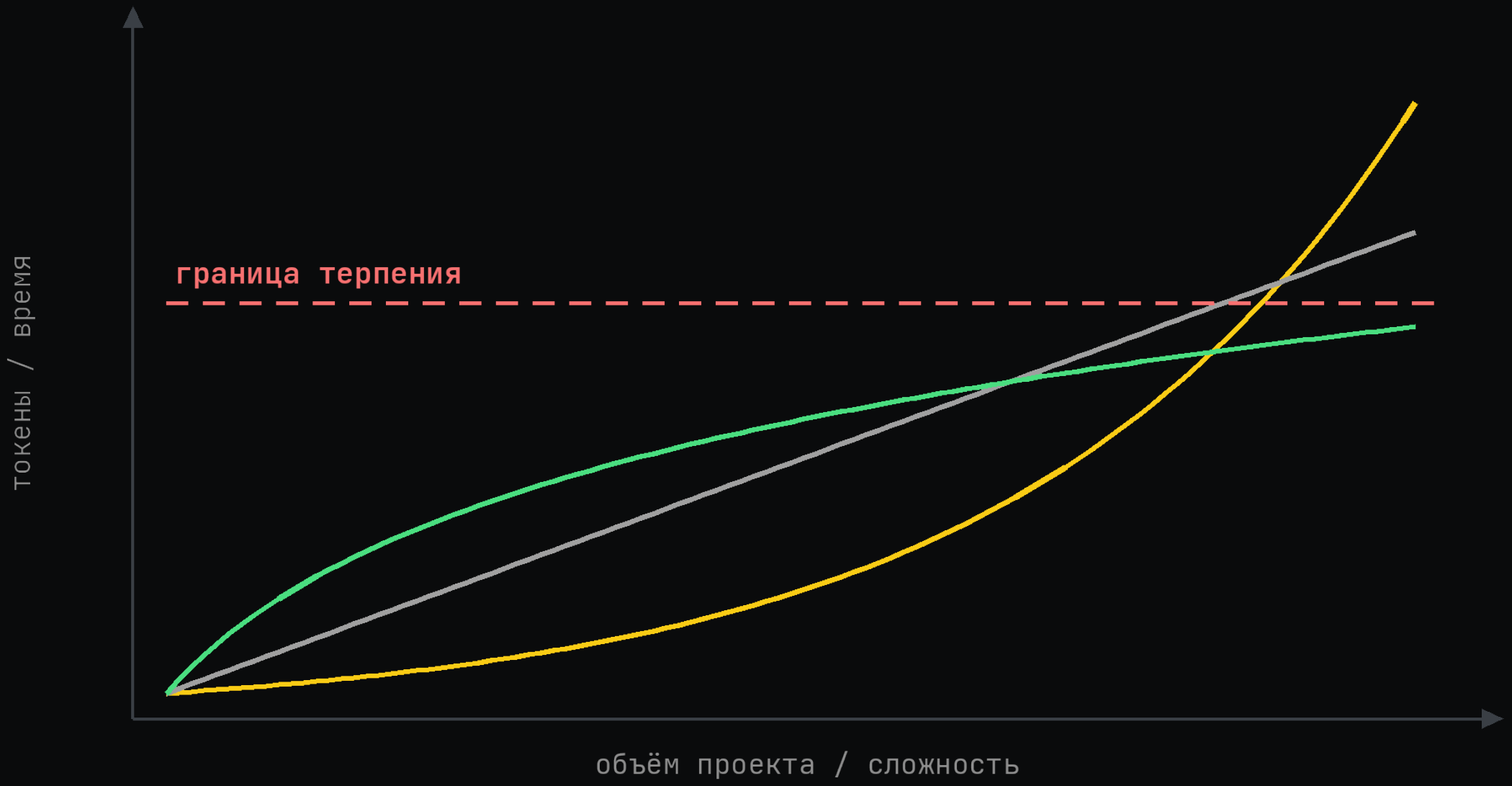
02 **Переполнение и автокомпакт** контекст переполняется, автоматическое сжатие теряет важные данные

03 **Некорректные решения** лишний контекст уводит модель от правильного ответа

04 **Нужный файл не прочитан** нужной информации может просто не оказаться в контексте

05 **Модель гадает объём чтения** не знает, сколько строк читать из файла, и делает много вызовов инструментов

06 **И всё это умножается** на количество найденных файлов для чтения и записи


граница терпения
токены / время
объём проекта / сложность

02 / 07

Один баг и бенчмарк

APACHE calcite™

Фреймворк для построения баз данных
и SQL-движков на Java: парсер и валидатор
SQL, реляционная алгебра, оптимизатор.
Сам данные не хранит, а подключается
к источникам через адаптеры.



github.com/apache/calcite

Ключевой проект Java-экосистемы для больших данных

Flink	Hive	Druid	Ignite	Drill	Beam
Apache	Apache	Apache	Apache	Apache	Apache

CALCITE-7377

Validator should reject a DESCRIPTOR in a table function
when it is not an identifier



issues.apache.org/jira

The following query crashes Calcite with a ClassCastException instead of a proper validation error:

```
select * from table(hop(table orders, descriptor(1 + 2), interval '2' hour, interval '1' hour))
```

```
java.lang.ClassCastException: class org.apache.calcite.sql.SqlBasicCall cannot be cast
to class org.apache.calcite.sql.SqlIdentifier
    at org.apache.calcite.sql.SqlIdentifier.lambda$simpleNames$3(SqlIdentifier.java:340)
    at org.apache.calcite.util.Util$TransformingList.get(Util.java:2823)
    at java.base/java.util.AbstractList$Itr.next(AbstractList.java:373)
    at org.apache.calcite.linq4j.Ord.forEach(Ord.java:167)
    at org.apache.calcite.sql.SqlWindowTableFunction$AbstractOperandMetadata
        .validateColumnNames(SqlWindowTableFunction.java:236)
```

Fix the validator so it rejects a non-identifier DESCRIPTOR argument with a proper validation error. Make sure all affected table functions are covered and add tests.

Диалект Calcite: оконные табличные функции

TUMBLE
окна встык

HOP
окна внахлёст

SESSION
окна по паузам в данных

```
SELECT * FROM TABLE(
  HOP(TABLE orders,          → таблица заказов
      DESCRIPTOR(rowtime),   → колонка времени
      INTERVAL '2' MINUTE,   → сдвиг окна: каждые 2 минуты
      INTERVAL '5' MINUTE))  → размер окна: 5 минут
```

Запрос раскладывает заказы по окнам в 5 минут, которые сдвигаются каждые 2 минуты

Что делает запрос: пример на данных

Таблица orders

rowtime	productId	orderId
10:01	10	1
10:03	20	2

HOP: окно 5 минут, сдвиг 2 минуты

Результат: каждая строка попала в два окна

rowtime	productId	orderId	window_start	window_end
10:01	10	1	09:58	10:03
10:01	10	1	10:00	10:05
10:03	20	2	10:00	10:05
10:03	20	2	10:02	10:07

Правильно

```
SELECT * FROM TABLE(
  HOP(TABLE orders,
      DESCRIPTOR(rowtime),
      INTERVAL '2' MINUTE,
      INTERVAL '5' MINUTE))
```

в DESCRIPTOR имя колонки

С ошибкой

```
SELECT * FROM TABLE(
  HOP(TABLE orders,
      DESCRIPTOR(1),
      INTERVAL '2' MINUTE,
      INTERVAL '5' MINUTE))
```

в DESCRIPTOR индекс колонки

```
SELECT * FROM TABLE(HOP(TABLE orders, DESCRIPTOR(1),
  INTERVAL '2' MINUTE, INTERVAL '5' MINUTE))
```

Правильно

SqlValidatorException

The argument of DESCRIPTOR
must be an identifier

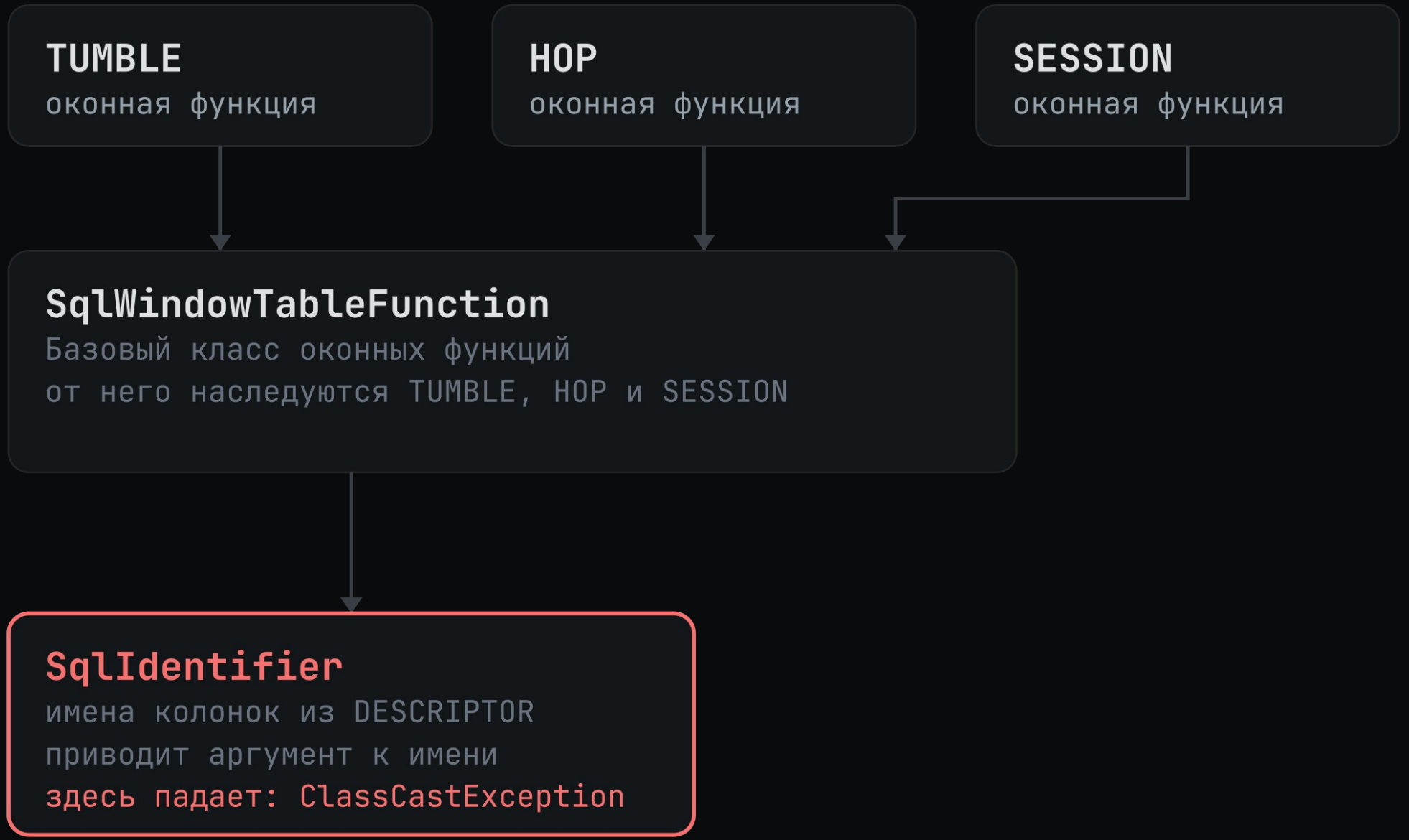
ошибка валидации запроса

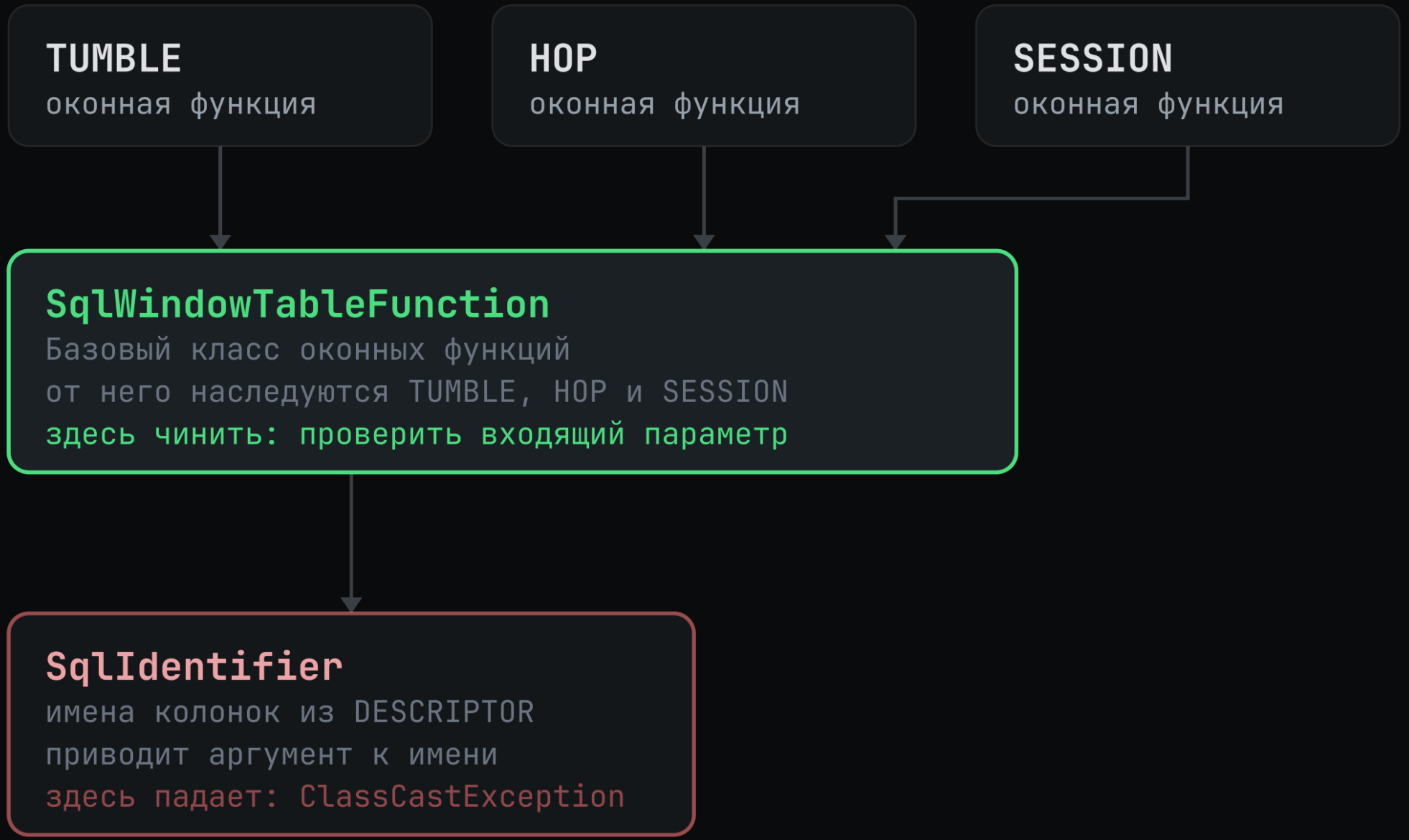
Неправильно

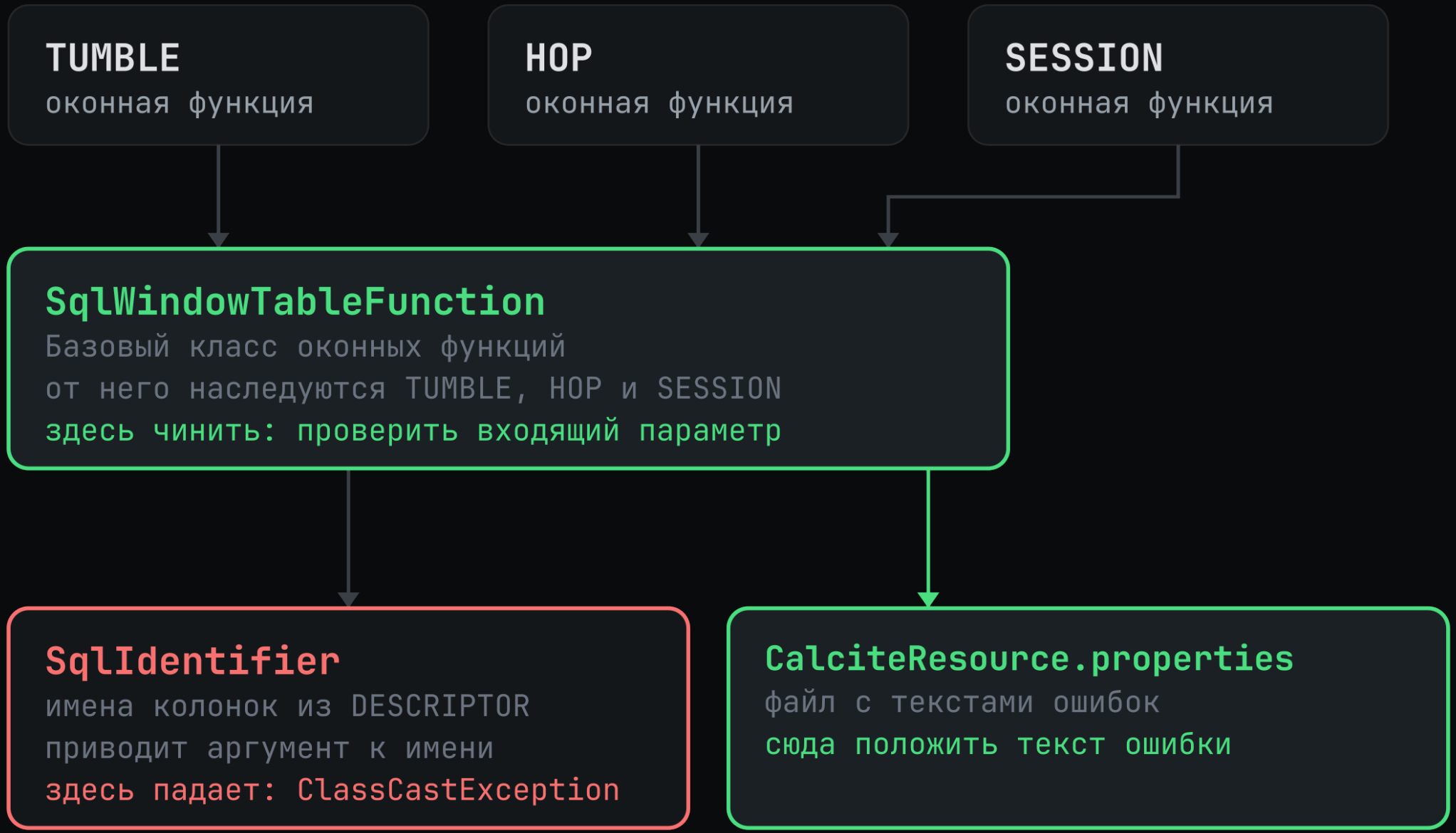
ClassCastException

class SqlNumericLiteral
cannot be cast to
class SqlIdentifier

Calcite падает целиком


TUMBLE
оконная функция
HOP
оконная функция
SESSION
оконная функция
SqlWindowTableFunction
Базовый класс оконных функций
от него наследуются TUMBLE, HOP и SESSION
SqlIdentifier
имена колонок из DESCRIPTOR
приводит аргумент к имени
здесь падает: ClassCastException


