

# Кунсткамера инструментов для анализа кода от vi до vibe программирования

Современные проблемы разработчиков инструментов  
разработчика

Денис Фокин, LRI  
1/7/2025

# Обзор инструментов

**VI, Emacs** – мало поддержки  
(возможно, только c-tags).

**Borland** – богато, но тяжело весно и  
дорого.

**JetBrains** – быстро и дёшево

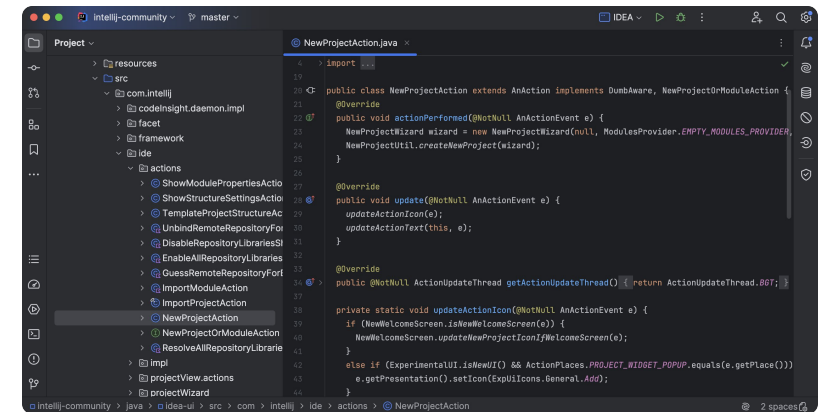
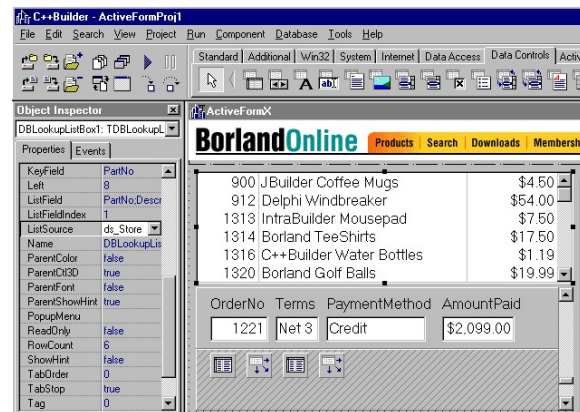
```
VIM - Vi IMproved

version 7.4.1848
by Bram Moolenaar et al.
Modified by <amazon-linux-ami@amazon.com>
Vim is open source and freely distributable

Help poor children in Uganda!
type :help iccf<Enter> for information

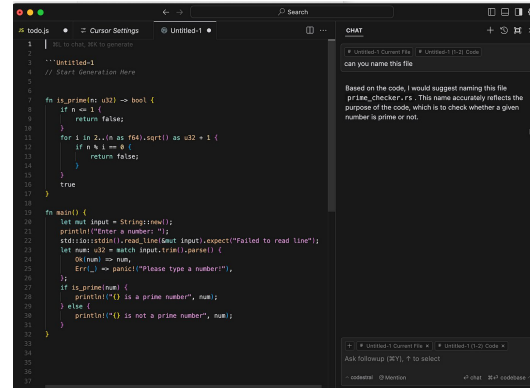
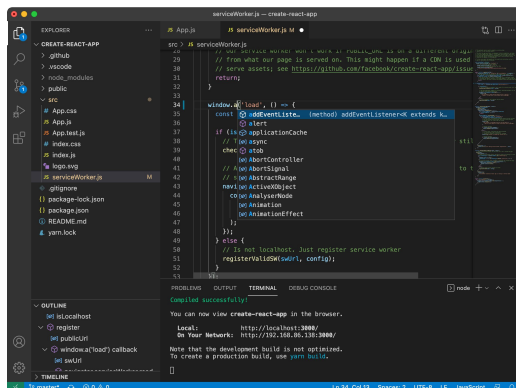
type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version7<Enter> for version info

0,0-1 All
```



