



Миссия невыполнима? Глубокая отладка Java- и Kotlin- приложений с помощью AI



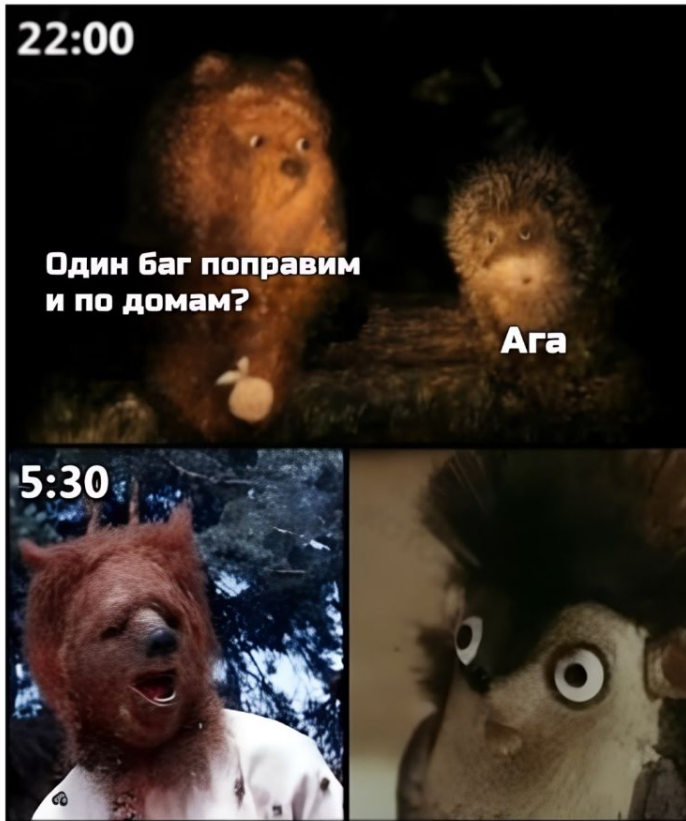
Егор Куликов

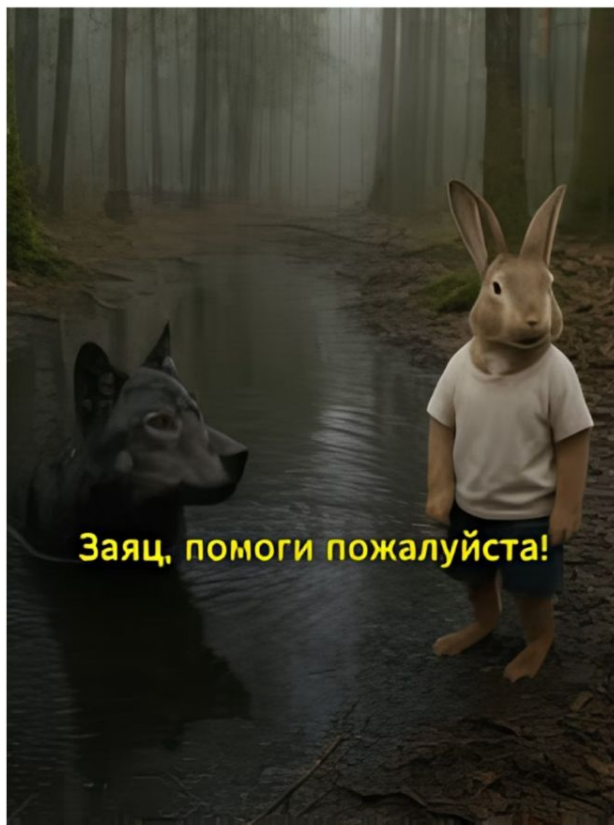
Veai

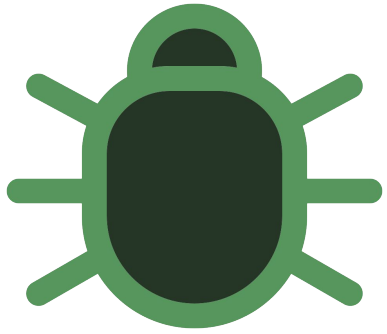


Илья Муравьев

Veai







+



AI
AGENT

=




Что падает?

```
@SpringBootTest(properties = { @ Alexey Sosnoviy
    "app.configuration.path=./src/test/resources/.bsl-language-server.json",
    "app.globalConfiguration.path=./src/test/resources/.bsl-language-server-only-en-param.json",
})
@CleanupContextBeforeClassAndAfterEachTestMethod
class ConfigurationTest {

    @Autowired
    private LanguageServerConfiguration configuration;

    @Test @ Alexey Sosnoviy
    void createDefault() {

        assertThat(configuration.getLanguage()).isEqualTo(Language.EN);
        assertThat(configuration.isUseDevSite()).isTrue();
        assertThat(configuration.getDiagnosticsOptions().getParameters()).isNotEmpty();
    }
}
```



Expecting value to be true but was false
Expected :true
Actual :false
[<Click to see difference>](#)

Почему баг нетривиальный?

```
@SpringBootTest(properties = { @ Alexey Sosnoviy
    "app.configuration.path=./src/test/resources/.bsl-language-server.json",
    "app.globalConfiguration.path=./src/test/resources/.bsl-language-server-only-en-param.json",
})
@CleanupContextBeforeClassAndAfterEachTestMethod
class ConfigurationTest {

    @Autowired
    private LanguageServerConfiguration configuration;

    @Test @ Alexey Sosnoviy
    void createDefault() {

        assertThat(configuration.getLanguage()).isEqualTo(Language.EN);
        assertThat(configuration.isUseDevSite()).isTrue();
        assertThat(configuration.getDiagnosticsOptions().getParameters()).isNotEmpty();
    }
}
```

⚡

Expecting value to be true but was false
Expected :true
Actual :false
[<Click to see difference>](#)

Почему баг нетривиальный?

```
@SpringBootTest(properties = { @ Alexey Sosnoviy
    "app.configuration.path=./src/test/resources/.bsl-language-server.json",
    "app.globalConfiguration.path=./src/test/resources/.bsl-language-server-only-en-param.json",
})
@CleanupContextBeforeClassAndAfterEachTestMethod
class ConfigurationTest {

    @Autowired
    private LanguageServerConfiguration configuration;

    @Test @ Alexey Sosnoviy
    void createDefault() {

        assertThat(configuration.getLanguage()).isEqualTo(Language.EN);
        assertThat(configuration.isUseDevSite()).isTrue();
        assertThat(configuration.getDiagnosticsOptions().getParameters()).isNotEmpty();
    }
}
```

⚡

Expecting value to be true but was false
Expected :true
Actual :false
[<Click to see difference>](#)

✓ "Test worker"@1 in group "main": RUNNING

↶ setUseDevSite:106, LanguageServerConfiguration (com.
10 hidden frames
copyPropertiesFrom:230, LanguageServerConfiguration
loadConfigurationFile:224, LanguageServerConfiguration
init:133, LanguageServerConfiguration (com.github._1c_...
155 hidden frames

Почему баг нетривиальный?

```
@SpringBootTest(properties = { @ Alexey Sosnoviy
    "app.configuration.path=./src/test/resources/.bsl-language-server.json",
    "app.globalConfiguration.path=./src/test/resources/.bsl-language-server-only-en-param.json",
})
@CleanupContextBeforeClassAndAfterEachTestMethod
class ConfigurationTest {

    @Autowired
    private LanguageServerConfiguration configuration;

    @Test @ Alexey Sosnoviy
    void createDefault() {

        assertThat(configuration.getLanguage()).isEqualTo(Language.EN);
        assertThat(configuration.isUseDevSite()).isTrue();
        assertThat(configuration.getDiagnosticsOptions().getParameters()).isNotEmpty();
    }
}
```

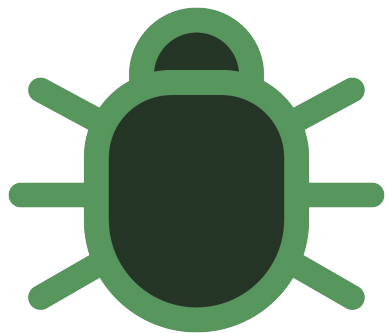
⚡

Expecting value to be true but was false
Expected :true
Actual :false
[<Click to see difference>](#)