TUMBLE
оконная функция
HOP
оконная функция
SESSION
оконная функция
SqlWindowTableFunction
Базовый класс оконных функций
от него наследуются TUMBLE, HOP и SESSION
здесь чинить: проверить входящий параметр
SqlIdentifier
имена колонок из DESCRIPTOR
приводит аргумент к имени
здесь падает: ClassCastException


TUMBLE
оконная функция
HOP
оконная функция
SESSION
оконная функция
SqlWindowTableFunction
Базовый класс оконных функций
от него наследуются TUMBLE, HOP и SESSION
здесь чинить: проверить входящий параметр
SqlIdentifier
имена колонок из DESCRIPTOR
приводит аргумент к имени
здесь падает: ClassCastException
CalciteResource.properties
файл с текстами ошибок
сюда положить текст ошибки

Критерии успеха для агента

01 **Место фикса** найдено корректно: проверка имён колонок, а не место падения

02 **Конвенция ошибок** три файла согласованно, а не захардкоженный throw

03 **Три функции** TUMBLE, HOP и SESSION закрыты одним фиксом

04 **Тесты** в классе тестов валидатора, на все три функции

05 **Эталон** сверка с фиксом мейнтейнера Calcite

Бенчмарк

01 **Один исходный промпт** информация из Jira: пример запроса, стек трейс, просьба починить и покрыть тестами

02 **Одна модель** DeepSeek V4 Flash, reasoning effort: max, агент opencode

03 **Семь наборов инструментов** от чтения файлов до графа знаний

04 **Ваншотом** терминальной командой opencode run, без подсказок по ходу, лимит 25 минут на прогон

05 **Замеряем** время, стоимость, число вызовов, попадание в эталон

Стандартные тулы агента

Read и ripgrep

Текст
Read
ripgrep

Синтаксис
ast-grep

Семантика
IntelliJ IDEA по MCP
Serena + JDTLS

База знаний
Graphify
Understand-Anything

Стек трейс из баг-репорта: всё, что агент знает о падении

```
java.lang.ClassCastException: class org.apache.calcite.sql.SqlBasicCall cannot be cast
to class org.apache.calcite.sql.SqlIdentifier
    at org.apache.calcite.sql.SqlIdentifier.lambda$simpleNames$3(SqlIdentifier.java:340)
    at org.apache.calcite.util.Util$TransformingList.get(Util.java:2823)
    at java.base/java.util.AbstractList$Itr.next(AbstractList.java:373)
    at org.apache.calcite.linq4j.Ord.forEach(Ord.java:167)
    at org.apache.calcite.sql.SqlWindowTableFunction$AbstractOperandMetadata
        .validateColumnNames(SqlWindowTableFunction.java:236)
```

```
java.lang.ClassCastException: class org.apache.calcite.sql.SqlBasicCall cannot be cast

to class org.apache.calcite.sql.SqlIdentifier

    at org.apache.calcite.sql.SqlIdentifier.lambda$simpleNames$3(SqlIdentifier.java:340) (2)

    at org.apache.calcite.util.Util$TransformingList.get(Util.java:2823)

    at java.base/java.util.AbstractList$Itr.next(AbstractList.java:373)

    at org.apache.calcite.linq4j.Ord.forEach(Ord.java:167)

    at org.apache.calcite.sql.SqlWindowTableFunction$AbstractOperandMetadata

        .validateColumnNames(SqlWindowTableFunction.java:236) (1)
```

(1) 1-й вызов в 100% прогонов

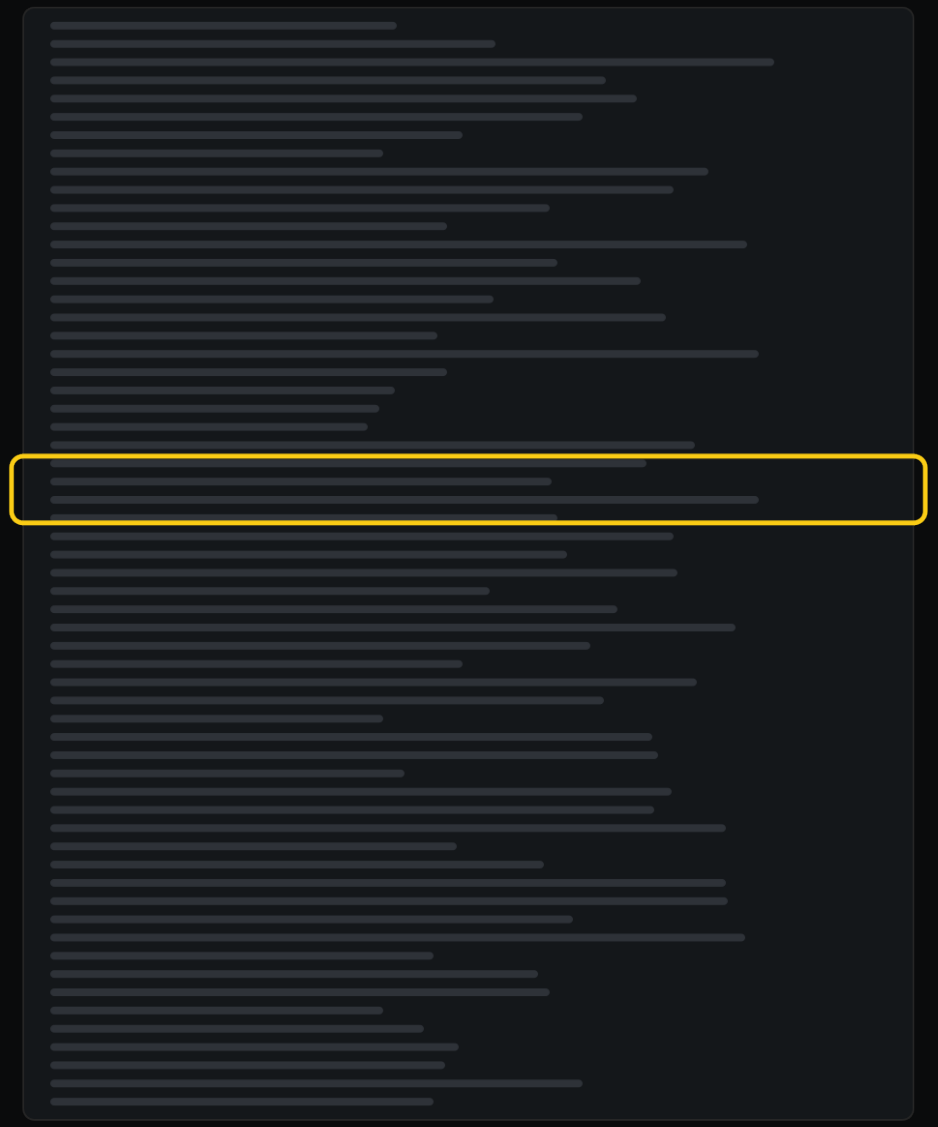
read SqlWindowTableFunction.java
файл целиком

(2) 2-й вызов в 80% прогонов

read SqlIdentifier.java
offset 300, limit 80

Имена файлов агент берёт прямо из стека

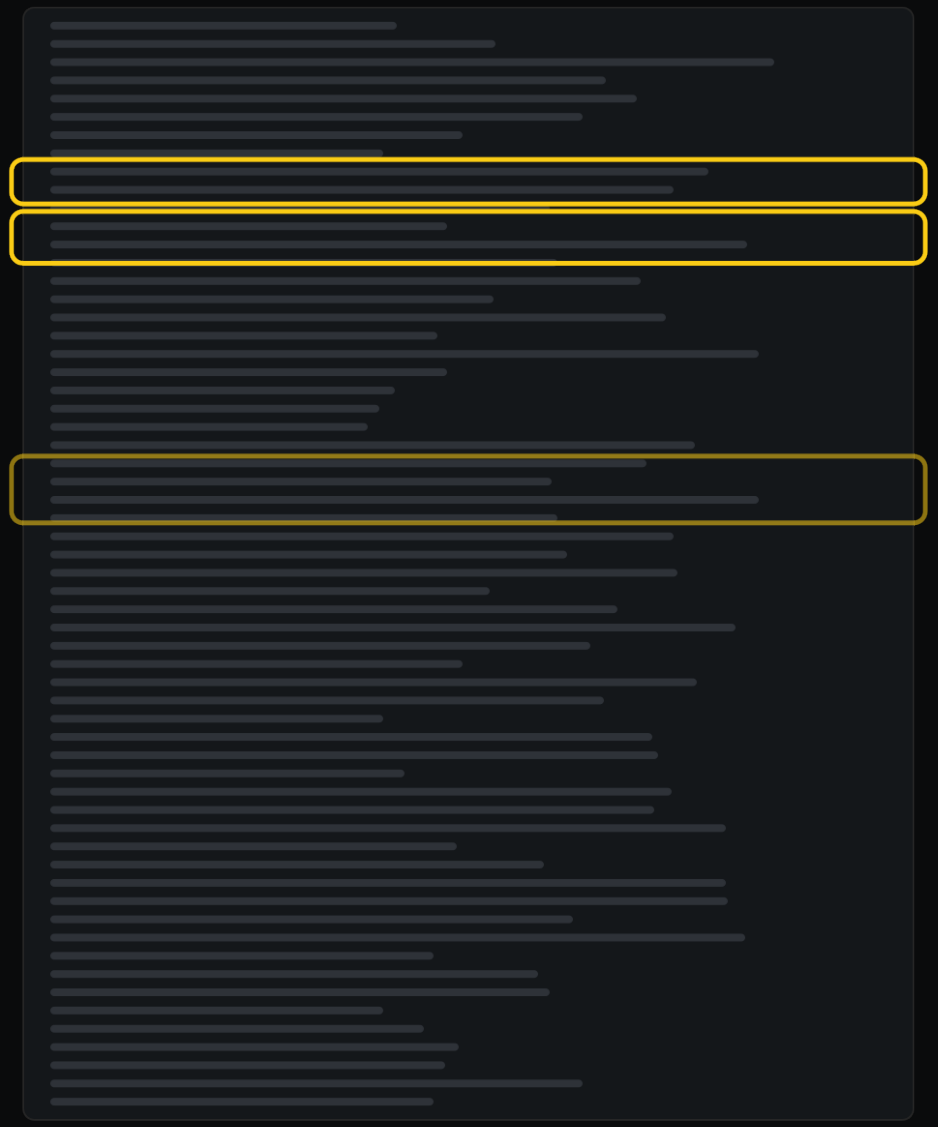
SqlValidatorTest.java: строки 11 550-11 850 из 13 777



Чтение 1: с 11 671-й строки

18 строк

SqlValidatorTest.java: строки 11 550-11 850 из 13 777



Чтение 1: с 11 671-й строки

18 строк

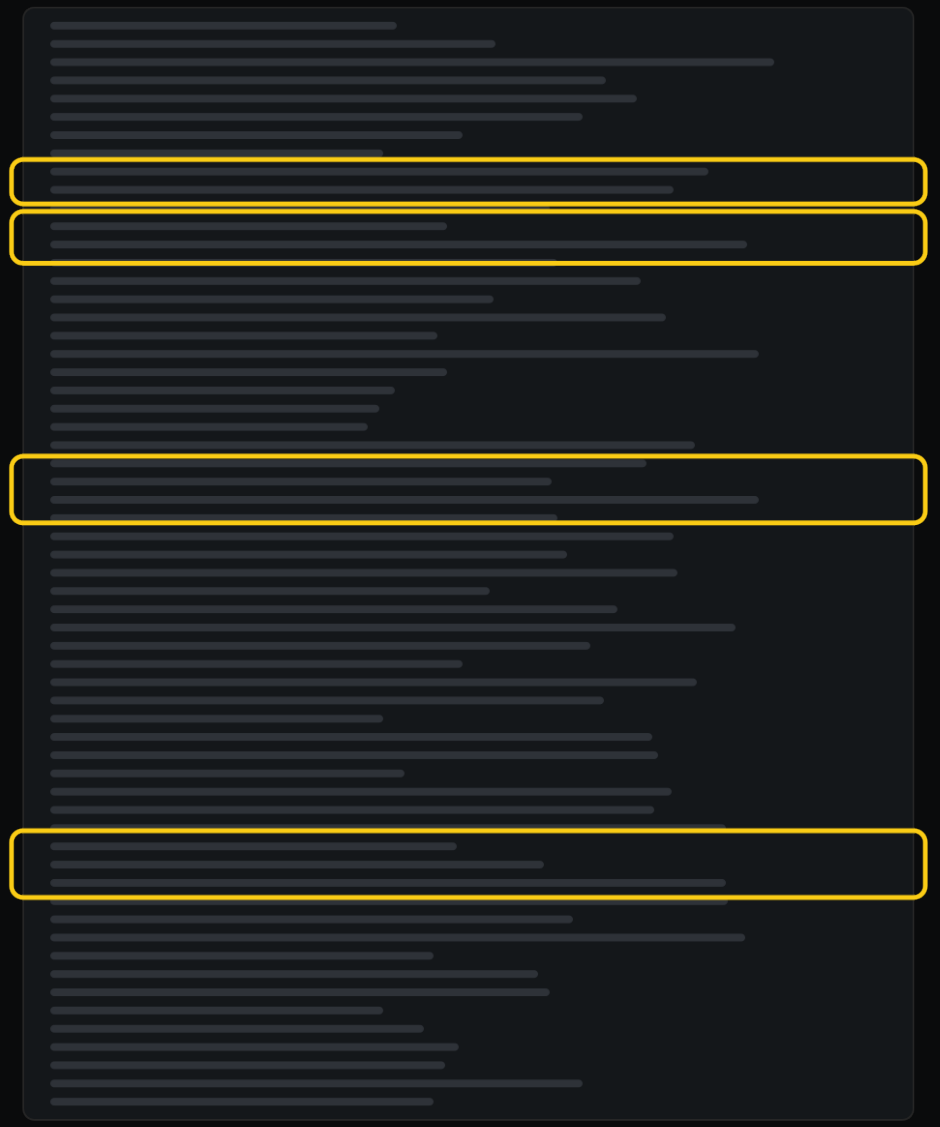
Чтение 2: с 11 605-й строки

14 строк, назад на 66

Чтение 3: с 11 591-й строки

12 строк, ещё назад на 14

SqlValidatorTest.java: строки 11 550-11 850 из 13 777



Чтение 1: с 11 671-й строки

18 строк

Чтение 2: с 11 605-й строки

14 строк, назад на 66

Чтение 3: с 11 591-й строки

12 строк, ещё назад на 14

Чтение 4: с 11 772-й строки

18 строк, вперёд на 181

Read tool: выводы

01 **Место падения находит быстро** по именам классов из стек трейса

02 **Наследников ищет перебором** от 33 до 85 поисков файлов по маскам за прогон

03 **Конвенцию находит в 40% прогонов** текст ошибки в .properties попадается случайно

04 **Зовёт поиск, которого ему не давали** модель пытается вызвать grep, хотя это запрещено, 2-3 раза за прогон

05 **Не укладывается во время** в 20% прогонов агент превысил заданное ограничение по времени