**VS Code**

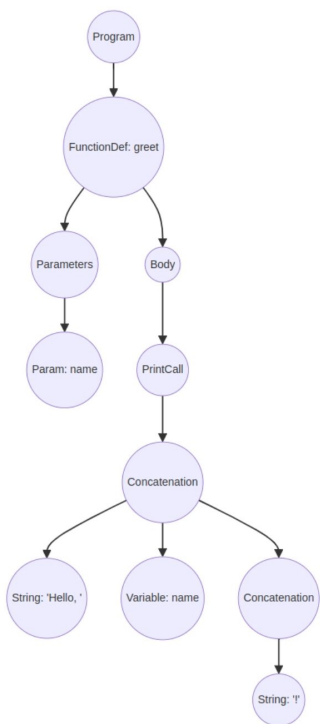
**Cursor**



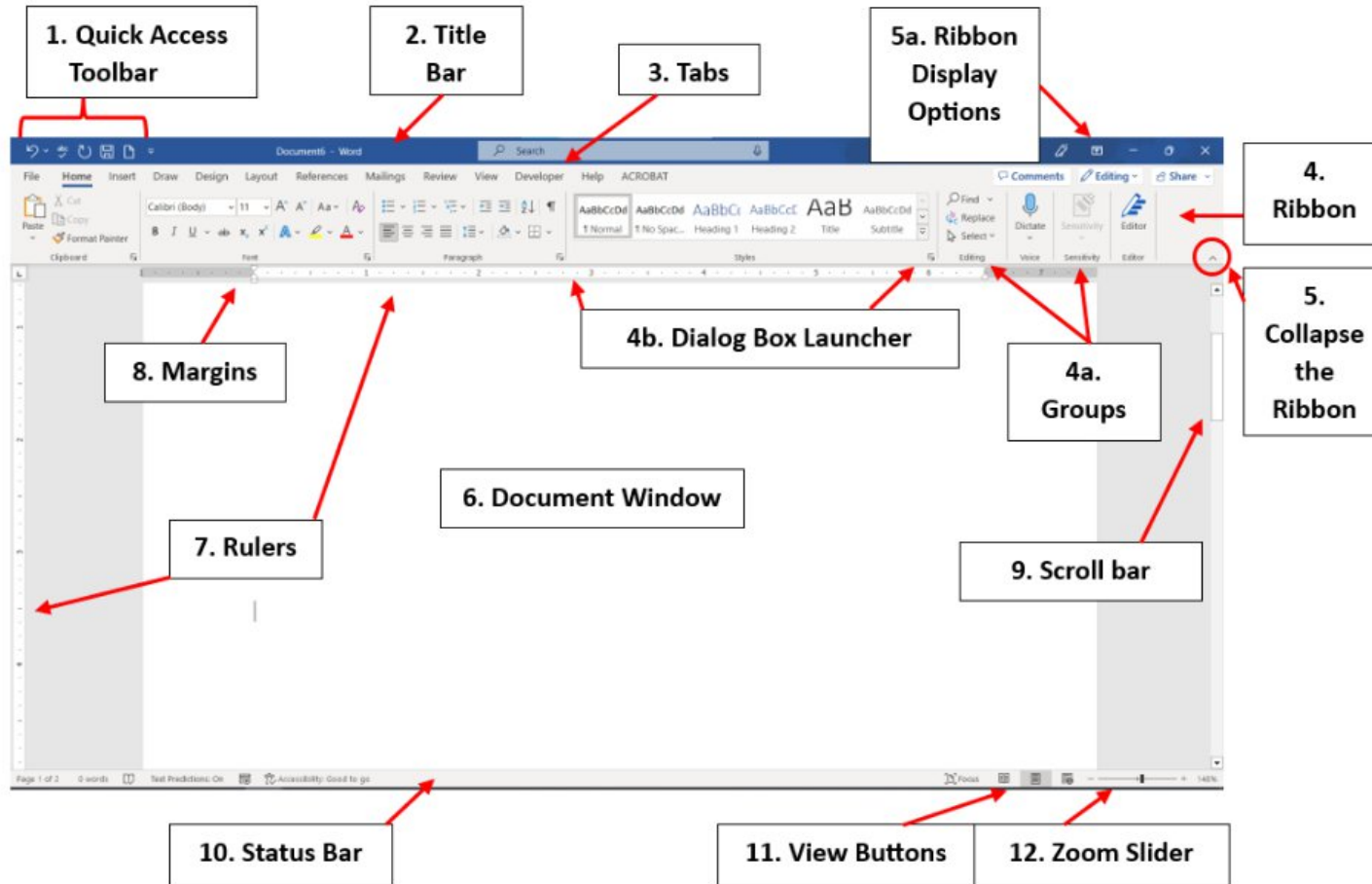
# Что нужно всем?

```
def greet(name):  
    print("Hello, " + name + "!")
```