✓ "Test worker"@1 in group "main": RUNNING

↶ setUseDevSite:106, LanguageServerConfiguration (com.
10 hidden frames
copyPropertiesFrom:230, LanguageServerConfiguration
loadConfigurationFile:224, LanguageServerConfiguration
init:133, LanguageServerConfiguration (com.github._1c_...
155 hidden frames

```
{
    ...,
    "useDevSite": true,
    ...
}
```



+



Opus 4.6
* Claude

=





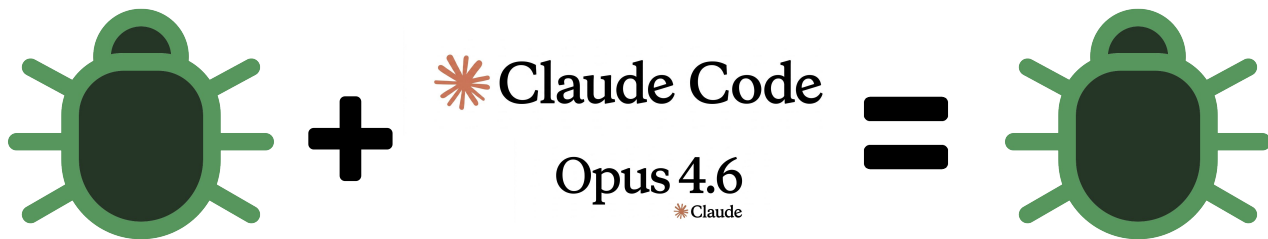
Когда починил баг

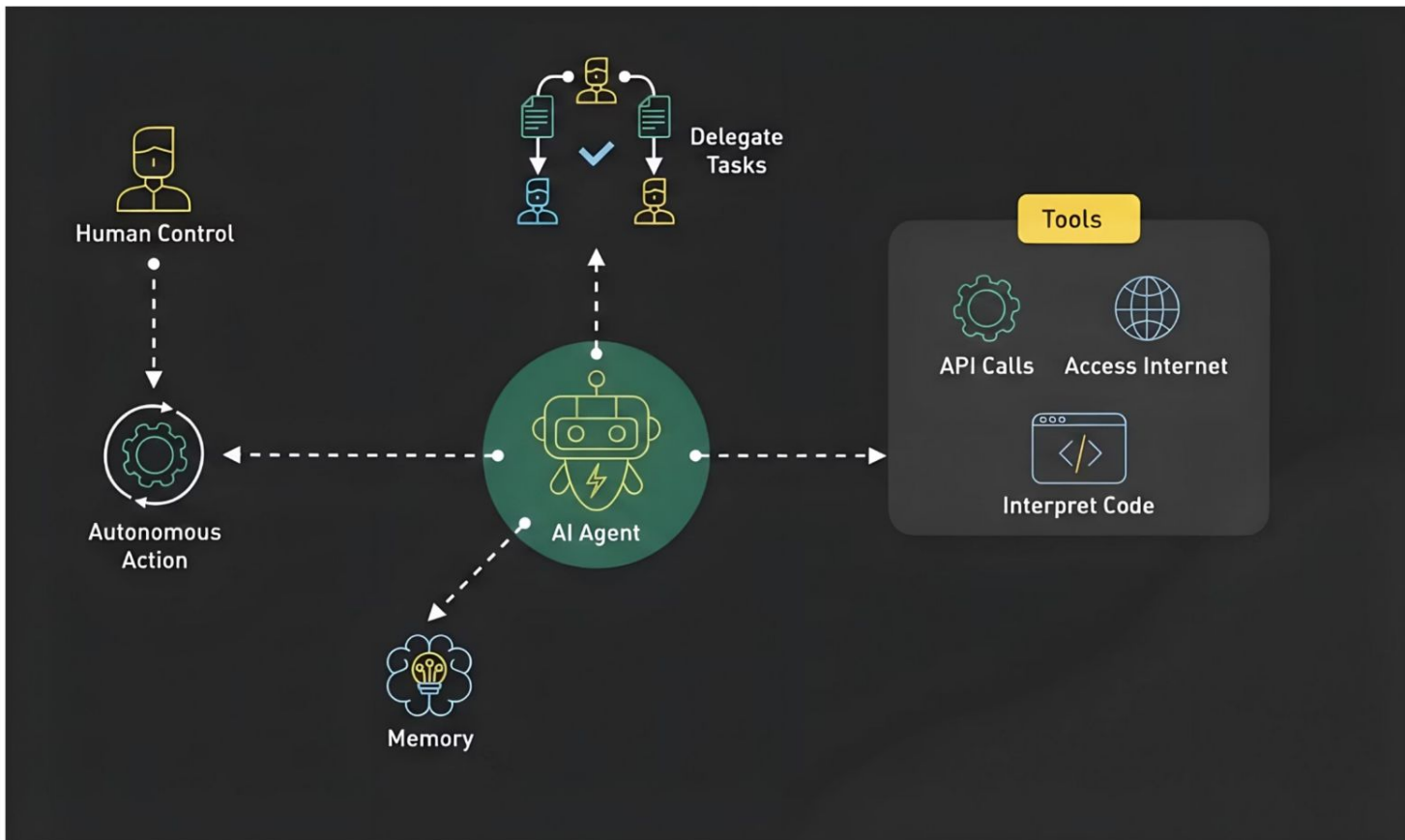
```
configuration = mapper.readValue(configurationFile, LanguageServerConfiguration.class);
```

```
// Workaround for binding issues:  
// some properties are not reliably populated due to  
// @JsonCreator(mode = DISABLED) + Lombok + custom deserializers.  
var tree = mapper.readTree(configurationFile);  
var useDevSiteNode = tree.get("useDevSite");  
configuration.setUseDevSite(useDevSiteNode.asBoolean());
```

```
this.configurationFile = configurationFile;  
copyPropertiesFrom(configuration);
```







Инструменты агента

Level 1

- Полнотекстовый поиск
- Чтение файлов
- Редактирование файлов
- Анализ логов

Level 2

- Запуск тестов
- Ребилд модуля
- Работа с терминалом
- Отладочная печать

Инструменты Veai

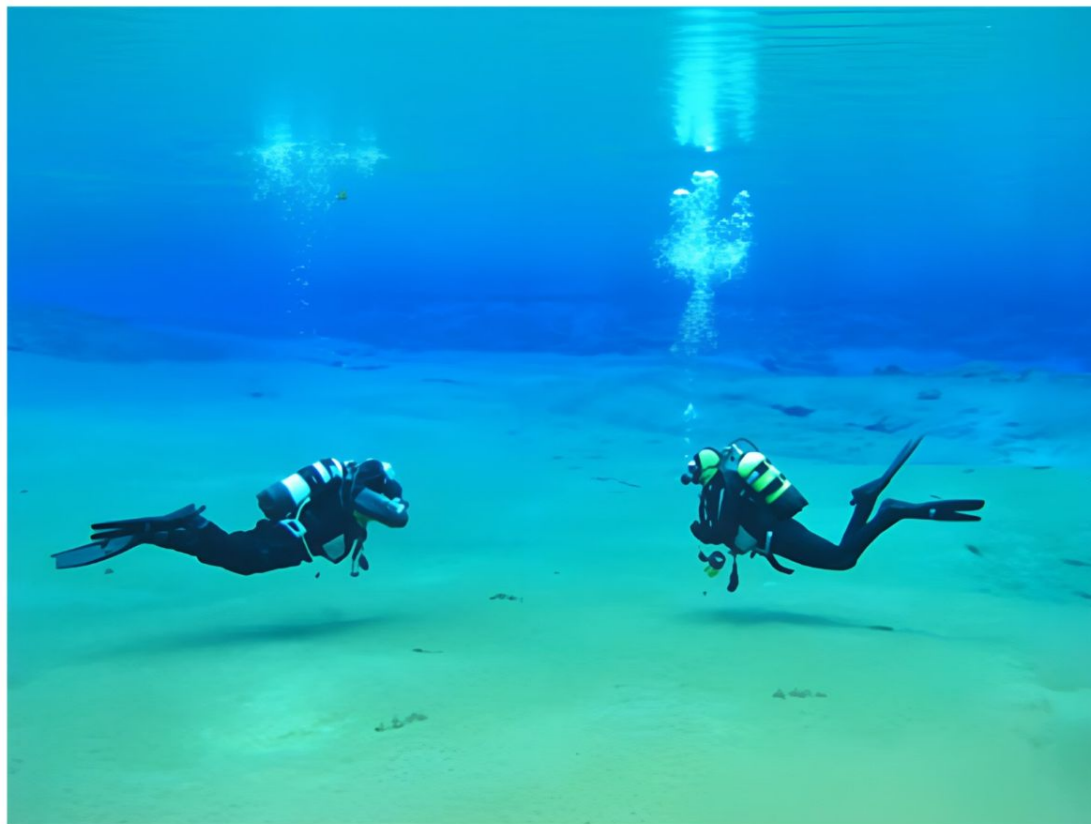
Глубокая интеграция с IDE

- Поиск по библиотекам
- Декомпиляция
- Получение ошибок компиляции средства IDE
- Запуск с настраиваемыми параметрами