Read tool

21.7 мин

время

$0.229

стоимость

DeepSeek V4 Flash

324K + 10.8M

кэш

токены

145

tool-calls

вызовов на прогон

100%

место фикса

соответствие мейнтейнеру

100%

тесты

добавлены в нужный тестовый класс

40%

код в стиле проекта

сообщение об ошибке по конвенции Calcite

grep -r

один поток, файл за файлом

core/src/main/java
исходники

core/build
результат сборки, в .gitignore
читает целиком

.git
история репозитория
читает целиком

***.jar, *.class**
бинарные файлы
читает целиком

ripgrep

несколько потоков параллельно

core/src/main/java
исходники

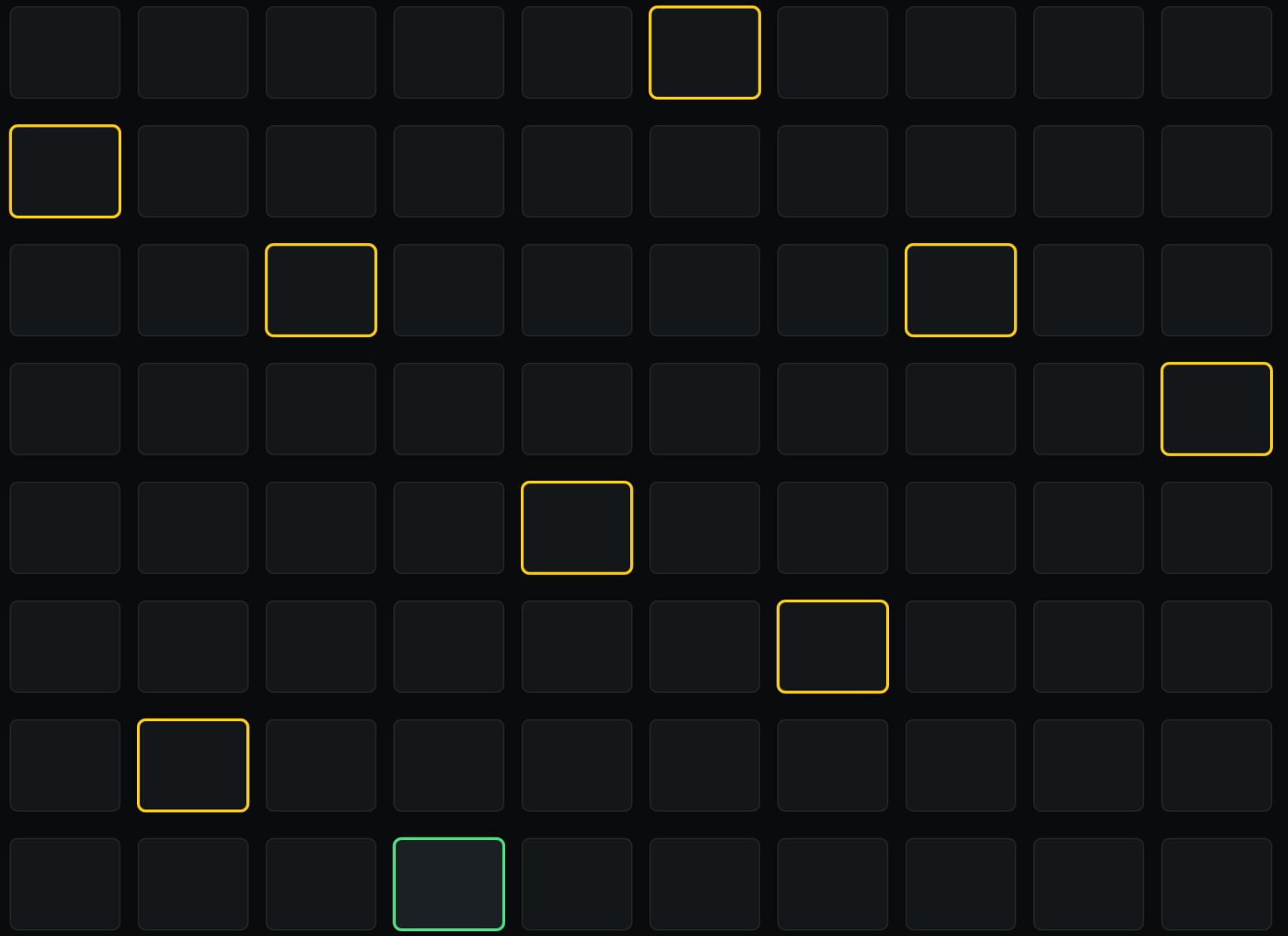
core/build
результат сборки, в .gitignore
пропущено: .gitignore

.git
история репозитория
пропущено: служебный каталог

***.jar, *.class**
бинарные файлы
пропущено: бинарный файл



492 Java-файла
каталога sql



492 Java-файла
каталога sql

9 файлов,
29 совпадений

1 нужный

```
rg --glob='*.java' 'DESCRIPTOR|Descriptor' core/src/main/java/org/apache/calcite/sql
```

SqlWindowTableFunction.java

```
171  boolean checkTableAndDescriptorOperands(SqlCallBinding…
181  if (operand.getKind() != SqlKind.DESCRIPTOR) {
197  boolean checkTimeColumnDescriptorOperand(SqlCallBinding…
```

SqlHopTableFunction.java

```
48   if (!checkTableAndDescriptorOperands(callBinding, 1)) {
51   if (!checkTimeColumnDescriptorOperand(callBinding, 1)) {
61   return opName + "(TABLE table_name, DESCRIPTOR(timecol), "
```

SqlSessionTableFunction.java

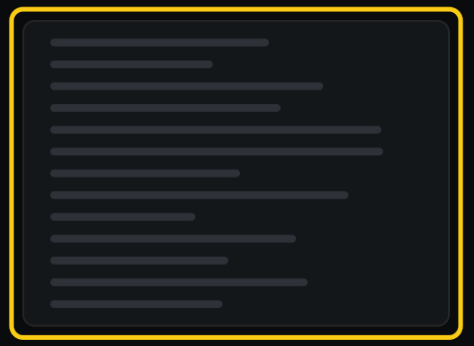
```
51   if (!checkTableAndDescriptorOperands(callBinding, 1)) {
54   if (!checkTimeColumnDescriptorOperand(callBinding, 1)) {
61   if (operand2.getKind() == SqlKind.DESCRIPTOR) {
```

SqlDescriptorOperator.java

```
28   * DESCRIPTOR appears as an argument in a function. …
36   public class SqlDescriptorOperator extends SqlOperator {
38   super("DESCRIPTOR",
```

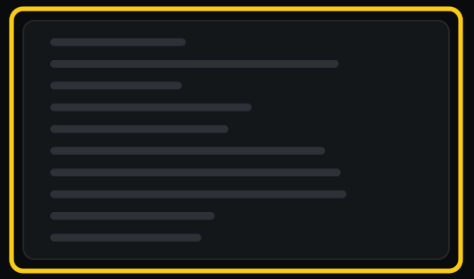
Найдено 29 совпадений в 9 файлах

SqlSessionTableFunction.java



Read: весь файл
83 строки

SqlHopTableFunction.java



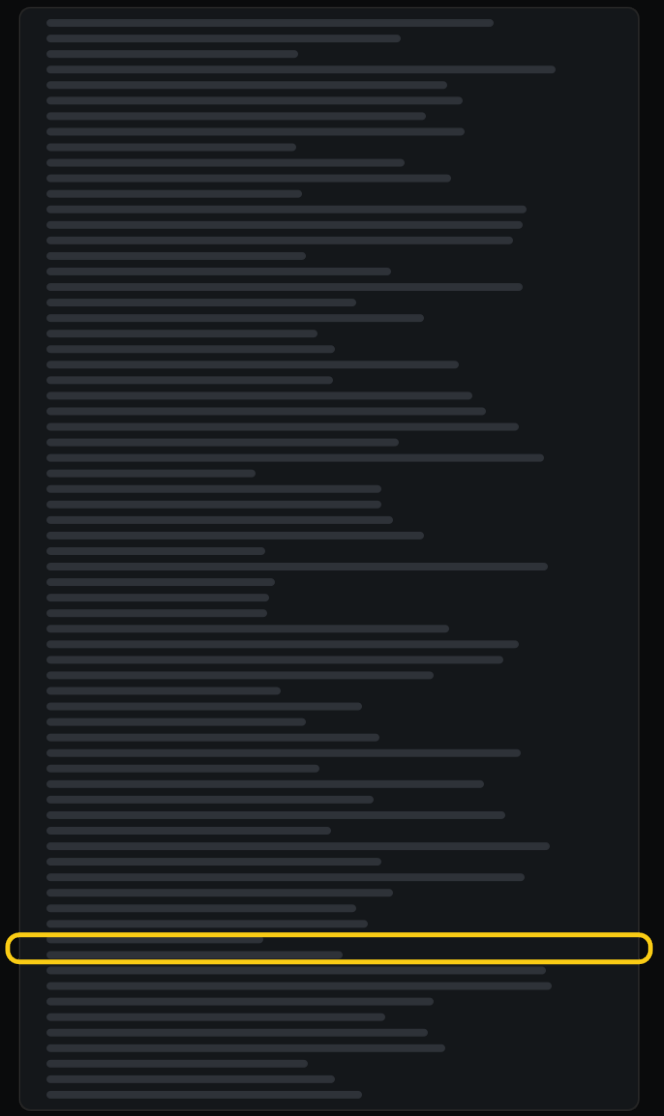
Read: весь файл
65 строк

SqlTumbleTableFunction.java



Read: весь файл
66 строк

SqlValidatorTest.java



Read: окно 240 строк
с 11 585-й из 13 777 строк

ripgrep: выводы

01 **Наследников находит одним запросом** все три функции одним поиском

02 **Поиск эффективнее перебора файлов** 3-15 поисков по маскам вместо 33-85, чтений 30-56 вместо 36-87

03 **Укладывается во время** 100% прогонов завершились сами

04 **Конвенцию находит в 60% прогонов** только если догадается искать по тексту ошибки

05 **Ищет маску со словами, а не смысл** поиск по маске дал 9 файлов, что в них, агент выясняет обычным чтением

ripgrep

16.7 мин
время

$0.145
стоимость
DeepSeek V4 Flash

244K + 6.9M
кэш
токены

109
tool-calls
вызовов на прогон

100%
место фикса
соответствие мейнтейнеру

100%
тесты
добавлены в нужный тестовый класс

60%
код в стиле проекта
сообщение об ошибке по конвенции Calcite

Работа с текстом

01 **Ищет слова, а не смысл** ищет текст, а не управляет информацией и её смысловой нагрузкой

02 **Одно слово даёт лишние файлы** по всему проекту слово descriptor находит 23 файла, нужный из них один

03 **Не понимает классов и иерархии** наследников и вызовы восстанавливает по буквам

04 **В эталон попадает случайно** конвенцию ошибок находит в 40-60% прогонов

ast-grep

Текст
Read
ripgrep

Синтаксис
ast-grep

Семантика
IntelliJ IDEA по MCP
Serena + JDTLS

База знаний
Graphify
Understand-Anything

SG
AST-GREP

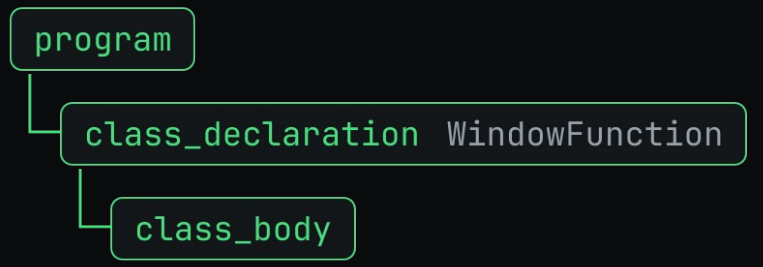
Структурный поиск и замена кода: паттерн пишется как обычный код, а совпадения ищутся по дереву синтаксиса, а не по тексту.

Внутри: tree-sitter
парсер, который строит дерево синтаксиса для десятков языков, в том числе Java



github.com/ast-grep/ast-grep

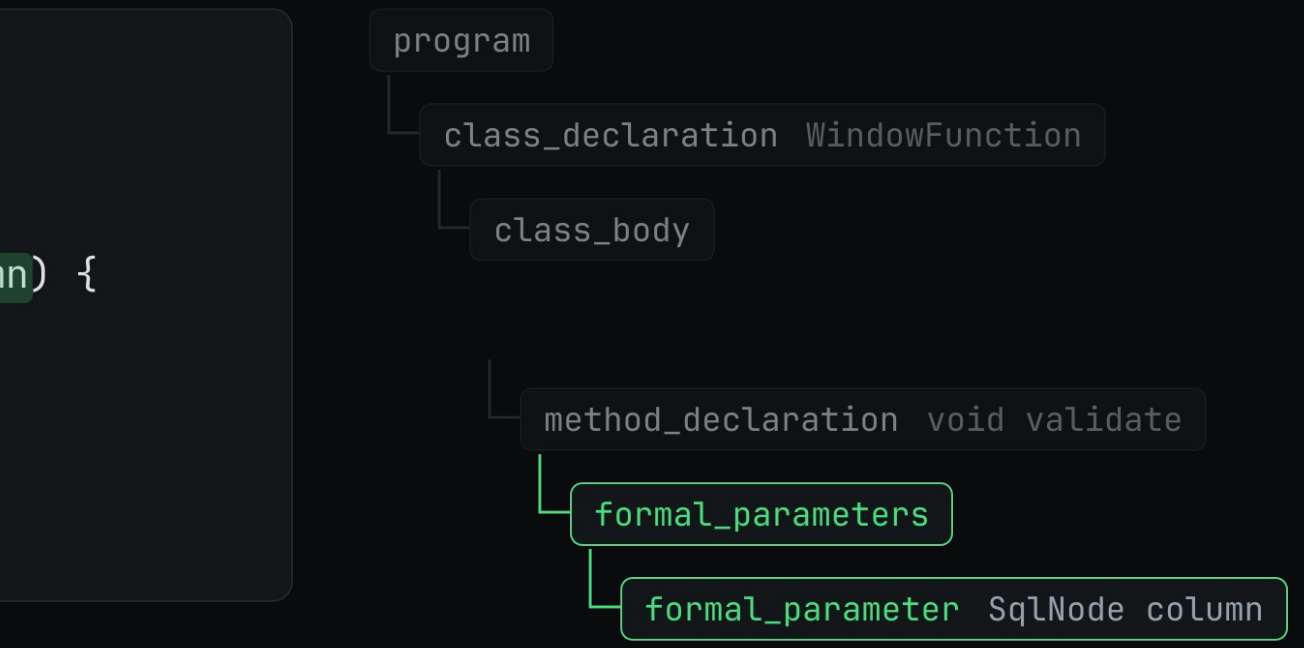
```
class WindowFunction {
  // проверить имя колонки
  void validate(SqlNode column) {
    check(column, 1);
  }
}
```



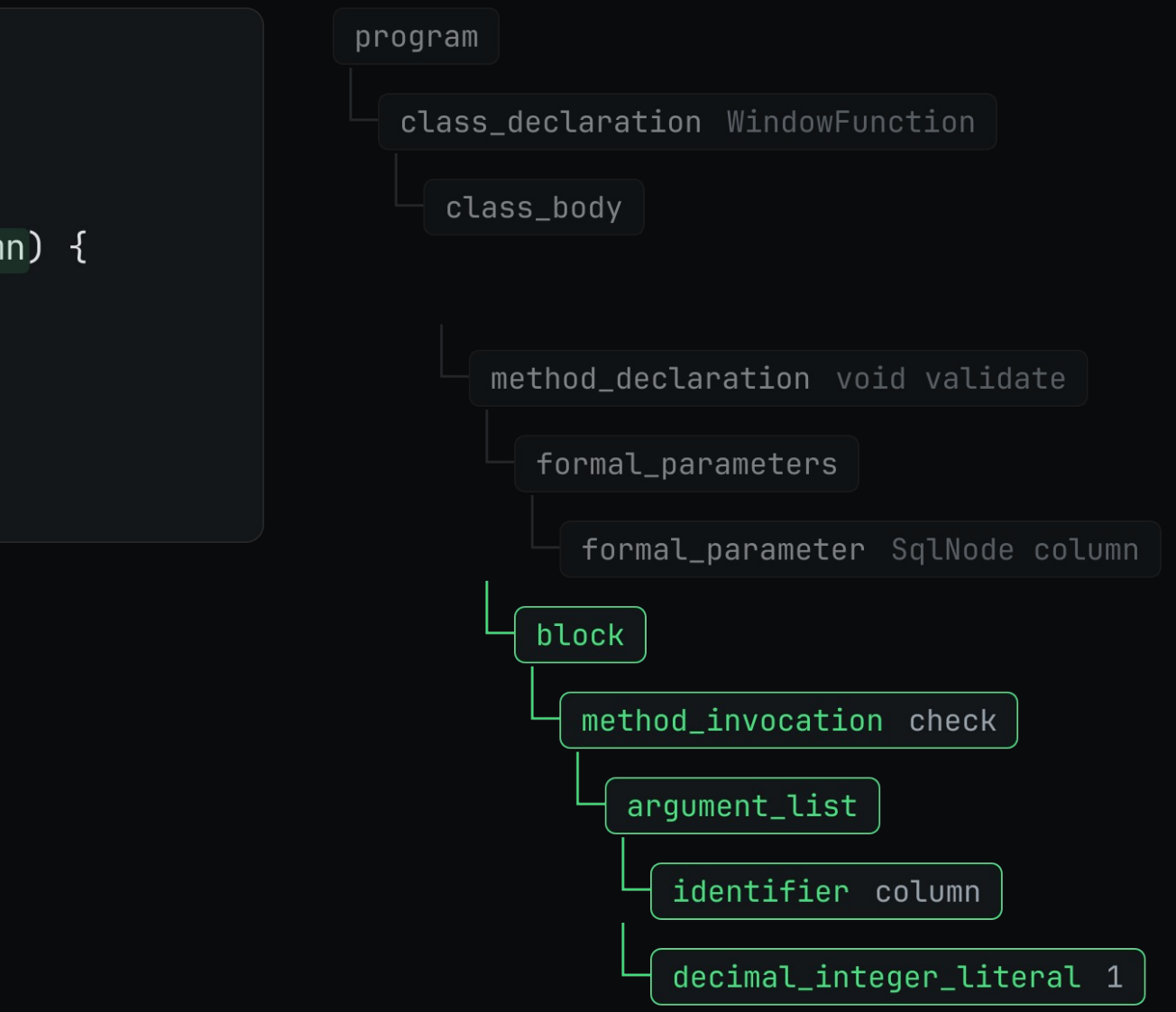
```
class WindowFunction {
  // проверить имя колонки
  void validate(SqlNode column) {
    check(column, 1);
  }
}
```



```
class WindowFunction {
  // проверить имя колонки
  void validate(SqlNode column) {
    check(column, 1);
  }
}
```