- Парсер-трансформер-генератор
- Фронтендер – это навсегда



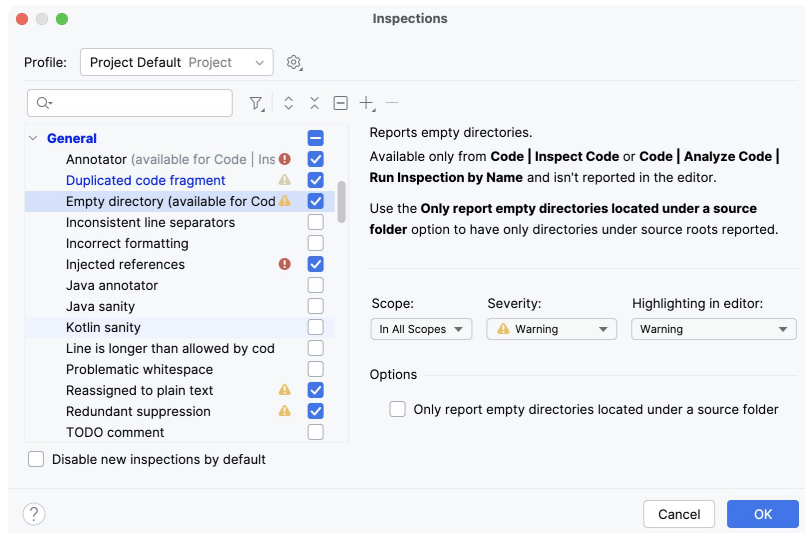
# Чего все хотят?