Формальный анализ кода

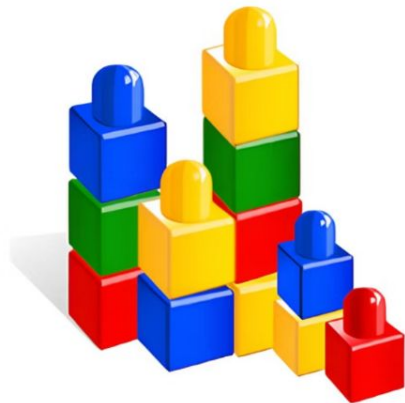
- Поиск уязвимостей







+



=



Соберем трейс выполнения, AOP

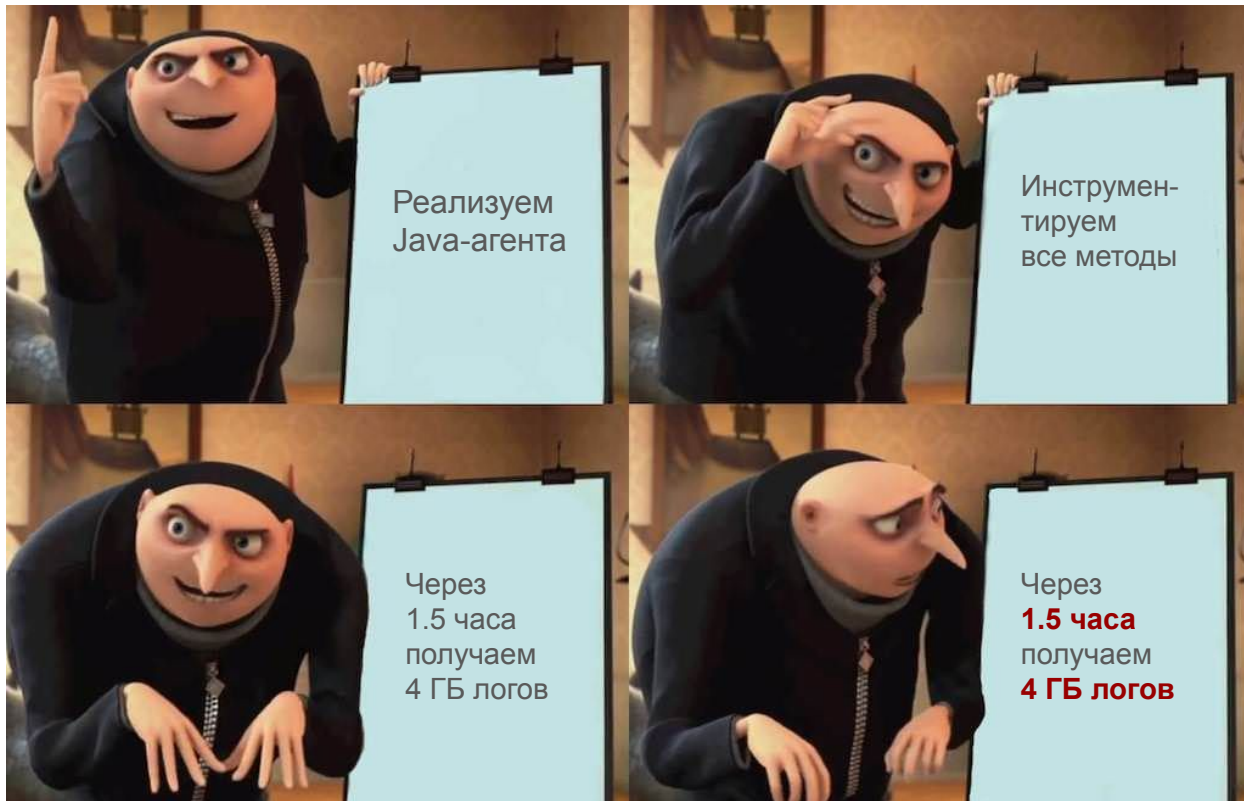
```
@Aspect
class SimpleTracingAspect {

    @Around("execution(* *(..))")
    Object onMethodCall(ProceedingJoinPoint methodCall) {

        Object result = methodCall.proceed();

        logger.info(methodCall, methodInvocation.getArgs(), result);

        return result;
    }
}
```



Как выглядит трейс?

... (2 миллиона строк)

```
+ ScannedGenericBeanDefinition.setResource(FileSystemResource(file [/home/ilya/IdeaProjects/bsl-language-server/
+ ScannedGenericBeanDefinition.setSource(FileSystemResource(file [/home/ilya/IdeaProjects/bsl-language-server/k
+ ClassPathScanningCandidateComponentProvider.isCandidateComponent(ScannedGenericBeanDefinition(Generic bean: c
  - AnnotatedBeanDefinition.getMetadata() returned SimpleAnnotationMetadata(com.github._1c_syntax.bsl.languages
    * ScannedGenericBeanDefinition.getMetadata() returned SimpleAnnotationMetadata(com.github._1c_syntax.bsl.la
  - AnnotationMetadata.isIndependent() returned true
  - AnnotationMetadata.isConcrete() returned true
+ Set.add(ScannedGenericBeanDefinition(Generic bean: scope=; abstract=false; lazyInit=null; autowireMode=0; dep
+ Resource.getFilename() returned "BSLLanguageServer.class"
+ "BSLLanguageServer.class".contains("$$") returned false
+ ClassPathScanningCandidateComponentProvider.getMetadataReaderFactory() returned CachingMetadataReaderFactory@
+ MetadataReaderFactory.getMetadataReader(FileSystemResource(file [/home/ilya/IdeaProjects/bsl-language-server/
+ ClassPathScanningCandidateComponentProvider.isCandidateComponent(SimpleMetadataReader@7b8c2e4a) returned fals
  - List.iterator() returned Itr@6d13fa90
  - Iterator.hasNext() returned false
  - List.iterator() returned Itr@4e7ab1c6
  - Iterator.hasNext() returned true
  - Iterator.next() returned AnnotationTypeFilter@2c91d4fe
  - ClassPathScanningCandidateComponentProvider.getMetadataReaderFactory() returned CachingMetadataReaderFactory
  - TypeFilter.match(SimpleMetadataReader@7b8c2e4a, CachingMetadataReaderFactory@5f2a9c1d) returned false
  - Iterator.hasNext() returned false
+ Resource.getFilename() returned "BSLTextDocumentService.class"
+ "BSLLanguageServer.class".contains("$$") returned false
+ ClassPathScanningCandidateComponentProvider.getMetadataReaderFactory() returned CachingMetadataReaderFactory@
+ MetadataReaderFactory.getMetadataReader(FileSystemResource(file [/home/ilya/IdeaProjects/bsl-language-server/
+ ClassPathScanningCandidateComponentProvider.isCandidateComponent(SimpleMetadataReader@1a5f8b3e) returned fals
```

... (ещё 27 миллионов строк)

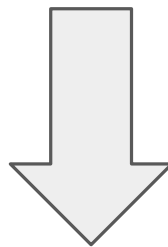
Есть ли что-то полезное в этих 4 ГБ?

```
+ ... (ещё 7 успешно десериализованных полей)
+ BeanDeserializer.deserializeAndSet("language")
+ BeanDeserializer.deserializeAndSet("codeLens")
+ BeanDeserializer.deserializeAndSet("annotations")
+ BeanDeserializer.handleUnknownProperty("traceLog")
+ BeanDeserializer.handleUnknownProperty("useDevSite")
+ ... (ещё 3 неуспешно десериализованные поля)
```

после поля
“annotations”
десериализация
перестает работать

```
+ ... (ещё 7 успешно десериализованных полей)
+ BeanDeserializer.deserializeAndSet("language")
+ BeanDeserializer.deserializeAndSet("codeLens")
+ BeanDeserializer.deserializeAndSet("annotations")
+ BeanDeserializer.handleUnknownProperty("traceLog")
+ BeanDeserializer.handleUnknownProperty("useDevSite")
+ ... (ещё 3 неуспешно десериализованные поля)
```

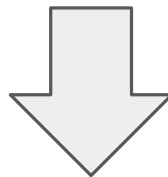
после поля
“annotations”
десериализация
перестает работать



 veai (2 минуты)

```
Set<String> annotationsSet = new TreeSet<>(String.CASE_INSENSITIVE_ORDER);
var objectMapper = (ObjectMapper) p.getCodec();
// Avoid re-reading consumed JsonParser.
objectMapper.readerForUpdating(annotationsSet).readValue(p);
objectMapper.readerForUpdating(annotationsSet).readValue(annotations);
```

```
Expecting value to be true but was false
Expected :true
Actual   :false
<Click to see difference>
```



2 минуты

```
Set<String> annotationsSet = new TreeSet<>(String.CASE_INSENSITIVE_ORDER);
var objectMapper = (ObjectMapper) p.getCodec();
// Avoid re-reading consumed JsonParser.
objectMapper.readerForUpdating(annotationsSet).readValue(p);
objectMapper.readerForUpdating(annotationsSet).readValue(annotations);
```

Я в восхищении!