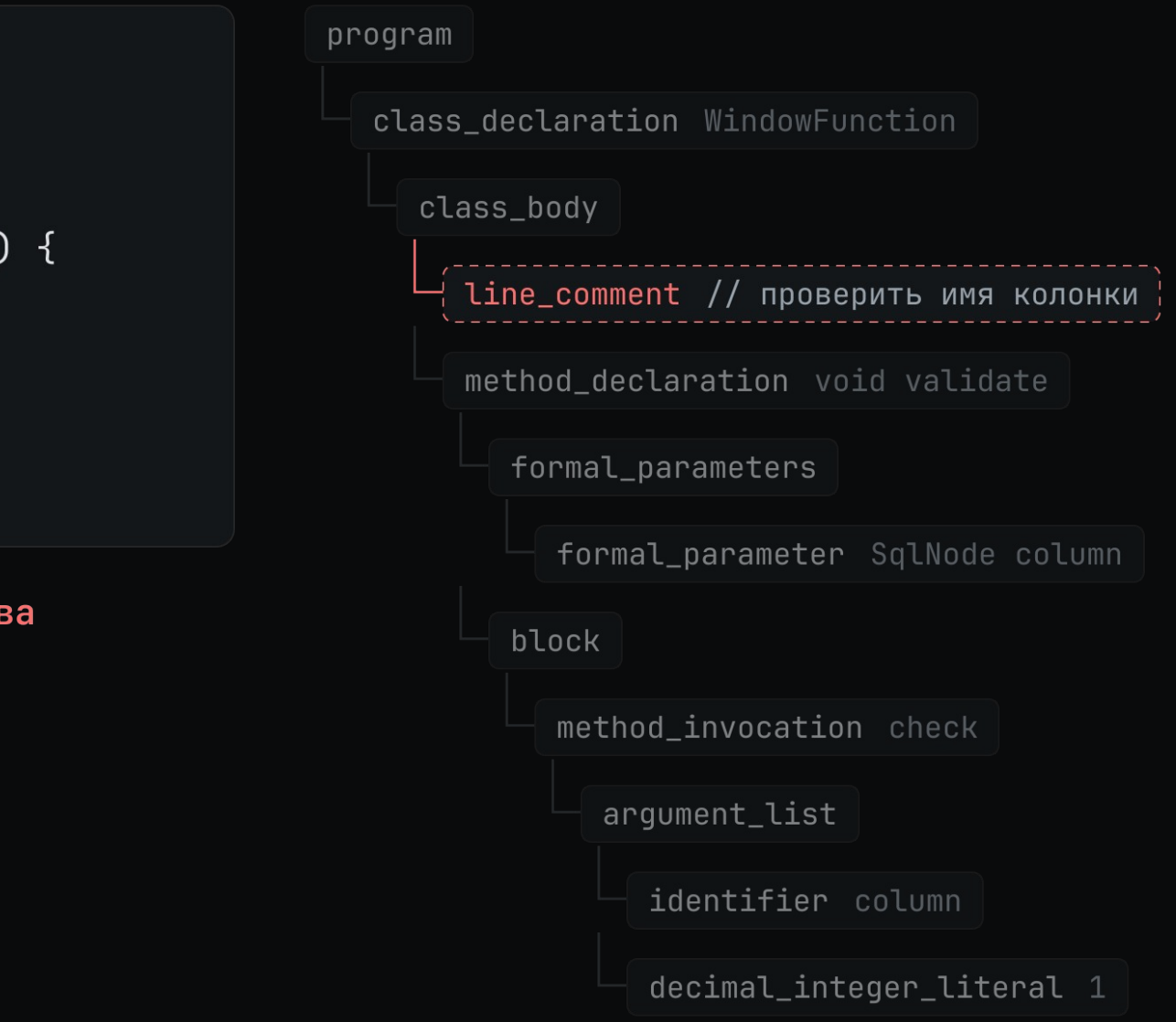


```
class WindowFunction {
  // проверить имя колонки
  void validate(SqlNode column) {
    check(column, 1);
  }
}
```



```
class WindowFunction {
  // проверить имя колонки
  void validate(SqlNode column) {
    check(column, 1);
  }
}
```

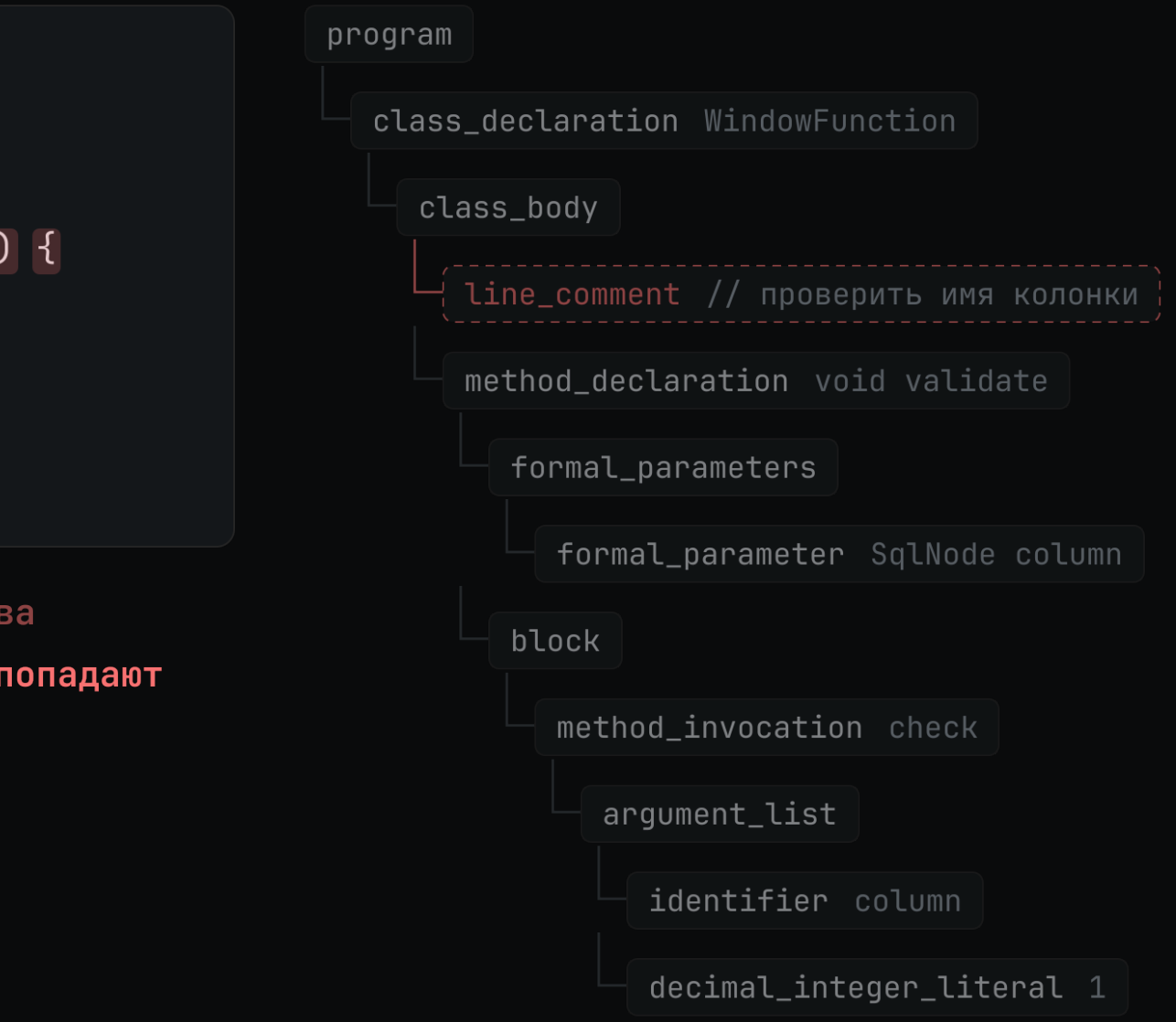
комментарий: отдельный узел дерева



```
class WindowFunction {
  // проверить имя колонки
  void validate(SqlNode column) {
    check(column, 1);
  }
}
```

комментарий: отдельный узел дерева

скобки, запятые и ; в дерево не попадают



```
class WindowFunction {
  // проверить имя колонки
  void validate(SqlNode column) {
    check(column, 1);
  }
}
```

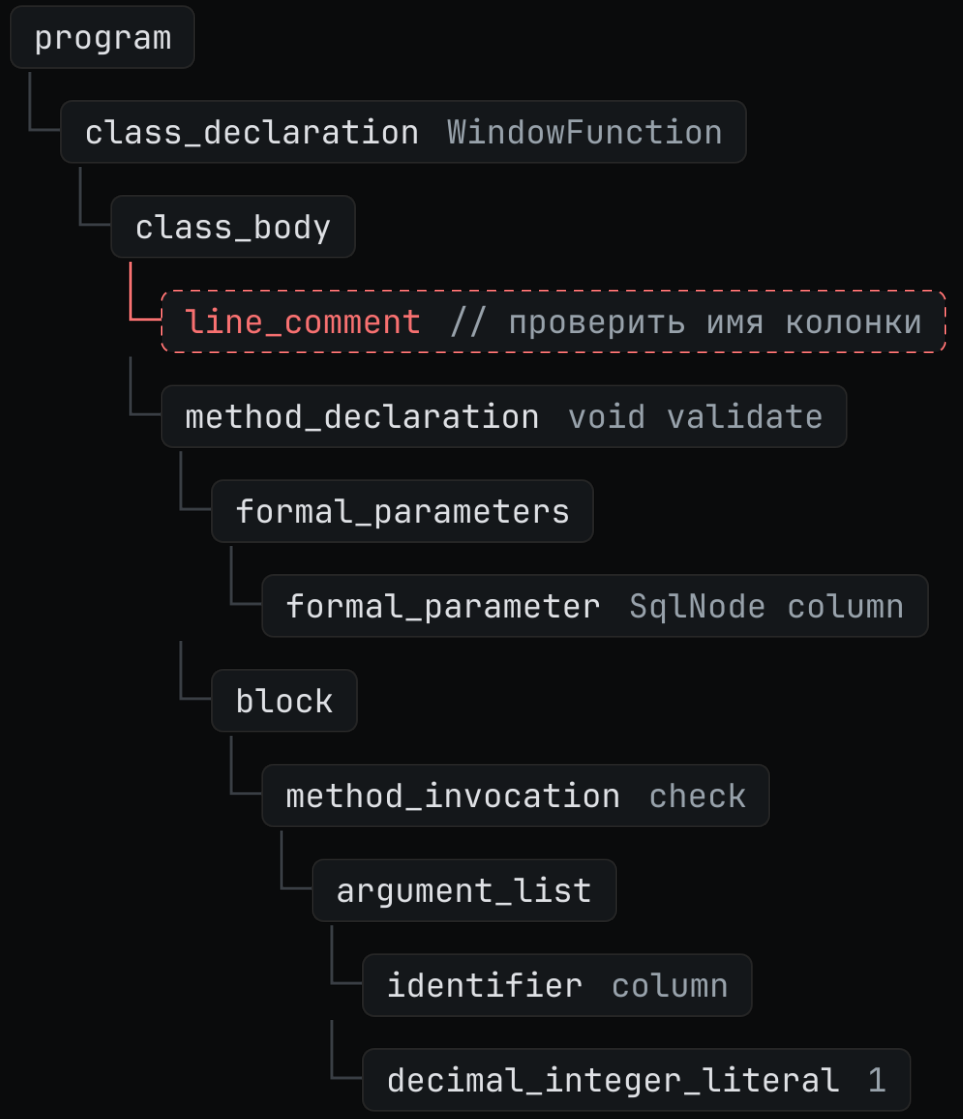
комментарий: отдельный узел дерева

скобки, запятые и ; в дерево не попадают

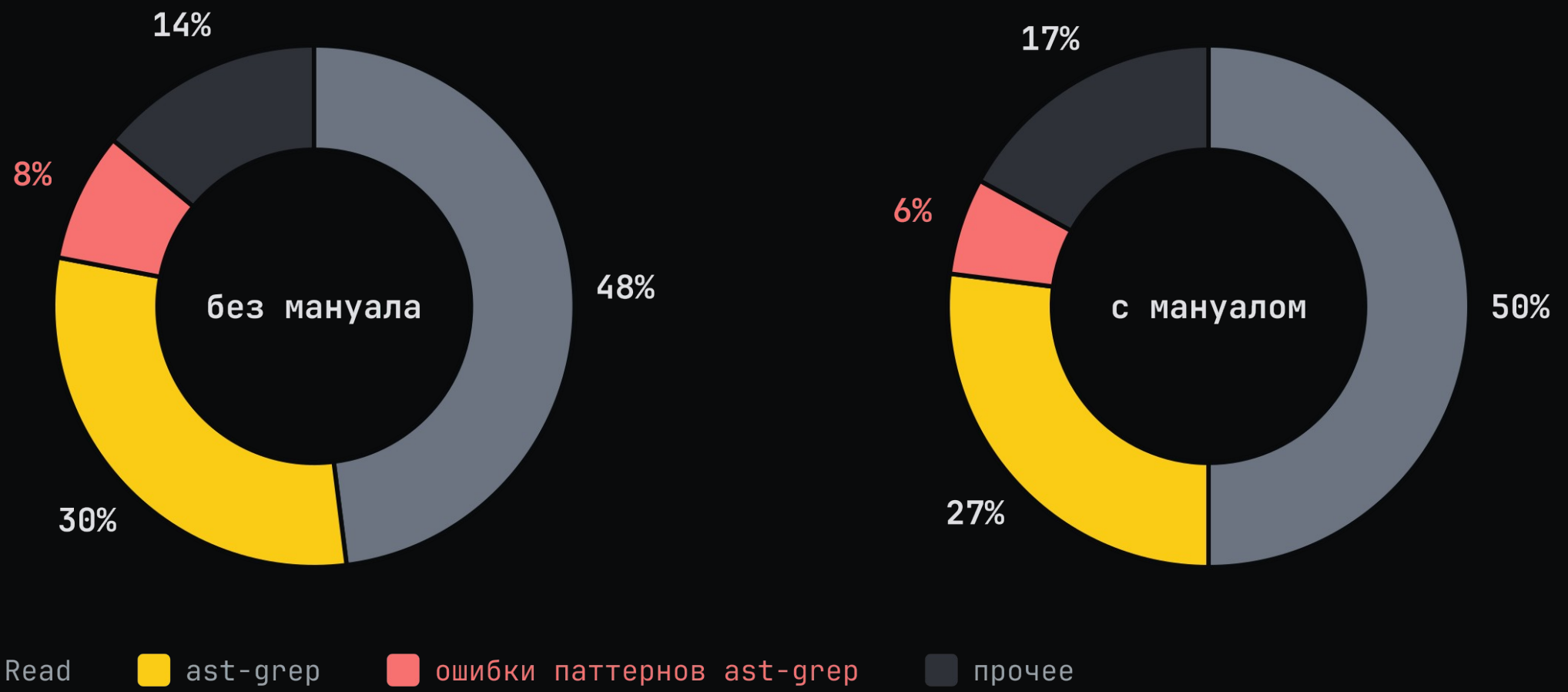
```
class $C { void $M($T $P) { $$$B } }
```