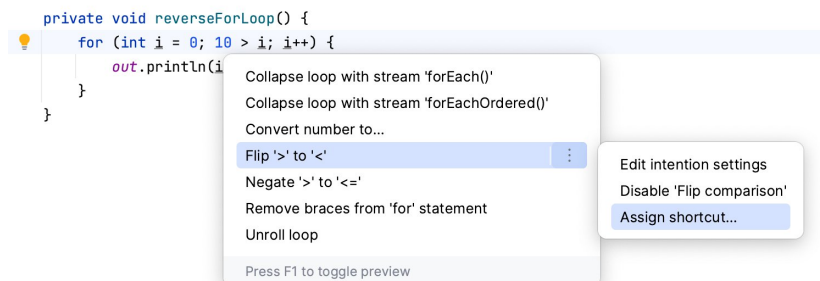
- Word
- Word для программистов – IDE

# Что все получают?

## Инспекции



## Предупреждая – исправляй. Auto-fixes.



## Радость -> Раздражение > Отключение

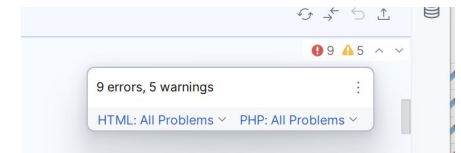


Gary

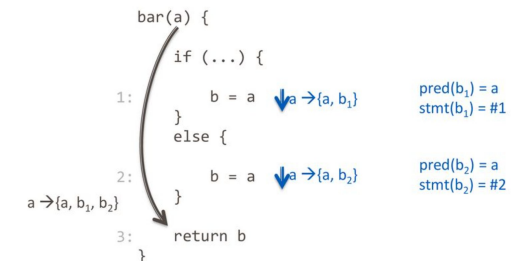
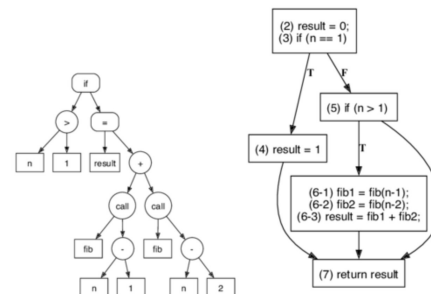
Created January 10, 2024 at 1:48 AM

It seems that this comes up from time to time but JetBrains is sticking by a statement made by Dmitry Jemerov over 10 years ago. For me, the inspections produce a load of false warnings - it doesn't correctly interpret/parse Codeigniter classes and numerous other issues (like telling me a single line PHP if statement is missing braces). I don't want to turn it off completely but I want it to only show me syntax errors. I've attached 3 screenshots showing what I would like to do as a default for all files I am editing. Is this still not possible after all these years?

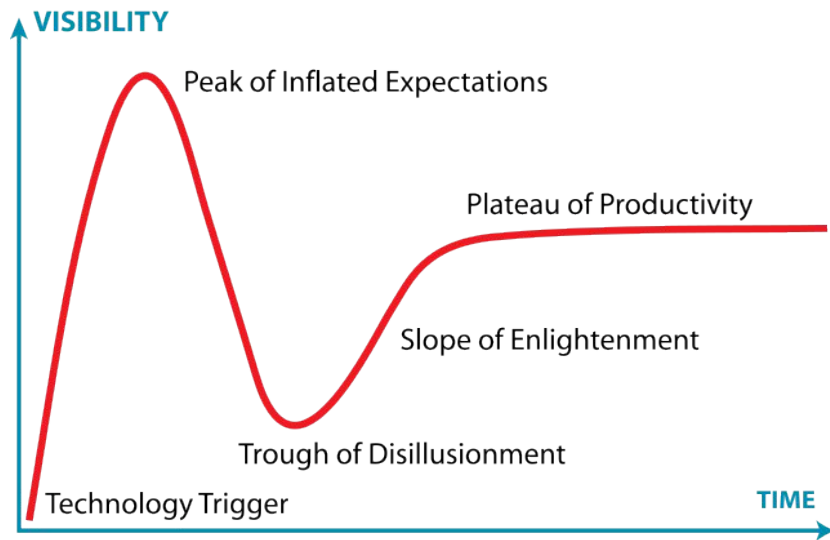
Snip 1 - what it defaults to.