Как собрать трейс быстрее?



Идея: не логировать библиотечные методы

```
@Aspect
class SimpleTracingAspect {

    @Around("execution(* *(..))")
    @Around("execution(* com.myapp.*(..))")
    Object onMethodCall(ProceedingJoinPoint methodCall) {

        Object result = methodCall.proceed();

        logger.info(methodCall, methodInvocation.getArgs(), result);

        return result;
    }
}
```

Идея: не логировать библиотечные методы

```
+ ...  
+ BeanDeserializer.deserializeAndSet("language")  
+ BeanDeserializer.deserializeAndSet("codeLens")  
+ BeanDeserializer.deserializeAndSet("annotations") ←  
+ BeanDeserializer.handleUnknownProperty("useDevSite")  
+ BeanDeserializer.handleUnknownProperty("traceLog")  
+ ...
```

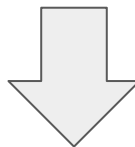
мы перестали
логировать всё
полезное

Что логируем?

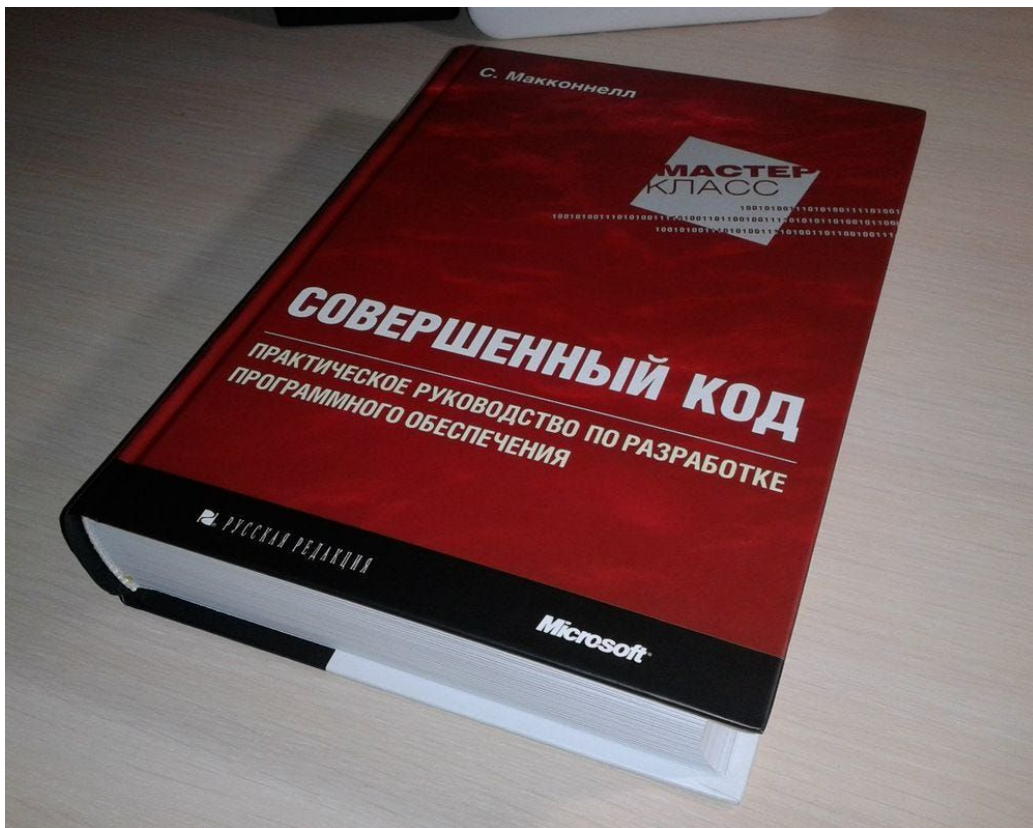
Вызов метода приложения	Вызов метода библиотеки
логируем	не логируем

Что логируем?

Вызов метода приложения	Вызов метода библиотеки
логируем	не логируем



	Вызов метода приложения	Вызов метода библиотеки
Вызов из приложения	логируем	логируем
Вызов из библиотеки	логируем (это колбеки)	не логируем (это внутренка библиотек)



1.5 часа → несколько секунд
общее время — 2 минуты

Call-аспекты

```
@Aspect
class SimpleTracingAspect {

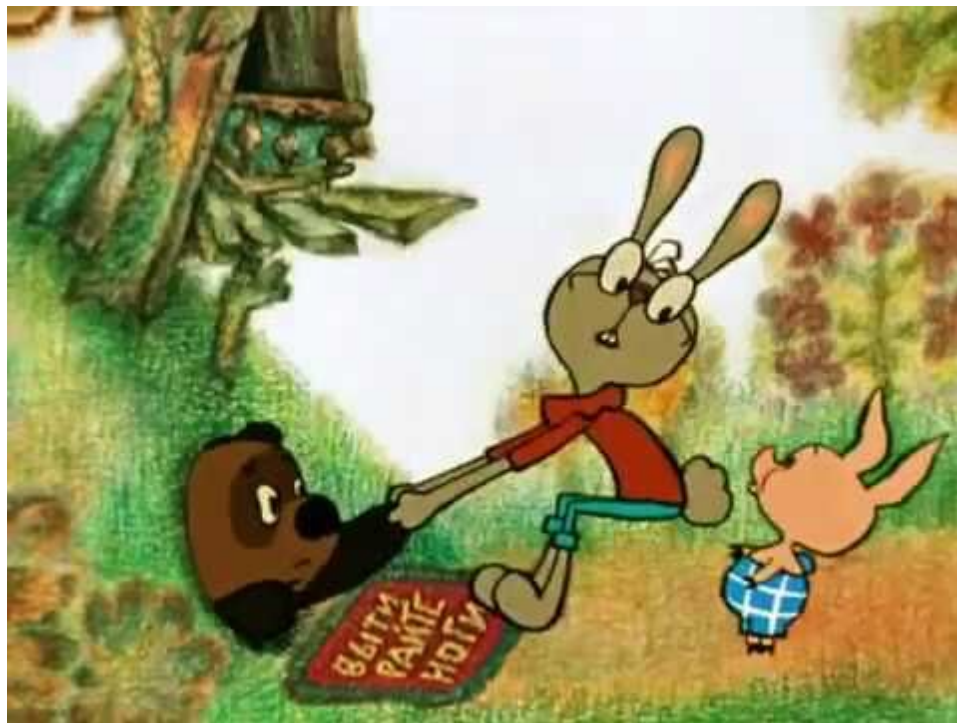
    @Around("execution(* com.myapp.*(..))")
    @Around("call(* *(..)) && within(com.myapp.*)")
    Object onMethodCall(ProceedingJoinPoint methodCall) {

        Object result = methodCall.proceed();

        logger.info(methodCall, methodInvocation.getArgs(), result);

        return result;
    }
}
```