паттерн без комментария

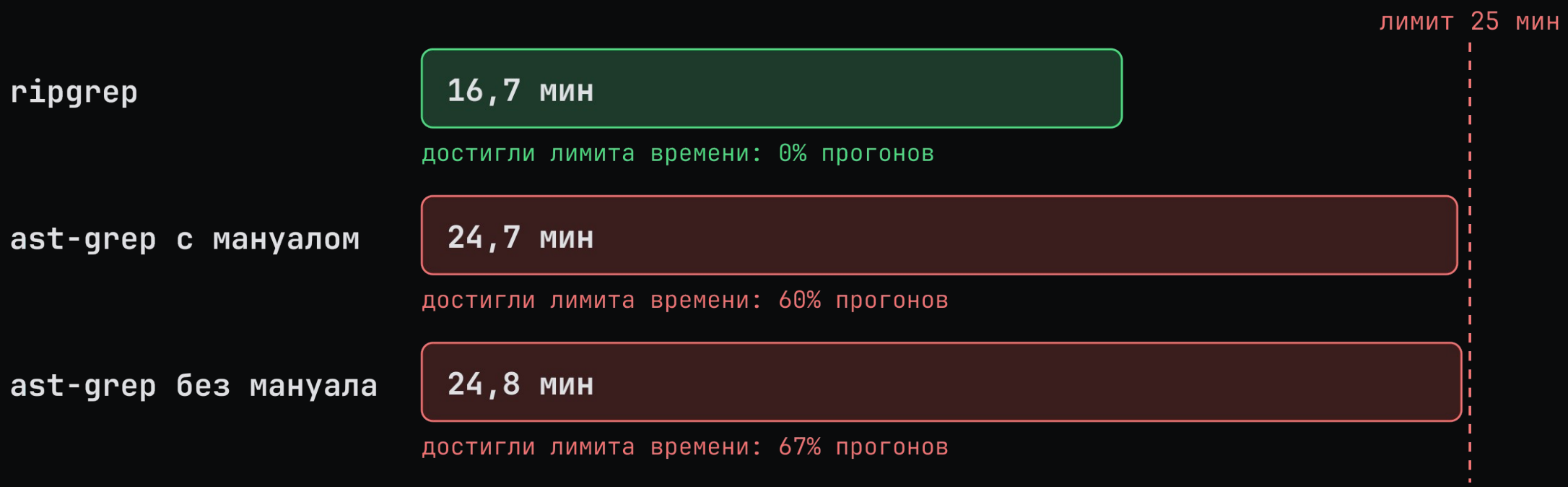
этот класс не находит



Куда уходит время прогона с ast-grep



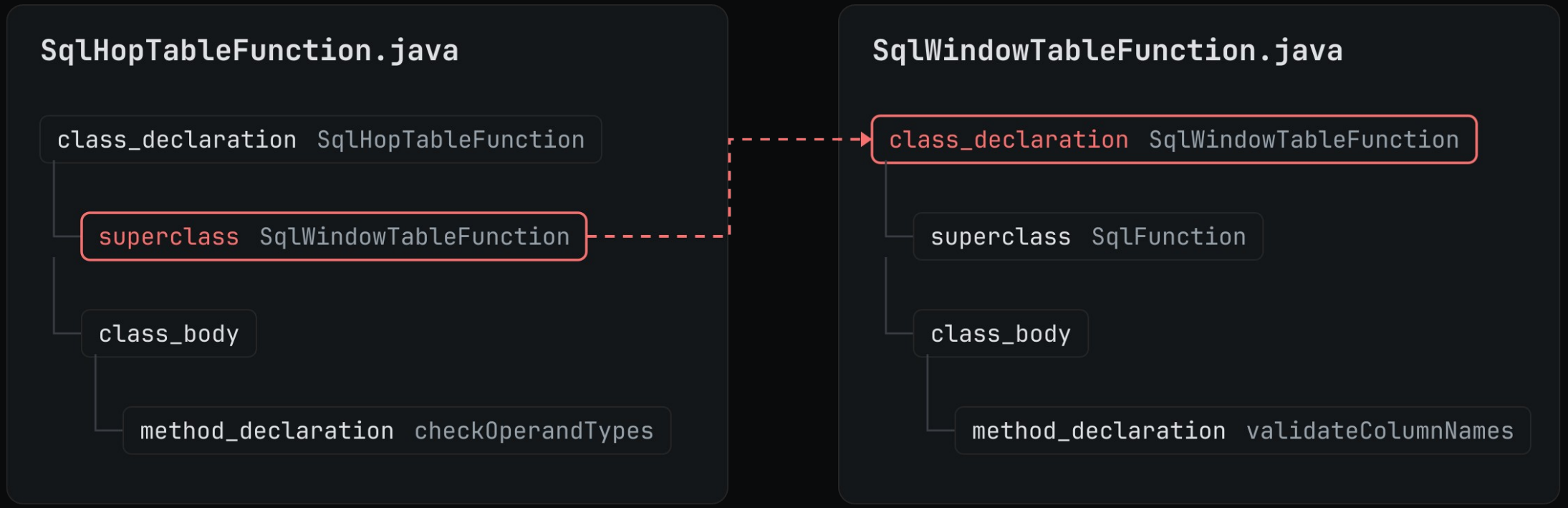
С ast-grep агенту обязательно нужен скилл: как писать его паттерны



01 Некорректные паттерны для Java: каждый 4-й вызов впустую

02 Специфичные .properties вне синтаксиса Java: агент их читает

03 Нет руководства по паттернам: 14 вызовов с несуществующими флагами, 33 попытки на файлы вне Java


SqlHopTableFunction.java
class_declaration SqlHopTableFunction
superclass SqlWindowTableFunction
class_body
method_declaration checkOperandTypes
SqlWindowTableFunction.java
class_declaration SqlWindowTableFunction
superclass SqlFunction
class_body
method_declaration validateColumnNames

ast-grep

24.7 мин

время

$0.166
стоимость
DeepSeek V4 Flash

512K + 6.0M
кэш
токены

144
tool-calls
вызовов на прогон

100%
место фикса
соответствие мейнтейнеру

60%
тесты
добавлены в нужный тестовый класс

20%
код в стиле проекта
сообщение об ошибке по конвенции Calcite

05 / 07

IDE и языковой сервер

Три популярных решения

IntelliJ IDEA
PSI

Eclipse
JDT

VS Code
JDTLS

IntelliJ IDEA

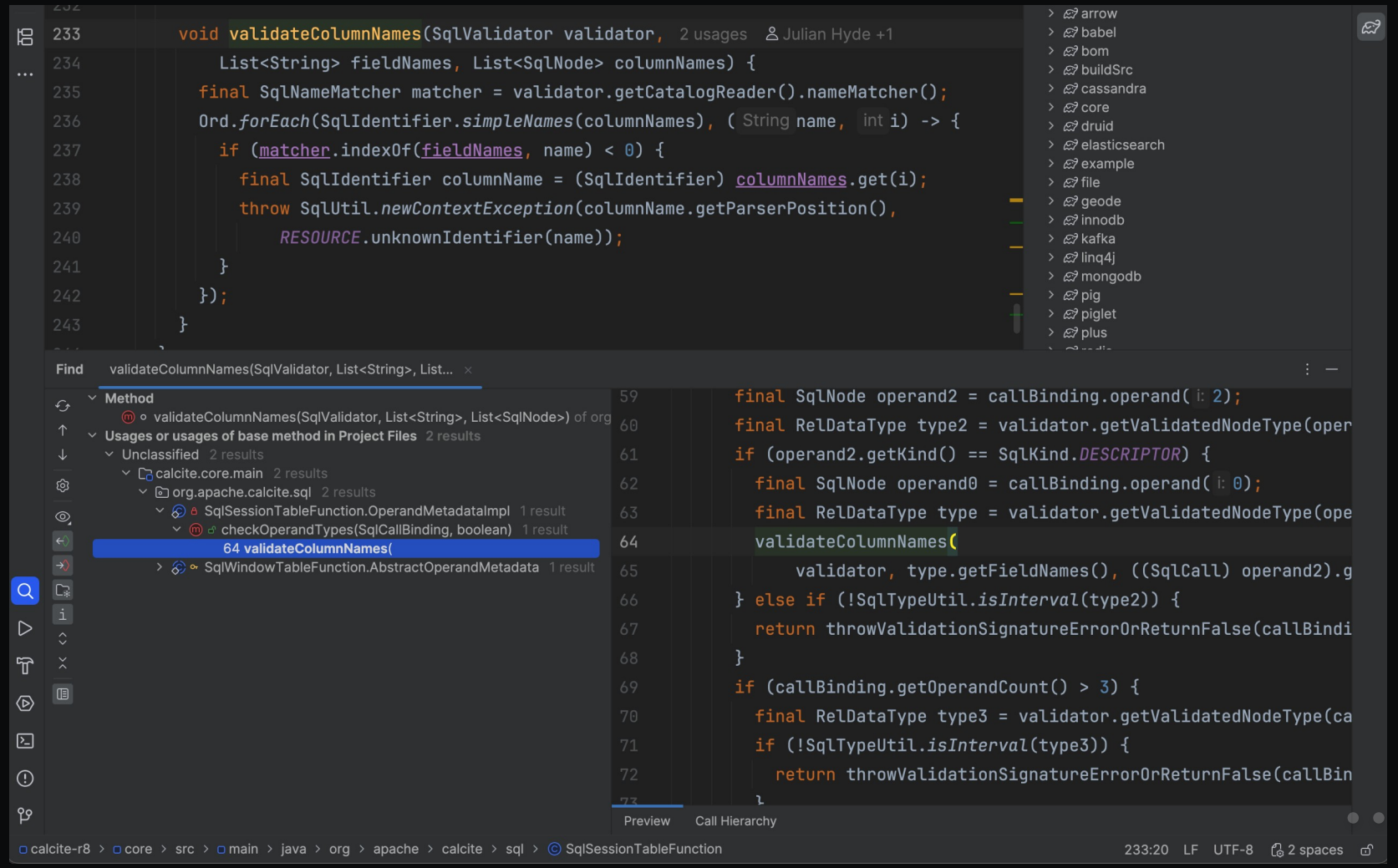
IDE от JetBrains для Java и Kotlin.
Открытая часть: IntelliJ Platform,
репозиторий intellij-community.

github.com/JetBrains/intellij-community

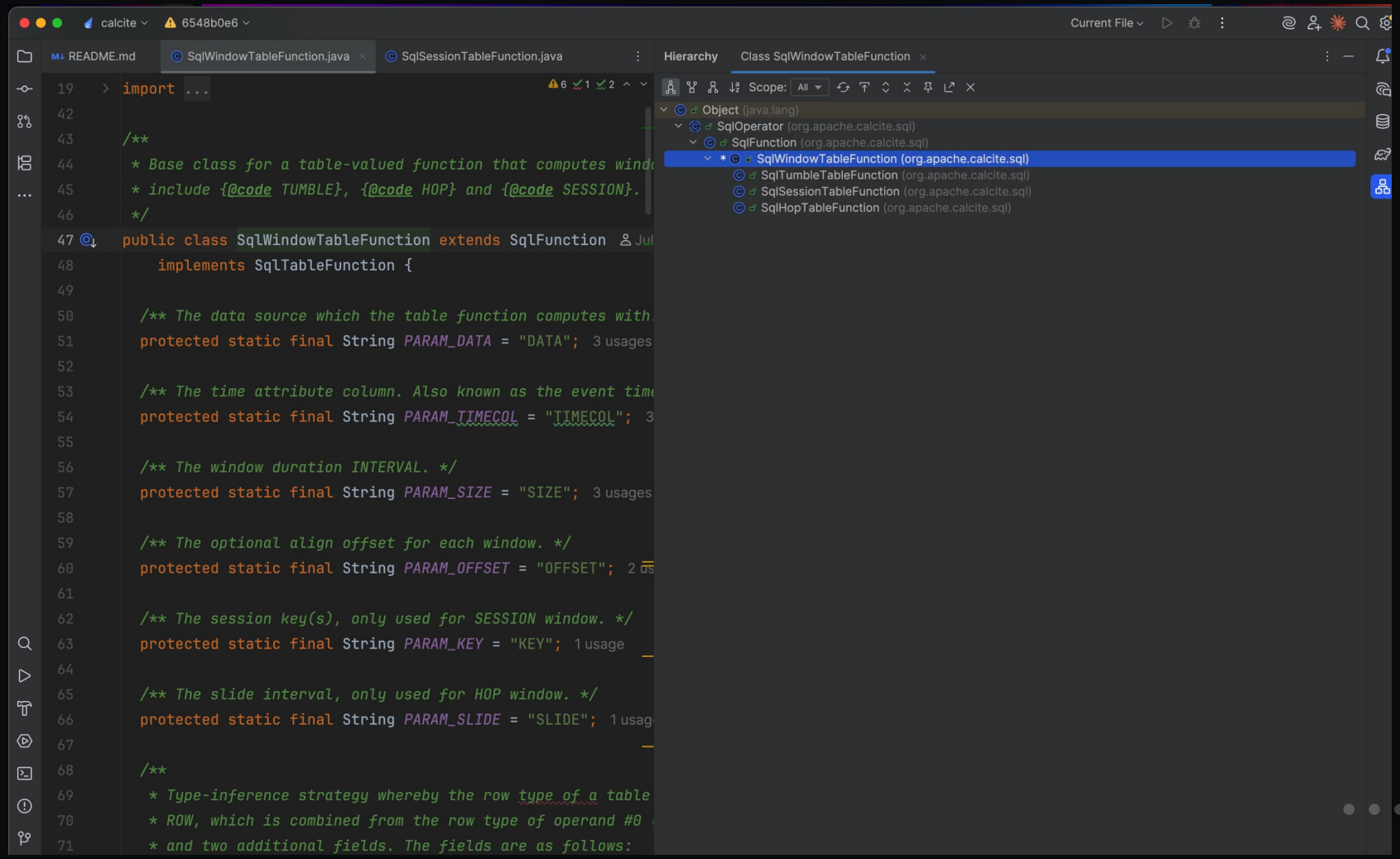
Внутри: PSI и индексы
дерево кода и связи между файлами,
с версии 2025.2 встроенный MCP-сервер

PSI | индексы | инспекции | MCP-сервер

Найти использования: ⌥F7



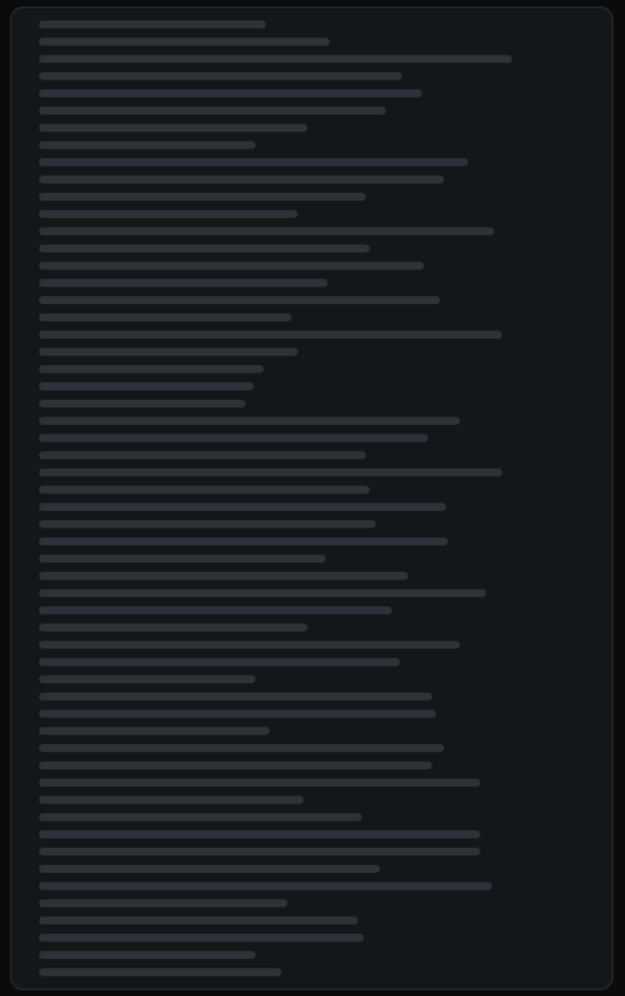
Иерархия типов: ^H



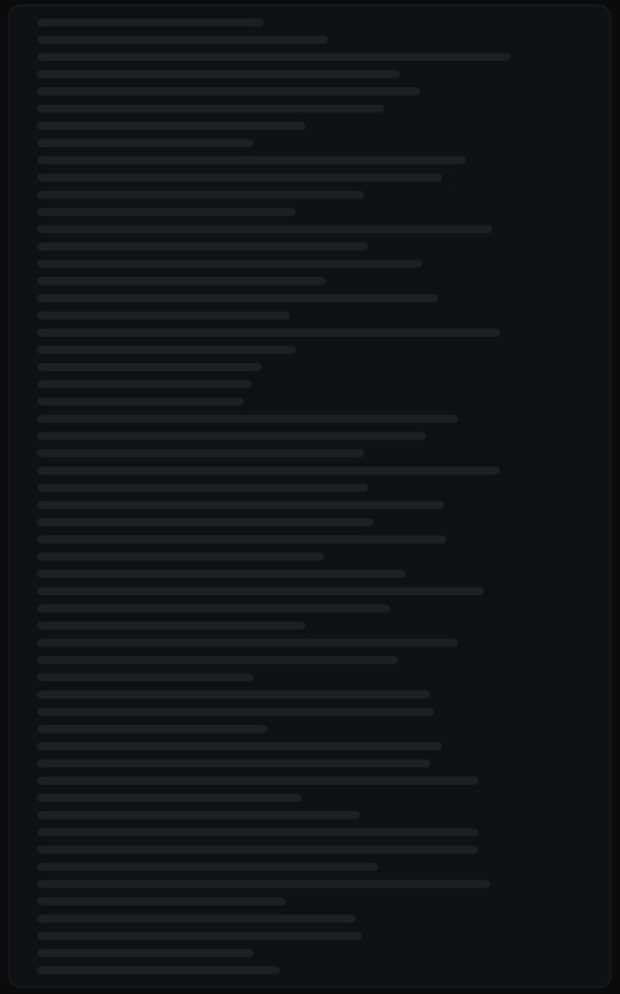
Что это за магия?