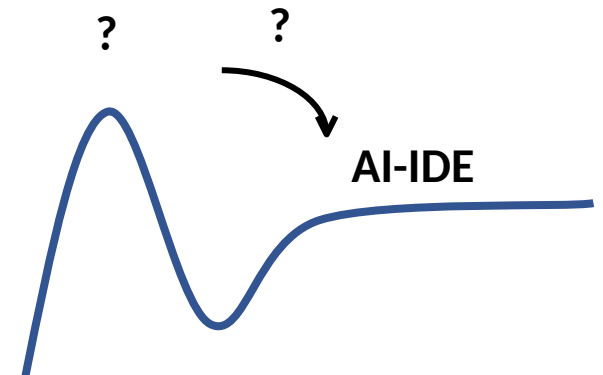
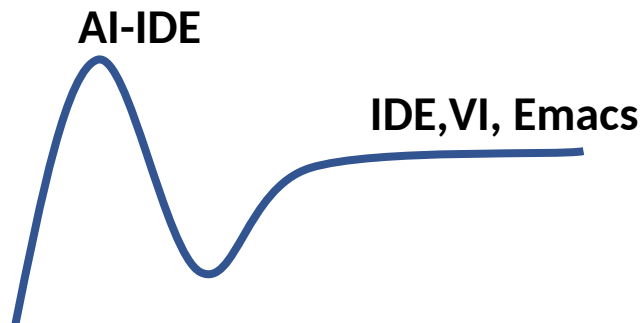
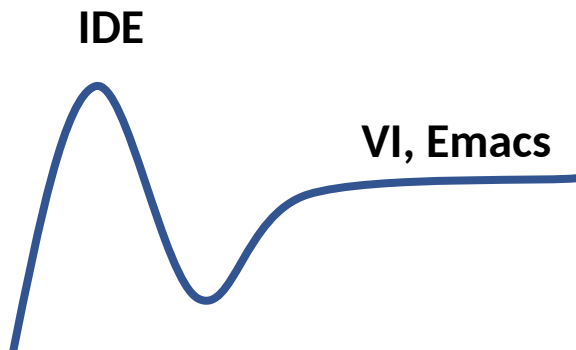
## AST недостаточно -> Граф потока управления -> Граф потока данных



# Диаграмма хайпа



- Все используют IDE -> многие используют vi
- Все смотрят на AI -> большинство используют IDE

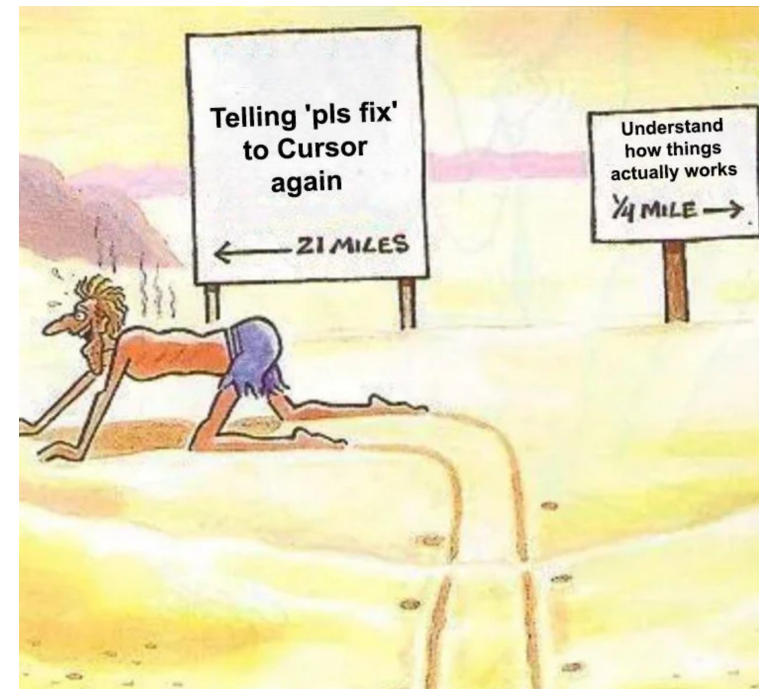


# Чего все хотят на самом деле?

Рассказал, чего хочу -> получил программу -> profit

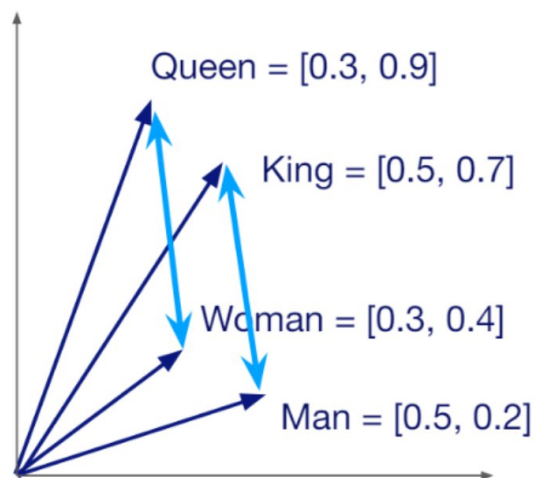


На самом деле



# Размер важен, контекст важен

Скажи кто твой друг и я скажу, кто ты



**King — Man + Woman = Queen**

$$\vec{v}_{king} - \vec{v}_{man} + \vec{v}_{woman} \approx \vec{v}_{queen}$$

Ищите женщину? Queen или queen? (пример о том, когда контекст неточный)