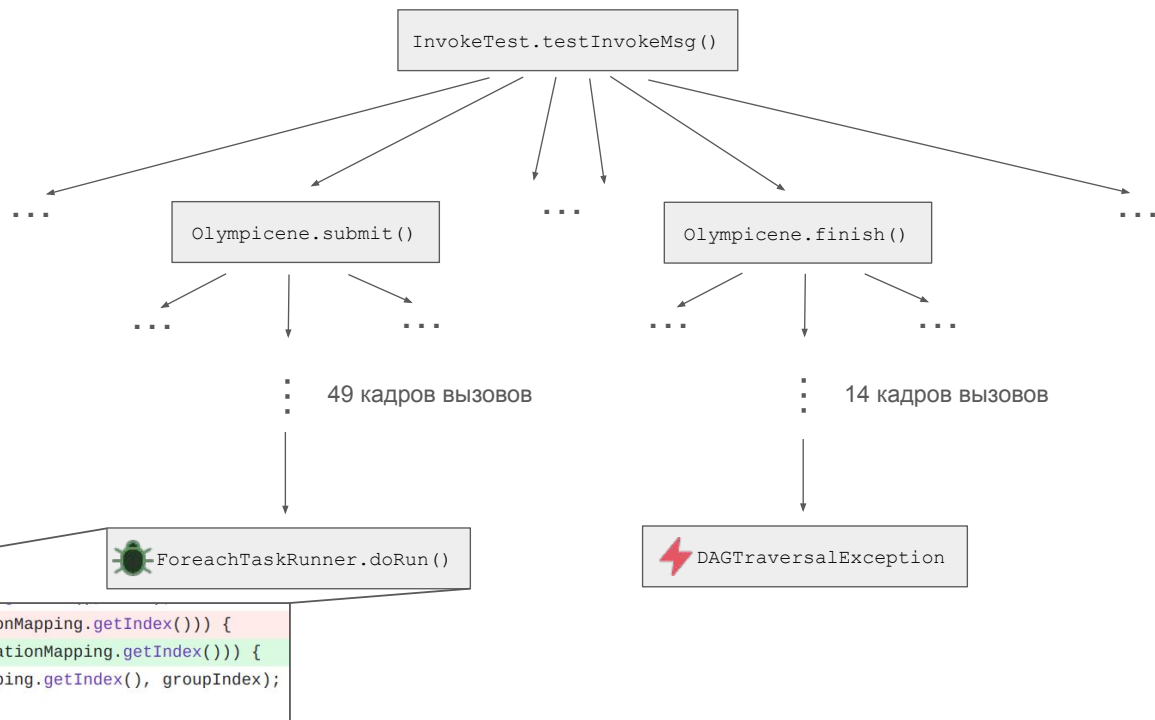
Как уместиться в контекстное окно?



Отрежем глубокие вызовы



Пример 2

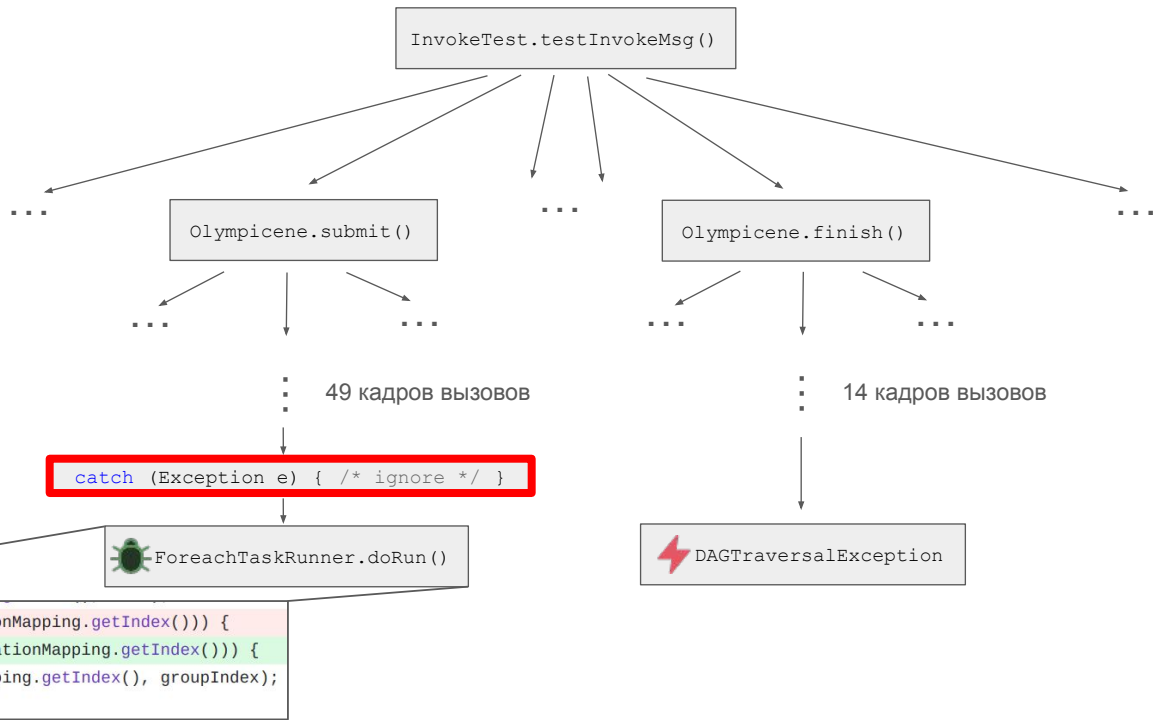


Увеличим глубину отсечения?

4 ГБ → 50 МБ

**Похудел но все равно
жирный (**

Приоритезация ветвей



Эвристики

Уже рассмотренные эвристики

- не инструментировать всё подряд
- ограничение глубины
- приоритезация ветвей с исключениями

Другие эвристики

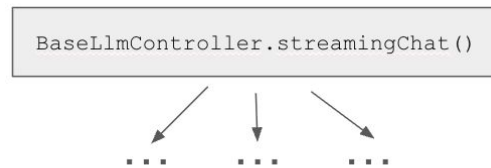
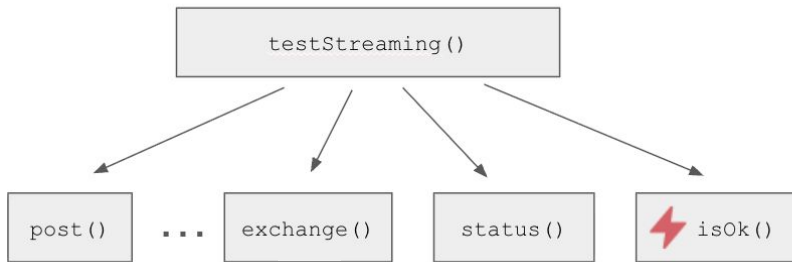
- Сжатие циклов
- Выделение репрезентативных вызовов
- Сжатие идентификаторов (“a41eb0c6” -> “17”)
- ...

Что делать с многопоточным кодом?



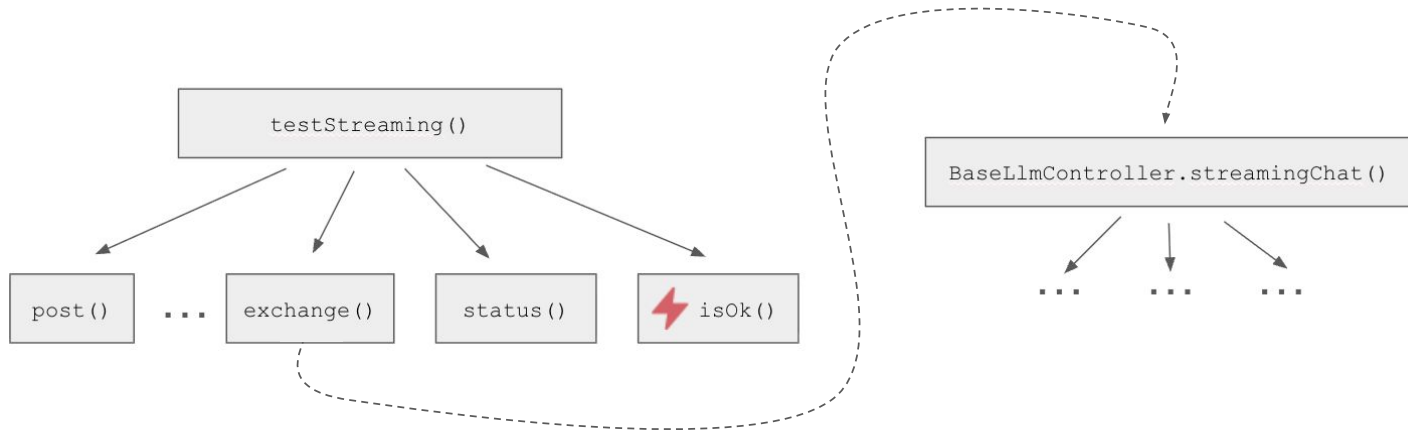
МНОГОПОТОЧНЫЙ КОД

```
fun testStreaming(prompt: Prompt, expectedAnswer: String) {  
    webClient  
        .post()  
        .uri("/v2/streaming")  
        .body(Mono.just(ChatRequest("gpt-5.4", prompt)))  
        .exchange()  
        .expectStatus().isOk  
        .expectBody().json(expectedAnswer)  
}
```