Ctrl+Click

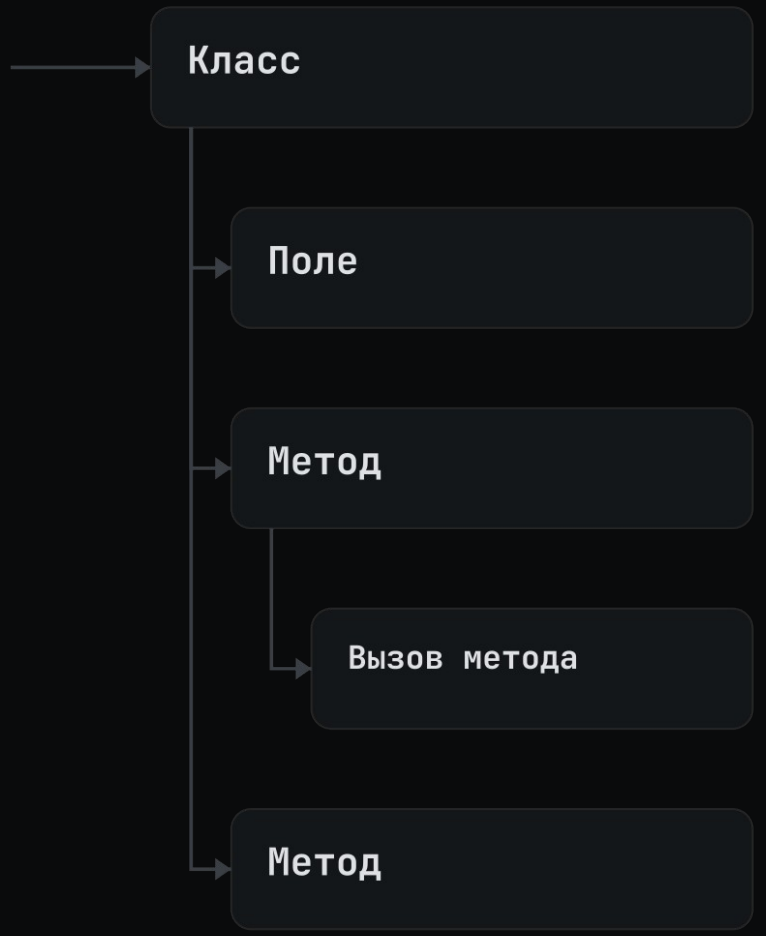
Текст класса



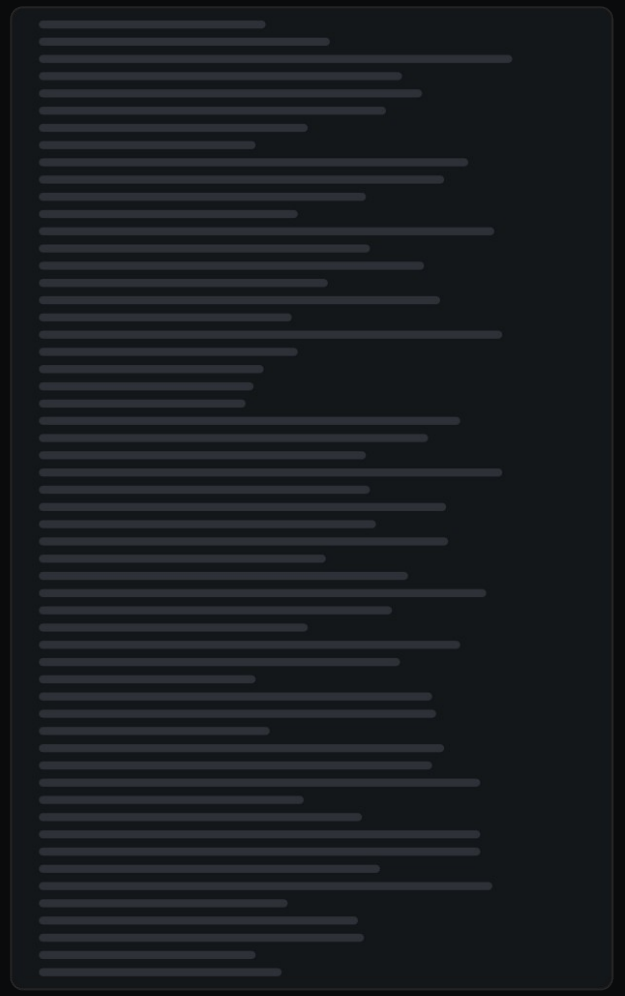
Текст класса



Дерево конструкций



Текст класса



Дерево конструкций

- Класс
  - Поле
  - Метод
    - Вызов метода
  - Метод

Связи между файлами

Вызовы
кто кого зовёт

Наследование
кто чей потомок

Индекс
держит связи в памяти
обновляется на лету

А что ещё есть у разработчика?

01 **Компилятор** собирает проект и сразу ловит ошибки

02 **Знание конвенций проекта,** не всё записано в CONTRIBUTING.md

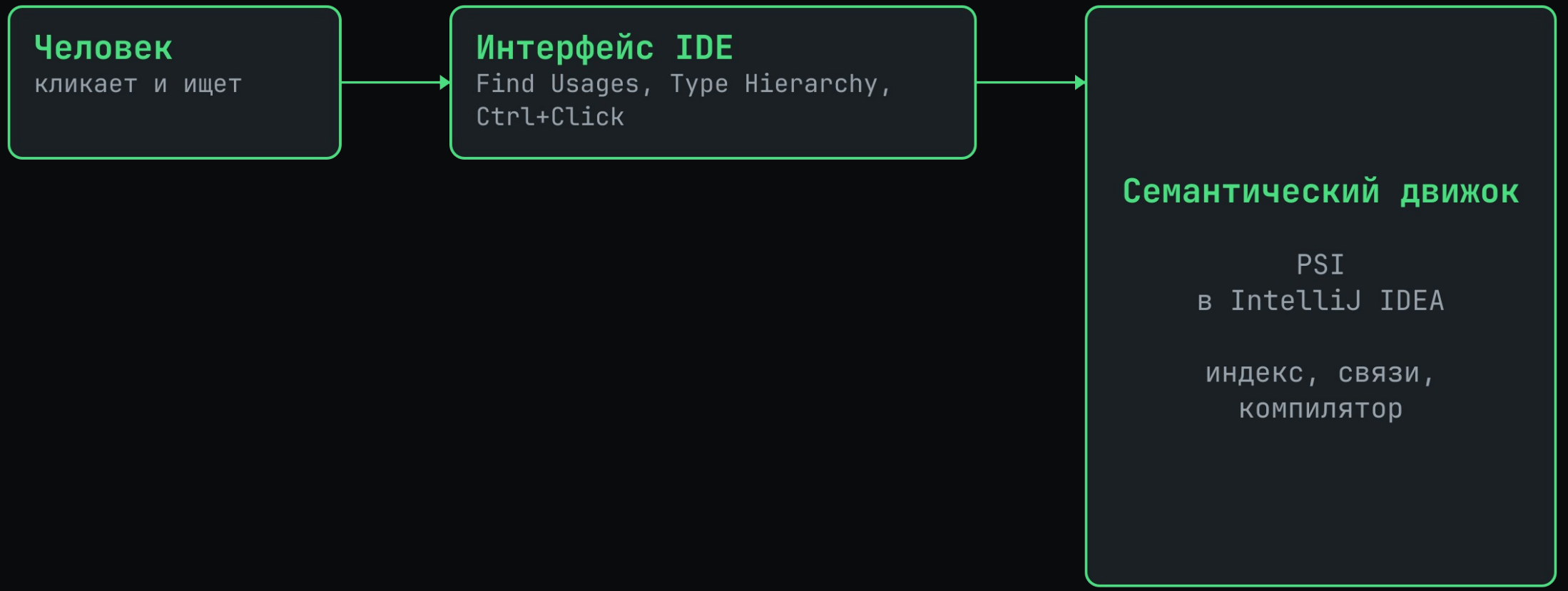
03 **Знание неочевидных вещей и связей** персонализированный опыт разработчика

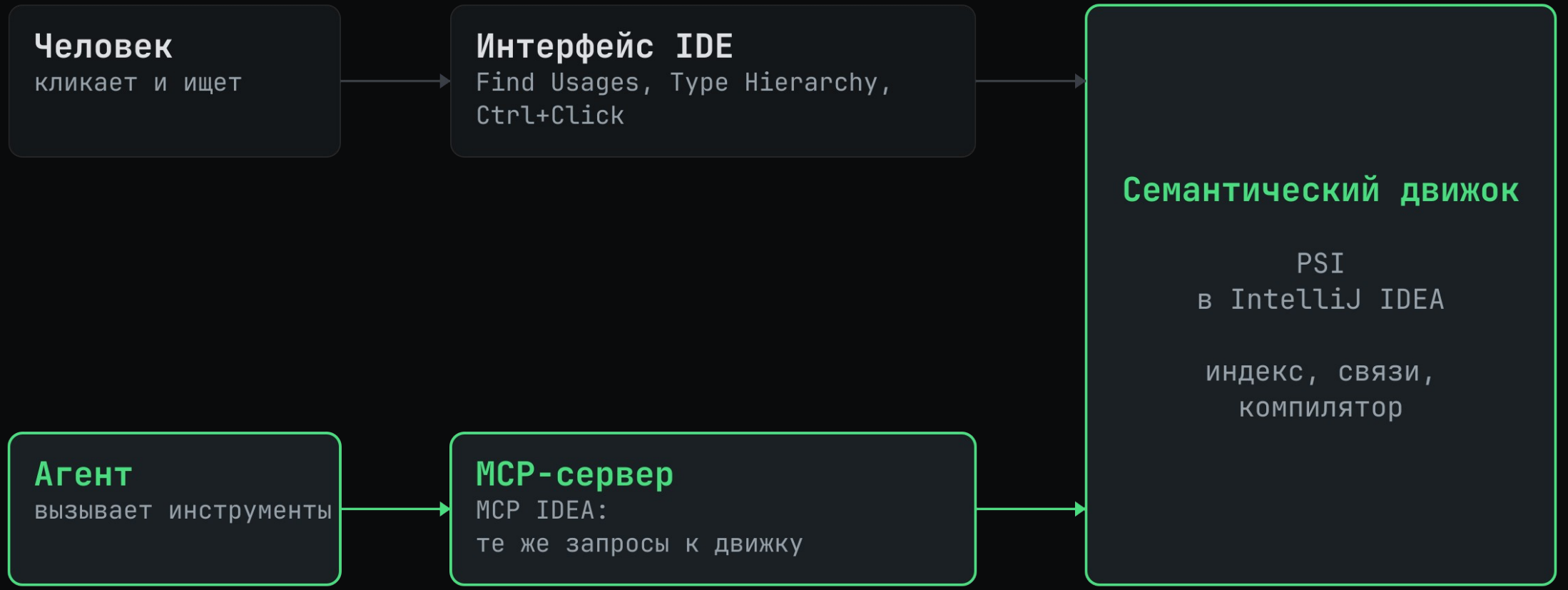
Текст
Read
ripgrep

Синтаксис
ast-grep

Семантика
IntelliJ IDEA по MCP
Serena + JDTLS

База знаний
Graphify
Understand-Anything


Человек
кликает и ищет
Интерфейс IDE
Find Usages, Type Hierarchy,
Ctrl+Click
Семантический движок
PSI
в IntelliJ IDEA
индекс, связи,
компилятор


Человек
кликает и ищет
Интерфейс IDE
Find Usages, Type Hierarchy,
Ctrl+Click
Семантический движок
PSI
в IntelliJ IDEA
индекс, связи,
компилятор
Агент
вызывает инструменты
MCP-сервер
MCP IDEA:
те же запросы к движку

Что пошло не так: MCP IDEA

01 **Зависания, сборка 2025.3** 60% сессий зависают под нагрузкой

02 **Несколько проектов, сборка 2025.3** при параллельной работе с проектами каждый вызов требует явного пути к проекту

03 **Тесты без вердикта, сборка 2026.2** на чтение логов прогона требуется дополнительное системное разрешение

MCP IDEA 2025.3

17.9 мин

время

$0.116

стоимость

DeepSeek V4 Flash

244K + 5.4M

кэш

токены

87

tool-calls

вызовов на прогон

100%

место фикса

соответствие мейнтейнеру

80%

тесты

добавлены в нужный тестовый класс

60%

код в стиле проекта

сообщение об ошибке по конвенции Calcite

MCP IDEA 2026.2

9.8 мин

время

$0.091
стоимость
DeepSeek V4 Flash

207K + 3.6M
кэш
токены

85
tool-calls
вызовов на прогон

100%
место фикса
соответствие мейнтейнеру

80%
тесты
добавлены в нужный тестовый класс

60%
код в стиле проекта
сообщение об ошибке по конвенции Calcite

А что, если у меня VPS с код-агентом?

SERENA

semantic retrieval & editing
noetic agent

Семантические инструменты для агента:
аналог MCP-инструментов IntelliJ IDEA.



github.com/oraios/serena

Внутри: языковые серверы (LSP)
для Java поднимает JDTLS,
подключается к агенту по MCP

find_symbol	find_referencing_symbols	get_symbols_overview	replace_symbol_body

ECLIPSE FOUNDATION JDTLS

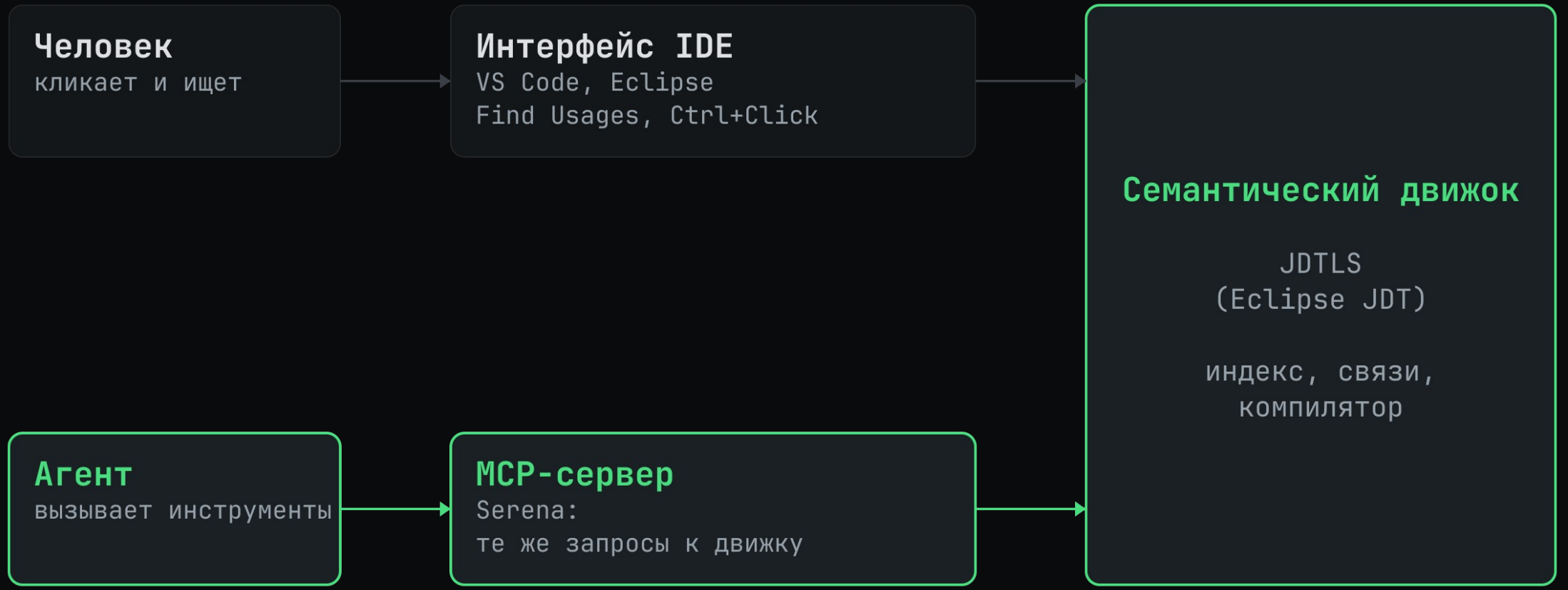
Eclipse JDT Language Server:
языковой сервер для Java по LSP,
работает с любым редактором,
в том числе с VS Code.



github.com/eclipse-jdtls/eclipse.jdt.ls

Внутри: Eclipse JDT
компилятор, индекс и поиск ссылок;
Maven и Gradle через M2Eclipse и Buildship

LSP4J	Eclipse JDT	M2Eclipse	Buildship


Человек
кликает и ищет
Интерфейс IDE
VS Code, Eclipse
Find Usages, Ctrl+Click
Семантический движок
JDTLS
(Eclipse JDT)
индекс, связи,
компилятор
Агент
вызывает инструменты
MCP-сервер
Serena:
те же запросы к движку

Serena + JDTLS: выводы

01 **Поиск кода по индексу** с учётом семантических связей

02 **Компилятор и диагностика** вызов компилятора и диагностика после правок

03 **Работает без IDE** в headless-режиме: на VPS или в CI

04 **MCP может не подняться** при некорректной настройке Serena

05 **Дорого по ресурсам** на каждый поток Serena и каждого агента около 3 ГБ оперативной памяти на проекте Calcite

06 **Всё ещё только 60%** прогонов попадают в конвенцию проекта

Serena + JDTLS

19.8 мин

время

медиана пяти прогонов

$0.100

стоимость

DeepSeek V4 Flash

193K + 3.6M

кэш

токены

104

tool-calls

вызовов на прогон

100%

место фикса

соответствие мейнтейнеру

100%

тесты

добавлены в нужный тестовый класс

60%

код в стиле проекта

сообщение об ошибке по конвенции Calcite

Попадание в конвенцию проекта ⩽60%

06 / 07

База знаний проекта

Текст
Read
ripgrep

Синтаксис
ast-grep

Семантика
IntelliJ IDEA по MCP
Serena + JDTLS

База знаний
Graphify
Understand-Anything

Graphify

Агент строит граф знаний проекта
и дальше в первую очередь обращается
к графу, а уже потом ищет по файлам.