# Понимание кода и кодогенерация. Точность предсказания.

```
1  ▾ #ifdef _MSC_VERV
2    __declspec(noreturn)
3  ▾ #else
4    [[noreturnV]]
5  #endif
6    void container_abort(const char* pMsg, ...);
7
```

Abstract Syntax Tree:

```
translation_unit [0,0] - [6,0]
  preproc_ifdef [0,0] - [5,45]
    name: identifier [0,7] - [0,16]
    declaration [1,1] - [5,45]
      ms_declspec_modifier [1,1] - [1,21]
        identifier [1,12] - [1,20]
        ERROR [2,0] - [2,5]
      attribute_declaration [3,1] - [3,14]
        attribute [3,3] - [3,12]
          name: identifier [3,3] - [3,12]
          ERROR [4,0] - [4,6]
        type: primitive_type [5,1] - [5,5]
        declarator: function_declarator [5,6] - [5,44]
          declarator: identifier [5,6] - [5,21]
          parameters: parameter_list [5,21] - [5,44]
            parameter_declaration [5,22] - [5,38]
              type_qualifier [5,22] - [5,27]
              type: primitive_type [5,28] - [5,32]
              declarator: pointer_declarator [5,32] - [5,38]
                declarator: identifier [5,34] - [5,38]
```

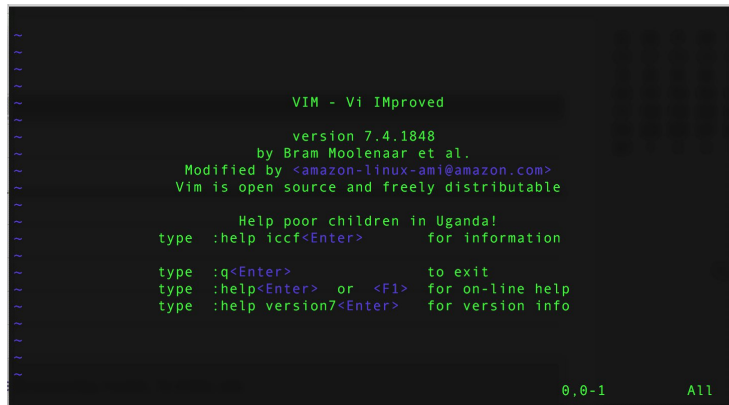
```
▾ #ifdef ANDROID_PLATFORM
  const char* tag = "ArkCompiler";
> void StdLog::PrintLog(LogLevel level, const char* fmt, ...)...
▾ #else
▾ #ifndef ENABLE_HILOG
  constexpr int32_t MAX_BUFFER_SIZE = 100;
> void StdLog::PrintLog(LogLevel level, const char* fmt, ...)...
#endif
#endif
```

Сложные запросы к структуре кода, сложно применить изменение к изначальному коду.

1. Можно попробовать использовать LLM
2. Окно LLM очень узкое для больших проектов
3. Нельзя просто резать код на куски

Сложно понять в чём проблема.

# Король контекста – контекст проекта.



VS



Почему многие разработчики продолжают пользоваться VI? Когда есть инструменты, которые позволяют прикладывать минимум усилий и получать работоспособную программу?

Разработчик готов писать код и без умного IDE, но не хотел бы разбираться в своём или чужом коде.

Наша задача – сделать кодогенерацию настолько точной, чтобы она не требовала отладки. Как это сделать – вопрос для обсуждения.

Обсуждение