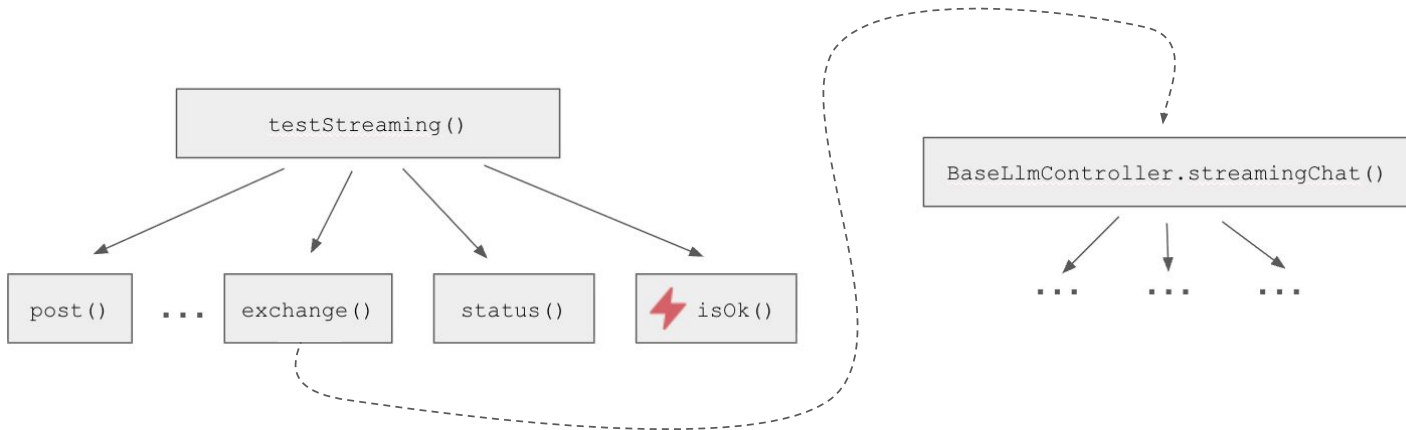
МНОГОПОТОЧНЫЙ КОД

```
fun testStreaming(prompt: Prompt, expectedAnswer: String) {  
    webClient  
        .post()  
        .uri("/v2/streaming")  
        .body(Mono.just(ChatRequest("gpt-5.4", prompt)))  
        .exchange()  
        .expectStatus().isOk  
        .expectBody().json(expectedAnswer)  
}
```



МНОГОПОТОЧНЫЙ КОД

```
fun testStreaming(prompt: Prompt, expectedAnswer: String) {  
    webClient  
        .post()  
        .uri("/v2/streaming")  
        .body(Mono.just(ChatRequest("gpt-5.4", prompt)))  
        .body(Mono.just(ChatRequest("GPT-5.4", prompt)))  
        .exchange()  
        .expectStatus().isOk  
        .expectBody().json(expectedAnswer)  
}
```





BugSwarm (Pro)

- 43 нетривиальных бага
- Первопричина бага не фигурирует в стек трейсе

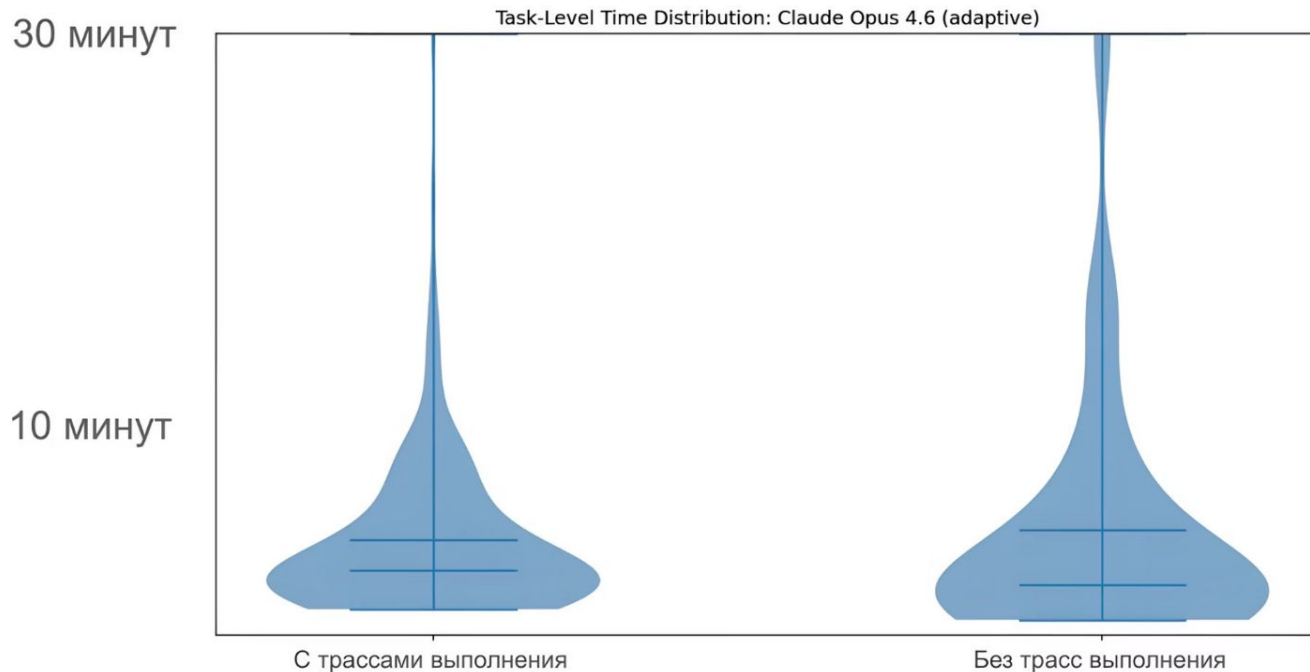
Veai AI Agent +

Claude Opus 4.6 (adaptive)

GPT 5.4 (xhigh)

DeepSeek V3.2 (AWQ)

Время на исправление одного бага



ЧТО СЛУЧИЛОСЬ,

ТО И ЦЕЛЬ 

Критерии успешности

1

Тест зеленеет

2

Тест зеленеет и
исправления в нужном
файле

3

Семантическая
эквивалентность
(LLM as a judge)

Промпт судьи

You are evaluating a generated diff for a Java bug-fix task. The gold diff shows the known correct fix. The generated diff is what an AI agent produced. Tests pass with the generated diff (formal verification succeeded).

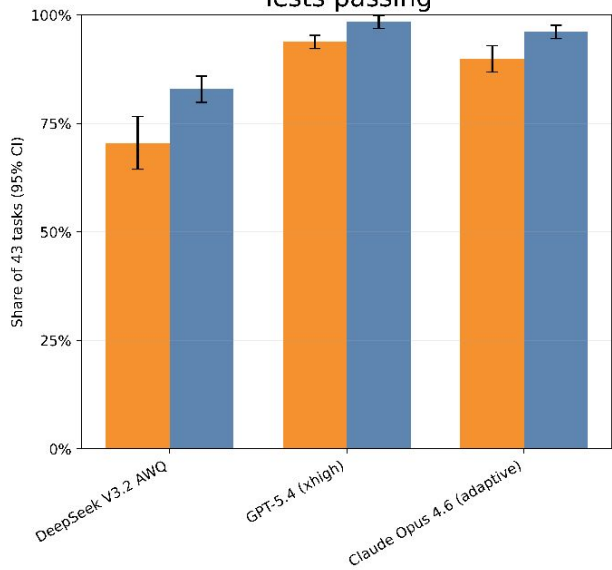
Classify the generated diff as either:

- SEMANTICALLY_CORRECT: The generated diff produces a correct fix that addresses the same root problem as the gold diff. The generated fix can be broader than the gold fix.
- SEMANTICALLY_INCORRECT: The generated diff makes tests pass but the fix is wrong: the fix will break under scenarios not covered by the current tests.