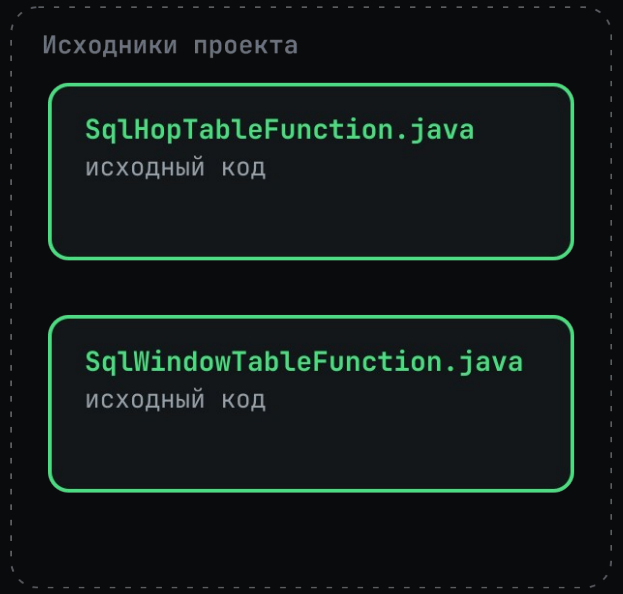


github.com/Graphify-Labs/graphify

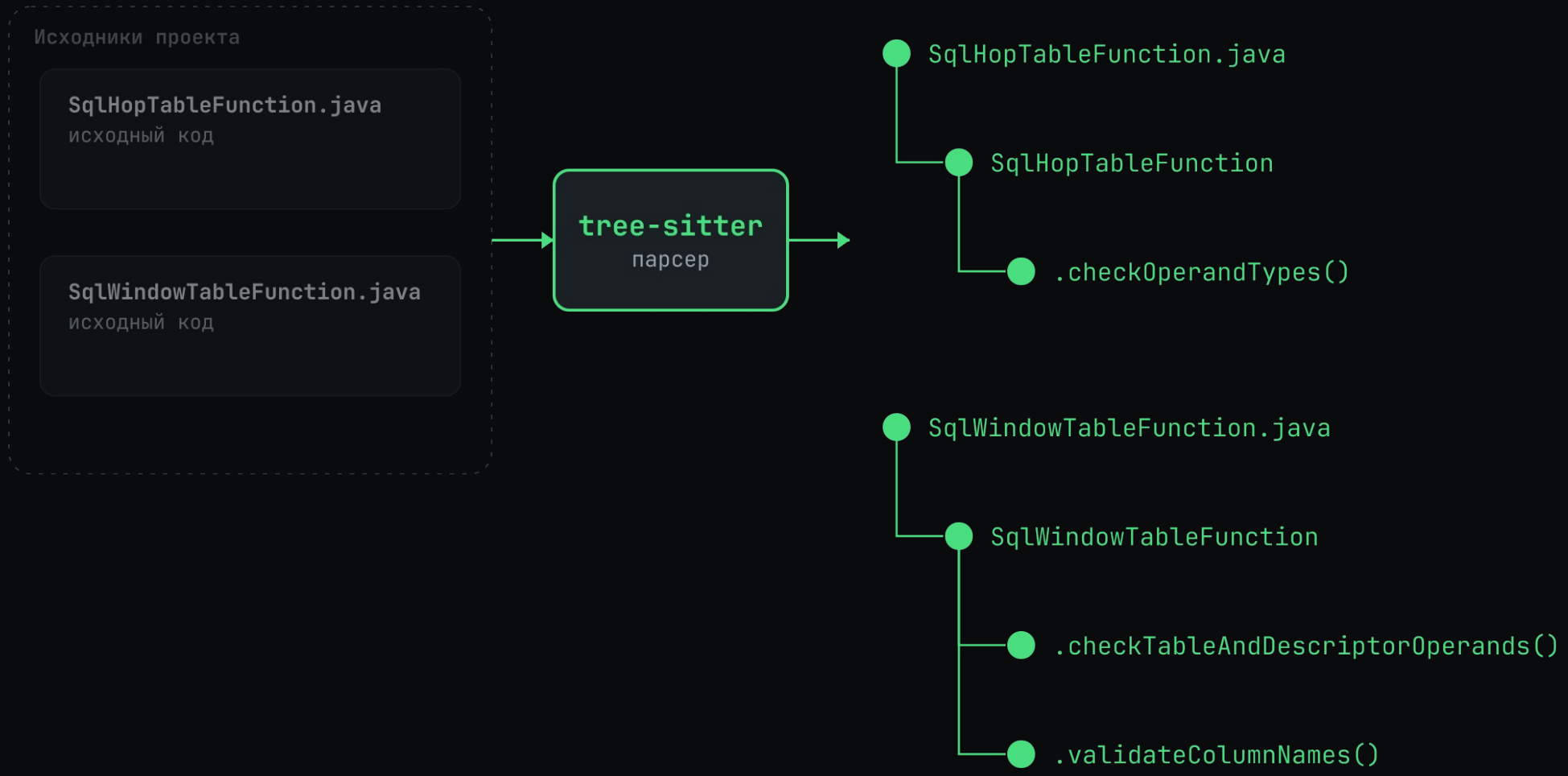
Внутри: tree-sitter
код разбирается локально, без модели;
рёбра EXTRACTED или INFERRED

tree-sitter | граф знаний | MCP

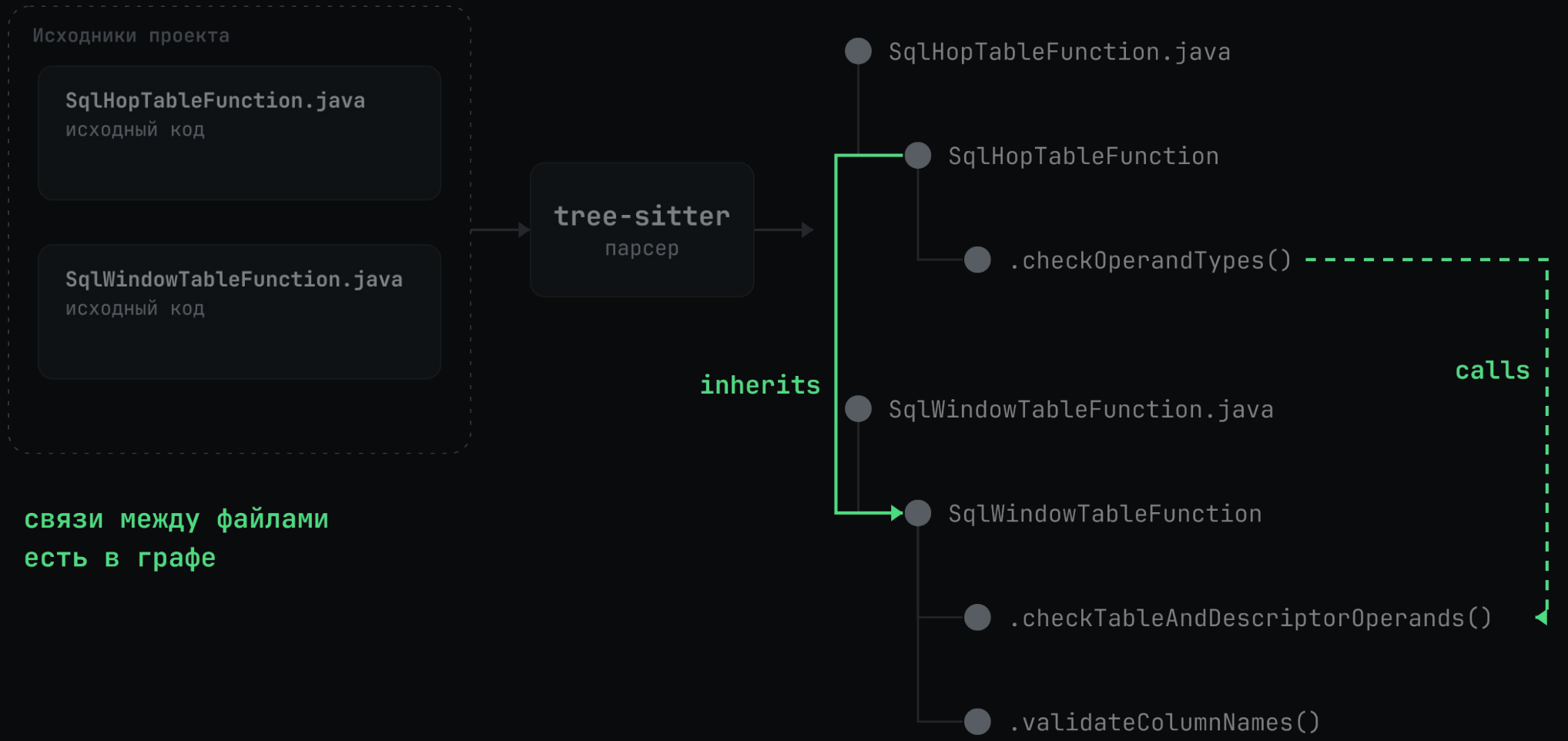
Graphify строит граф сам



Graphify строит граф сам



Graphify строит граф сам



связи между файлами
есть в графе

Graphify: ast-grep на максималках

01 **Граф строится один раз** — не разбирает проект заново на каждый запрос: весь Calcite за минуту и $0

02 **Связывает файлы** — наследование и вызовы между классами есть в графе

03 **Отдаёт граф через MCP** — агент спрашивает граф и перестаёт ошибаться с паттернами

04 **Всё ещё не видит .properties** — связи кода с текстом ошибки в графе нет: конвенция в 0% прогонов

Graphify

12.6 мин

время

$0.061

стоимость

DeepSeek V4 Flash

138K + 2.5M

кэш

токены

68

tool-calls

вызовов на прогон

100%

место фикса

соответствие мейнтейнеру

40%

тесты

добавлены в нужный тестовый класс

0%

код в стиле проекта

сообщение об ошибке по конвенции Calcite

Understand-Anything

Скилл строит предагрегированный граф
и связи в нём, затем с помощью LLM
находит неочевидные связи, проводит
ревью и собирает граф связей и знаний.

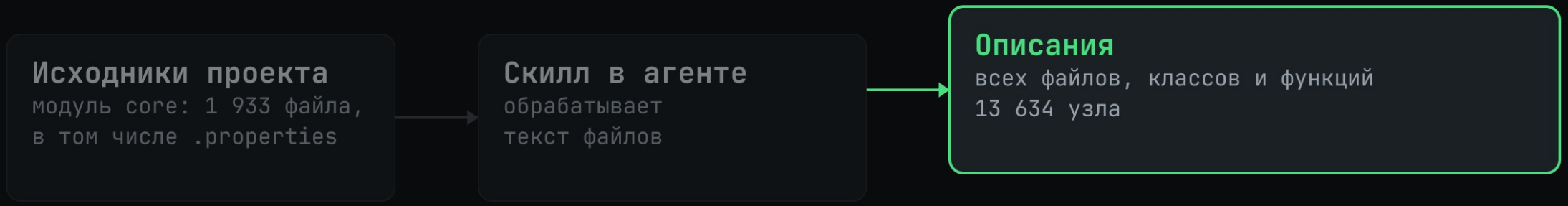


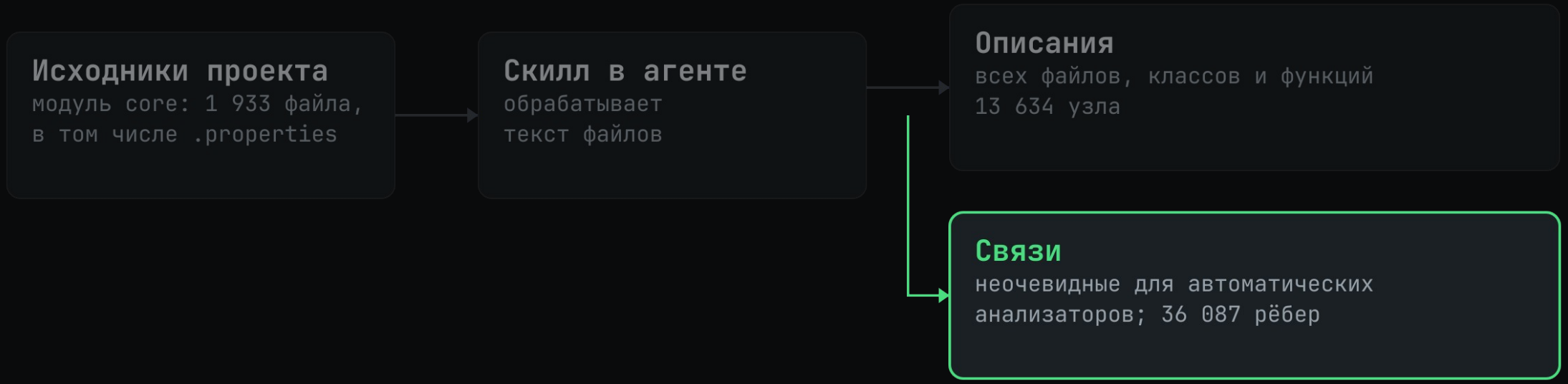
github.com/Egonex-AI/Understand-Anything

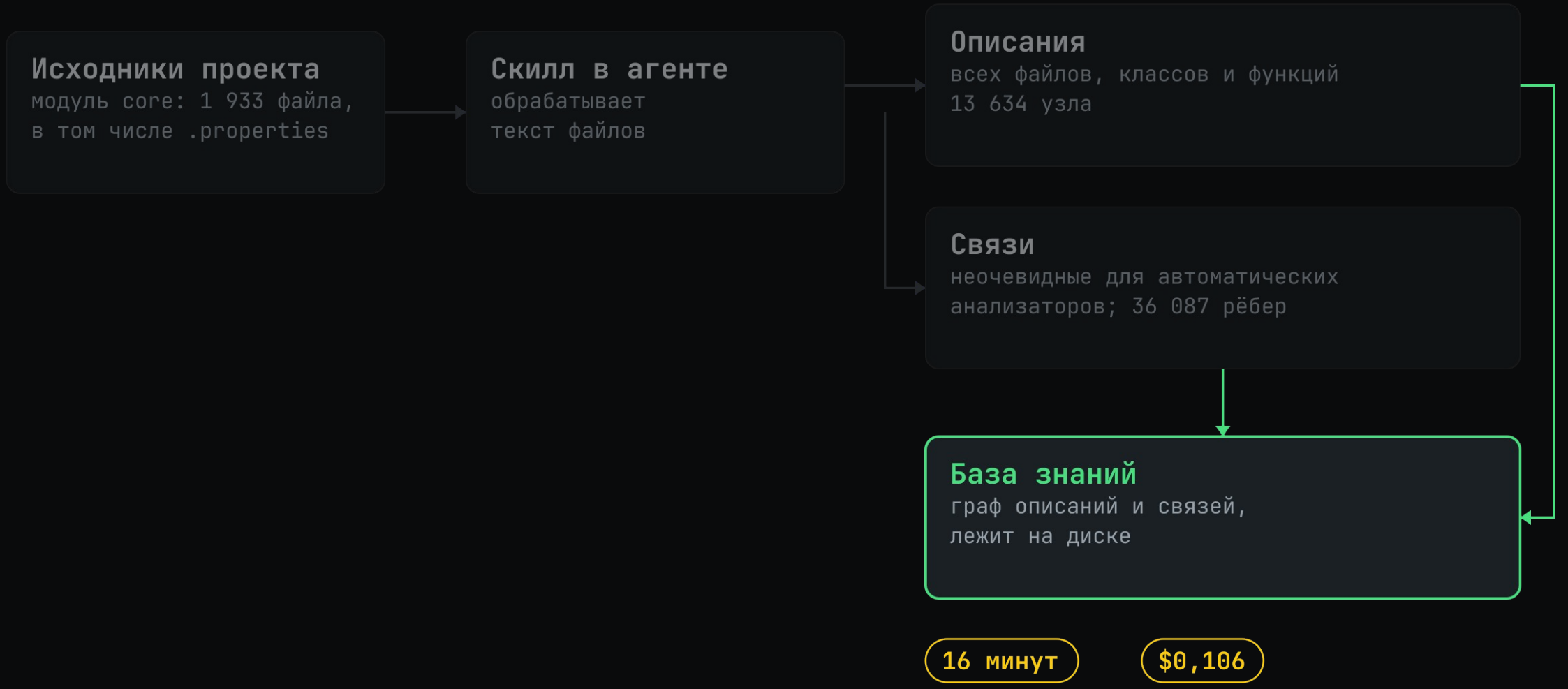
Внутри: Python-парсер
+ исследование и валидация с помощью LLM

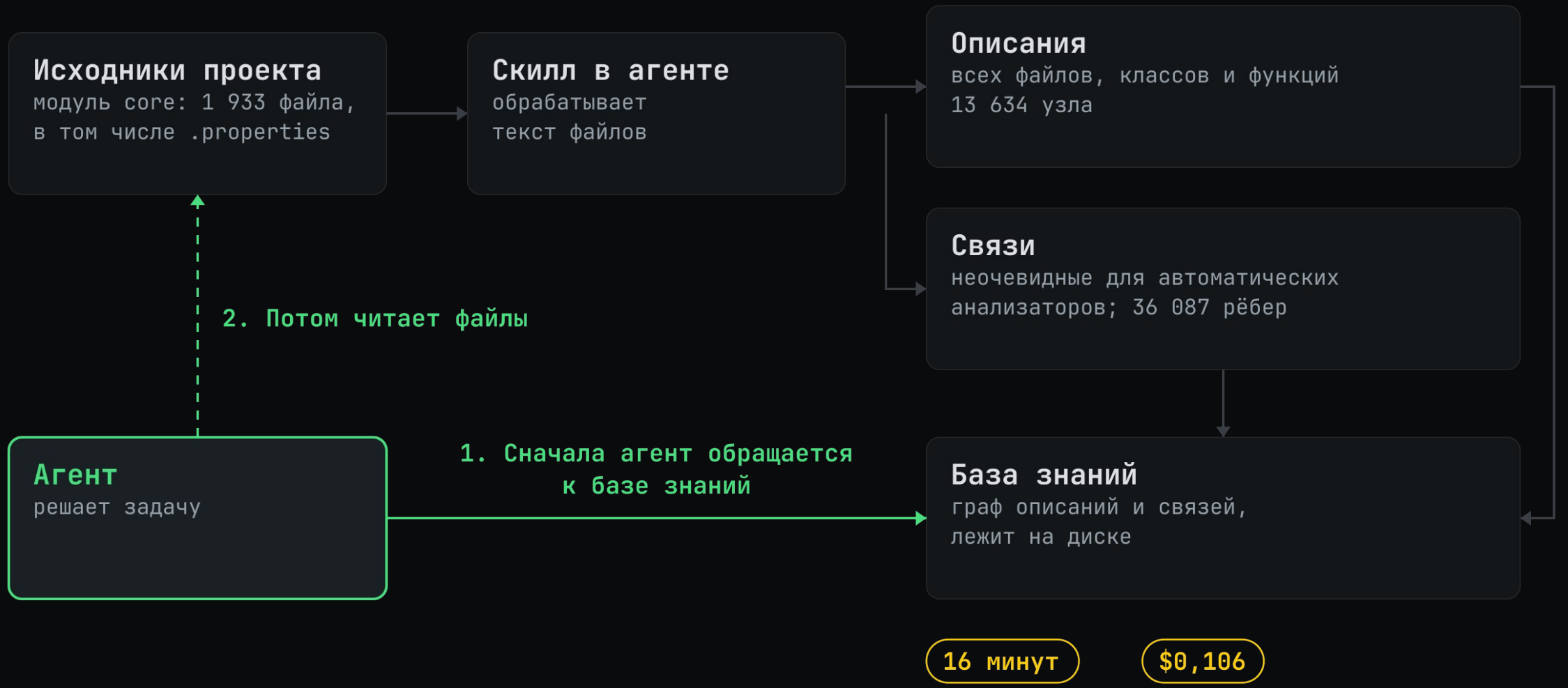
project-scanner | file-analyzer | architecture-analyzer | tour-builder

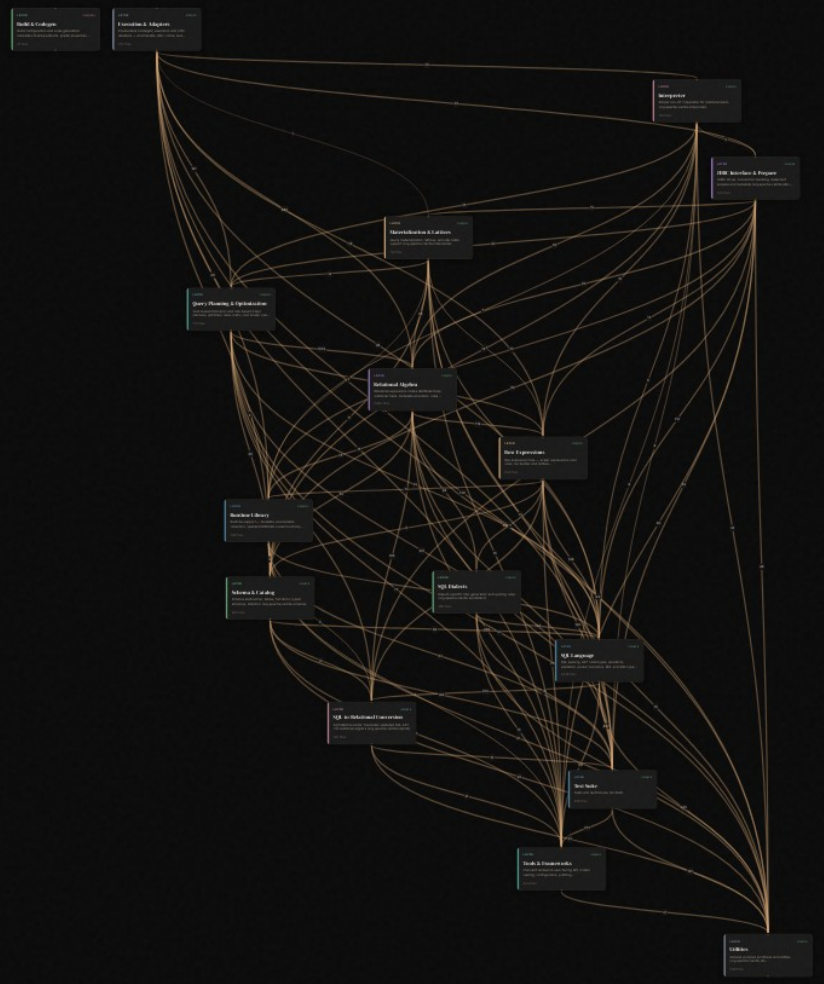

Исходники проекта
модуль core: 1 933 файла,
в том числе .properties
Скилл в агенте
обрабатывает
текст файлов


Исходники проекта
модуль core: 1 933 файла,
в том числе .properties
Скилл в агенте
обрабатывает
текст файлов
Описания
всех файлов, классов и функций
13 634 узла


Исходники проекта
модуль core: 1 933 файла,
в том числе .properties
Скилл в агенте
обрабатывает
текст файлов
Описания
всех файлов, классов и функций
13 634 узла
Связи
неочевидные для автоматических
анализаторов; 36 087 рёбер


Исходники проекта
модуль core: 1 933 файла,
в том числе .properties
Скилл в агенте
обрабатывает
текст файлов
Описания
всех файлов, классов и функций
13 634 узла
Связи
неочевидные для автоматических
анализаторов; 36 087 рёбер
База знаний
граф описаний и связей,
лежит на диске
16 минут
$0,106


Исходники проекта
модуль core: 1 933 файла,
в том числе .properties
Скилл в агенте
обрабатывает
текст файлов
Описания
всех файлов, классов и функций
13 634 узла
Связи
неочевидные для автоматических
анализаторов; 36 087 рёбер
2. Потом читает файлы
Агент
решает задачу
1. Сначала агент обращается
к базе знаний
База знаний
граф описаний и связей,
лежит на диске
16 минут
$0,106



Project Tour

14 steps · Guided walkthrough of the codebase

Start Tour

STEPS

1. Project Overview
2. JDBC Entry Point
3. SQL Parsing
4. SQL Validation
5. SQL-to-Relational Conversion
6. Row Expressions (Rex)
7. Query Planning & Optimization
8. Optimizer Rules & Metadata
9. Plan Preparation
10. Enumerable (Codegen) Execution
11. Interpreter (non-JIT path)
12. Schema, Catalog & Materialization
13. Utilities & Runtime
14. Testing

Understand-Anything: выводы

01 **Лучшее попадание в конвенцию** 80% прогонов: .properties в графе отдельным узлом

02 **Совпадение с мейнтейнером** в двух прогонах код фикса побайтно равен эталону

03 **Дорогое построение** 16 минут и $0,106 только на модуль core

04 **Интерактивный скилл** в headless остановился на вопросе: ответ пришлось дать заранее в промпте

Understand-Anything

13.2 мин

время

$0.081

стоимость

DeepSeek V4 Flash

175K + 3.5M

кэш

токены

80

tool-calls

вызовов на прогон

100%

место фикса

соответствие мейнтейнеру

100%

тесты

добавлены в нужный тестовый класс

80%

код в стиле проекта

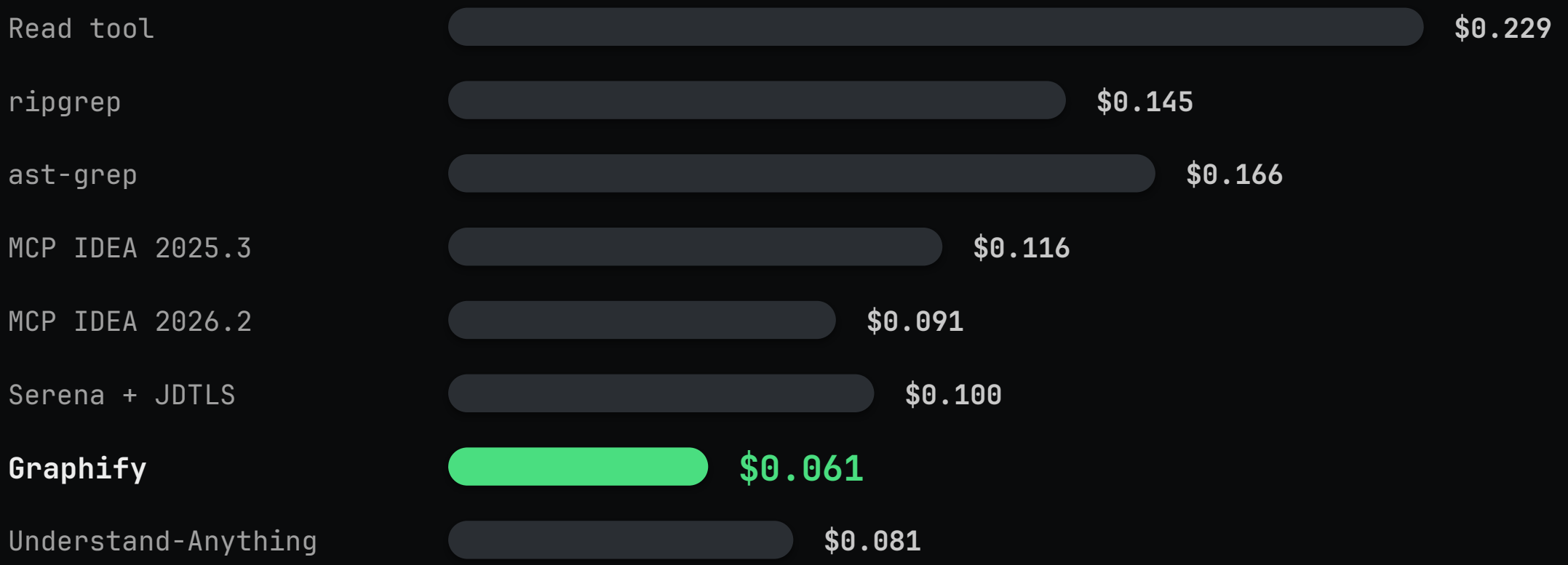
лучший результат эксперимента

07 / 07

Финальный результат

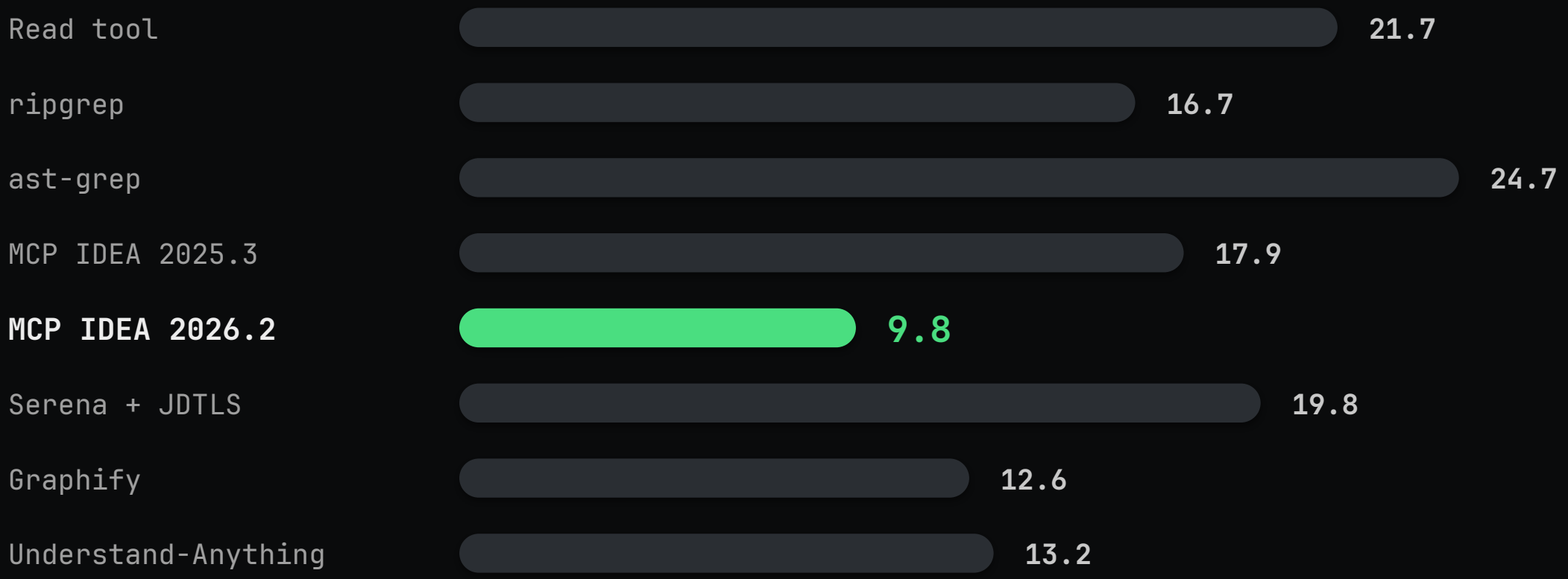
ИТОГ

Стоимость решения задачи



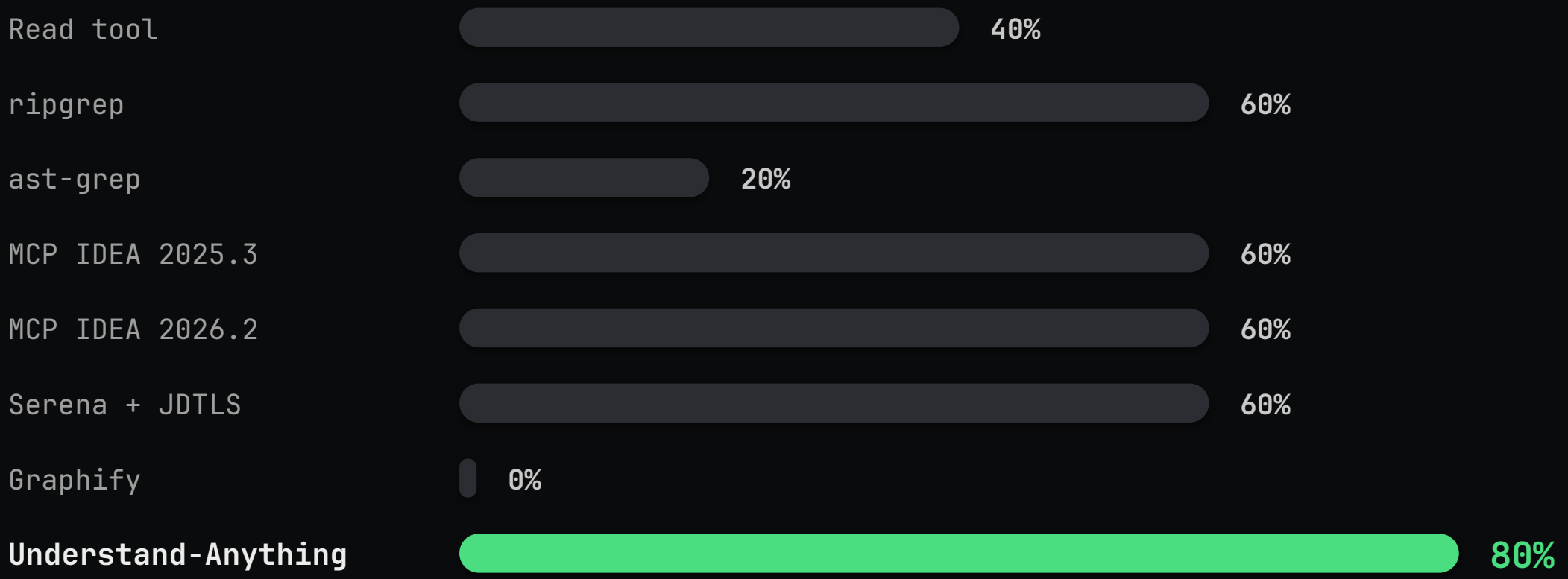
ИТОГ

Время решения задачи



ИТОГ · КАЧЕСТВО ПРОТИВ ЭТАЛОНА

Кто попал в конвенции проекта



Мой личный выбор

Локально



+



IntelliJ IDEA Understand-Anything

VPS и CI



+



Serena Understand-Anything

Рефлексируйте и экспериментируйте!

semantic-analyzers-for-java

- Бенчмарк
- Примеры
- Статистика



github.com/a-simeshin/semantic-analyzers-for-java