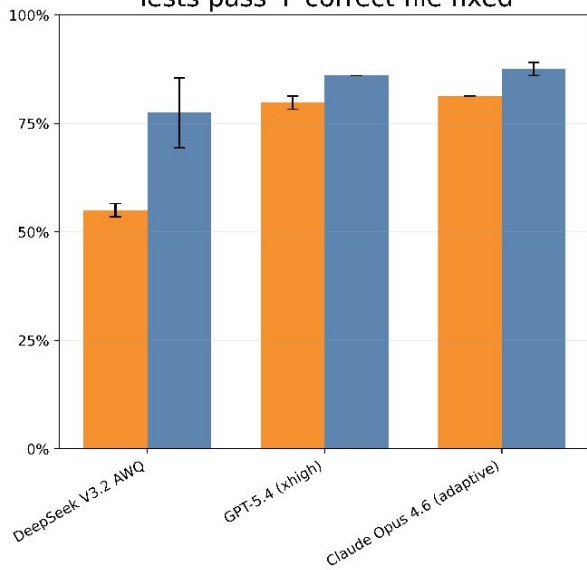
Reason step by step:

1. What does the gold diff fix and how?
2. What does the generated diff do?
3. Is the generated diff semantically correct?

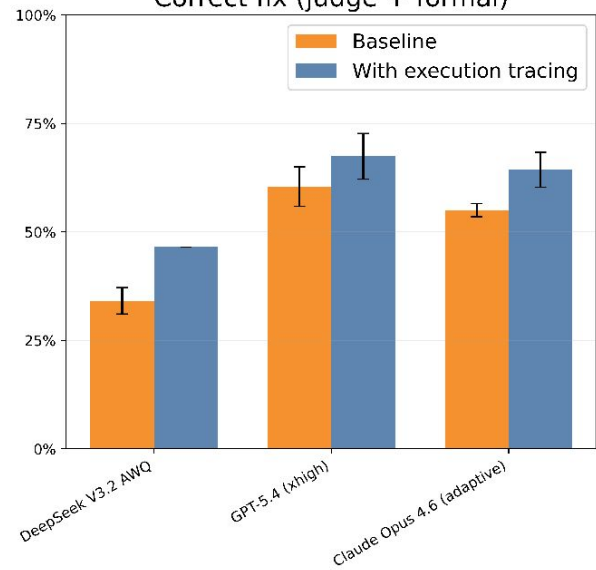
Tests passing

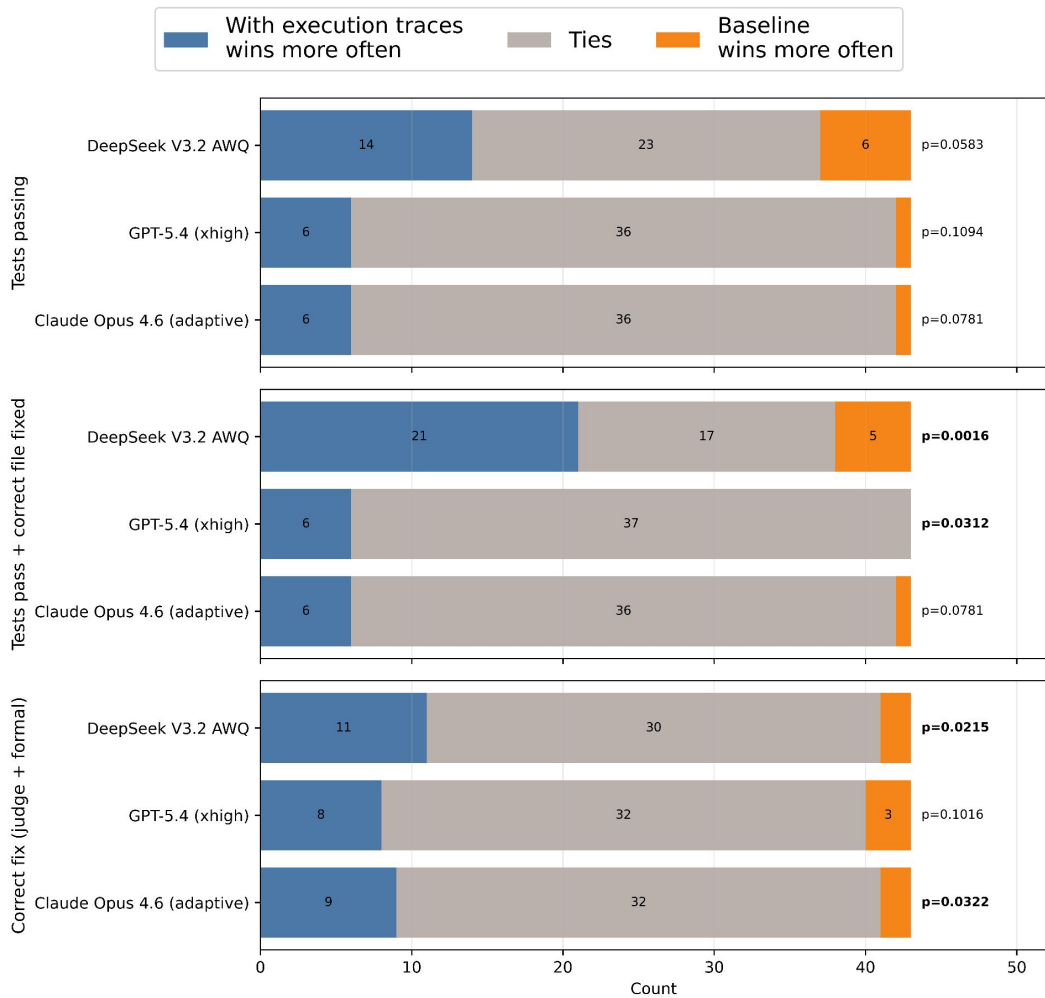


Tests pass + correct file fixed



Correct fix (judge + formal)





Bars show win/tie/loss counts for with execution traces relative to baseline. P-values are two-sided Wilcoxon.

 Start chatting with Instant

 Instant

 Expert

Message DeepSeek

 DeepThink

 Search

@

Что сегодня в повестке дня?

Спроси

+  Думаю



Некоторые выводы

1

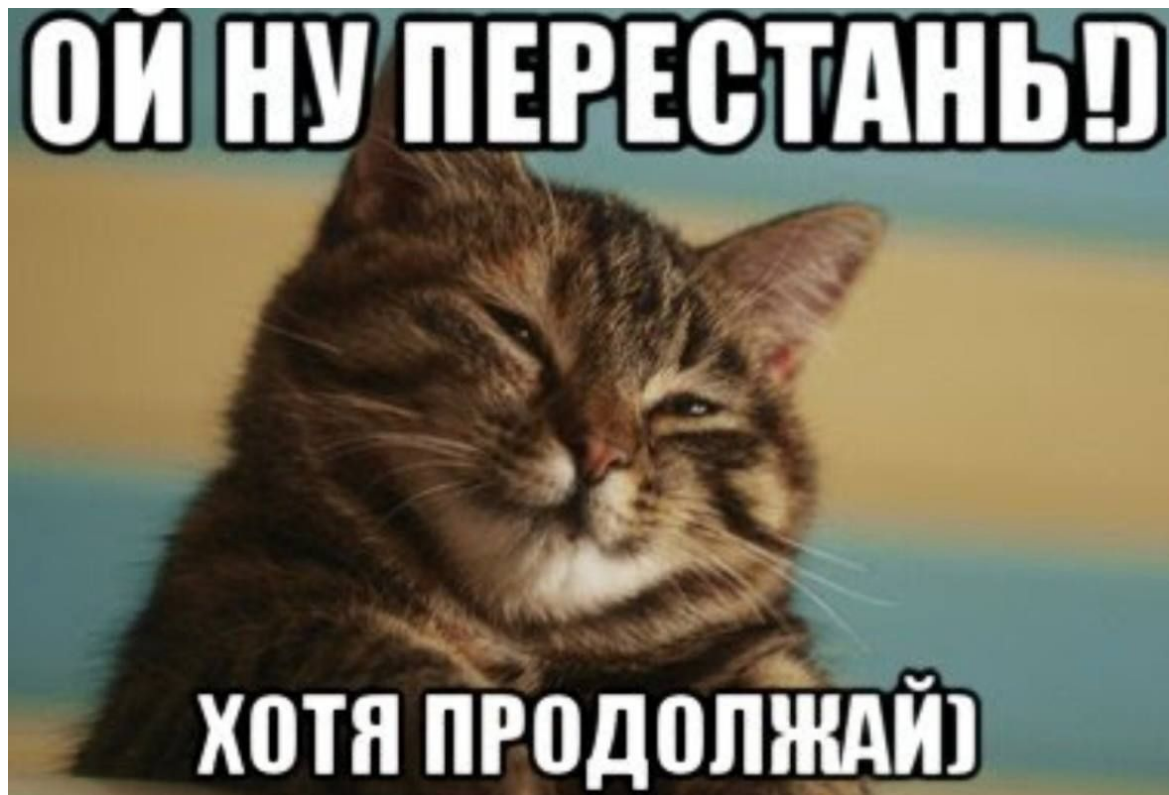
Если попросить AI-агента починить сложный баг, велик риск получить симптоматический фикс, усугубляющий реальную проблему.

2

Если дать AI-агенту содержательную часть трейса выполнения, вероятность получить корректный фикс повышается.

3

Выделение содержательной части трейса выполнения — задача, к сожалению, не самая простая.



Пример 4 на бис: необычный юзкейс

 Me

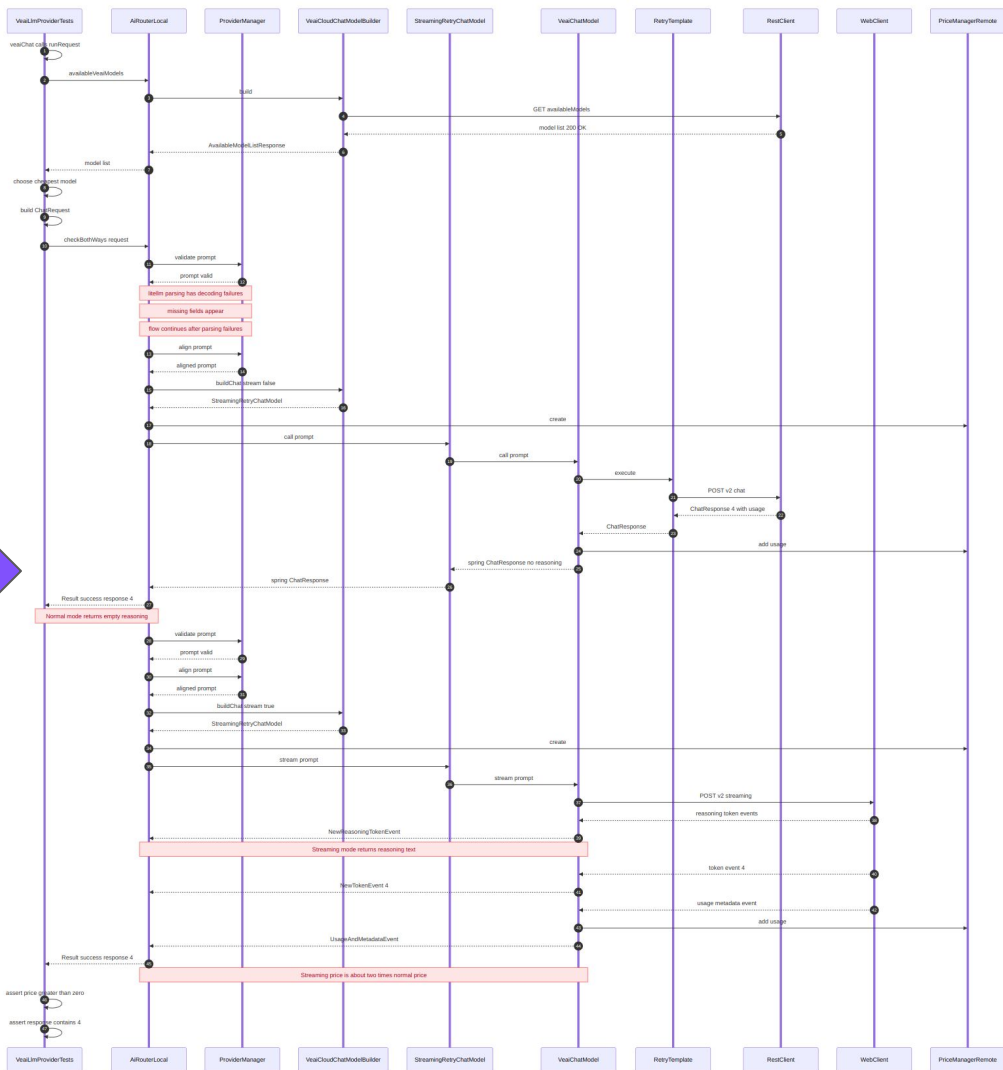


Запусти `VeaiLlmProviderTests.veaiChat()` с дампом дерева вызовов, полностью прочитай дамп и создай md-файл с mermaid диаграммой последовательностей.

Me

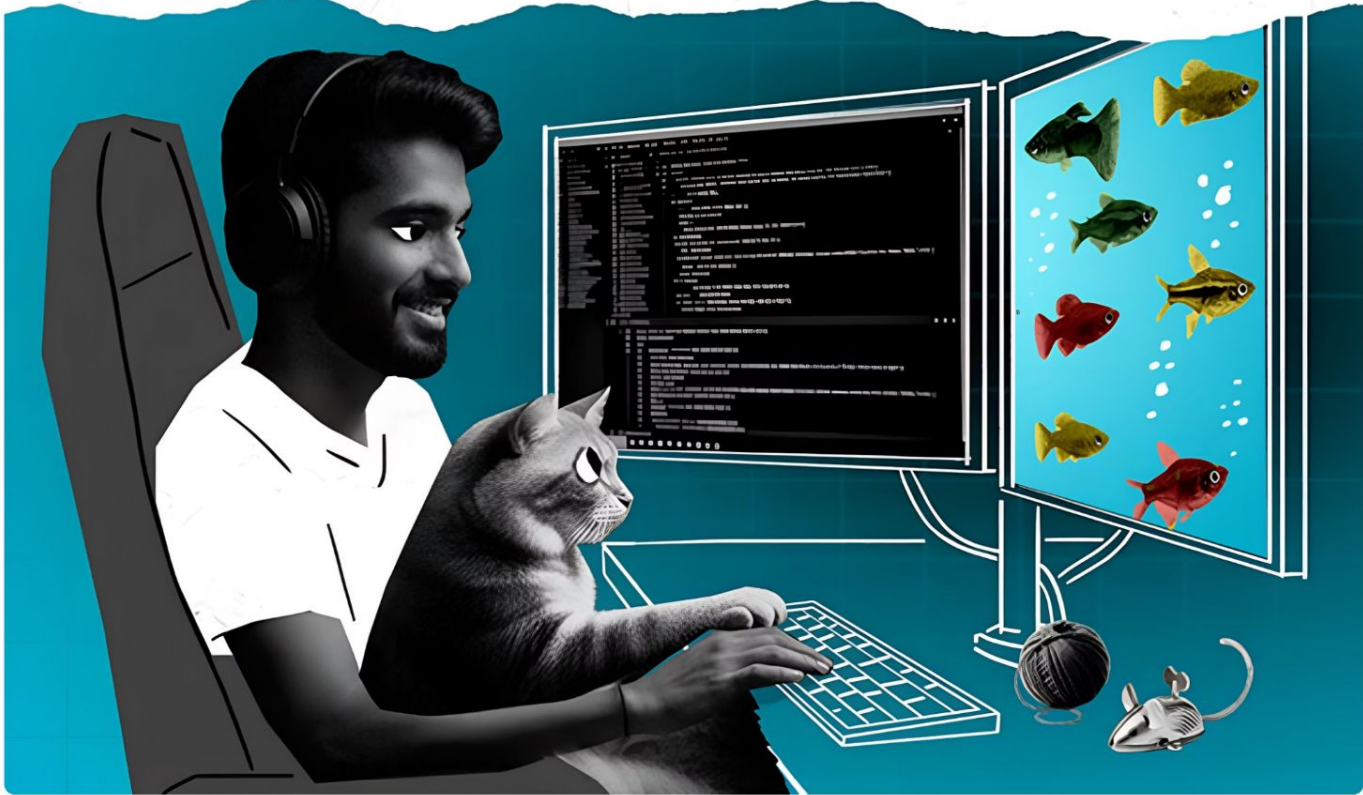


Запусти `VeaiLlmProviderTests.veaiChat()` с дампом дерева вызовов, полностью прочитай дамп и создай md-файл с mermaid диаграммой последовательностей.



Другие сценарии использования

- 1 — Ответы на вопросы: «А почему получается такой результат?»
- 2 — Ревью пулл реквеста через ревью изменений трассы выполнения
- 3 — Синхронизация документации и реального поведения





Спасибо за внимание!



Егор Куликов

Veai



Илья Муравьев

Veai