

Санкт-Петербургский государственный электротехнический университет им.
В.И. Ульянова (Ленина)

Разработка инструмента объединения и интеграции статических анализаторов кода для языка C++

Выполнил:

Габов Егор Сергеевич, гр. 7381

Руководитель:

Самойленко В.П. к.т.н., доцент

Консультант:

Борисенко К.А., к.т.н., ст. преподаватель

Цель:

Создание инструмента, который позволяет интегрировать в среды разработки дополнительные статические анализаторы кода, помимо тех, которые уже есть в среде разработки.

Задачи:

- Разработать идею, которая позволит интегрировать разработанное решение в среды разработки (IDE)
- Разработать алгоритм работы инструмента
- Объединить результаты работы нескольких статических анализаторов кода
- Предоставить пользователю возможности настройки инструмента

Актуальность

Разработчики при написании кода допускают ошибки. Ошибки необходимо выявлять **как можно раньше**.

Время обнаружения дефекта	Цена исправления
Кодирование	1
Тестирование	10
Поддержка	10-25

Способы выявления ошибок:

- Code review
- Покрытие кода программ тестами
- Сборка программы с указанием компилятору максимального уровня предупреждений
- Статический анализ кода *во время написания кода*

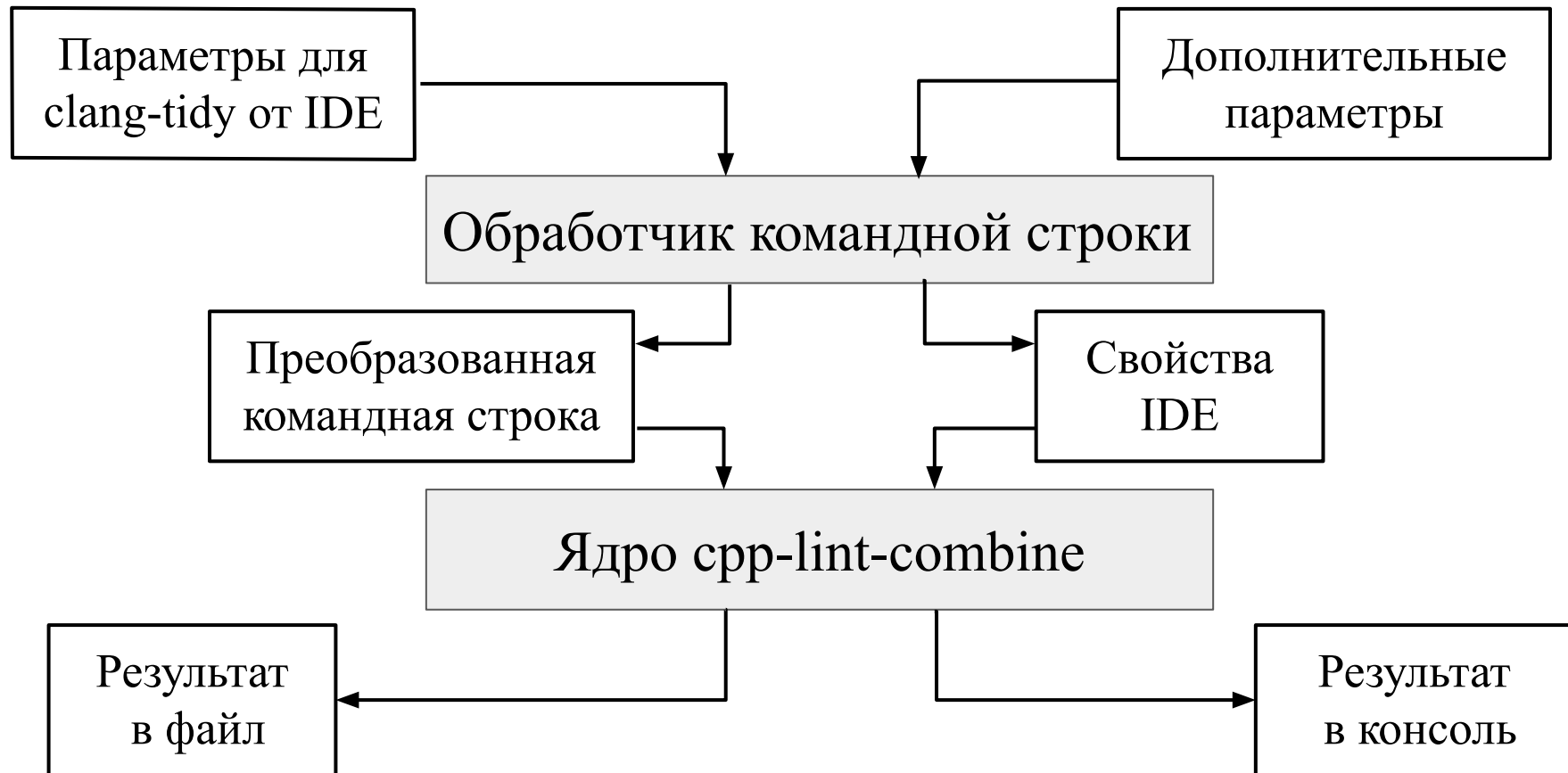
Статический анализ кода во время написания кода возможен благодаря интеграции статических анализаторов кода в среды разработки

Разработка способа интеграции в IDE

Задача: Научить IDE запускать вместо встроенного статического анализатора свою программу.

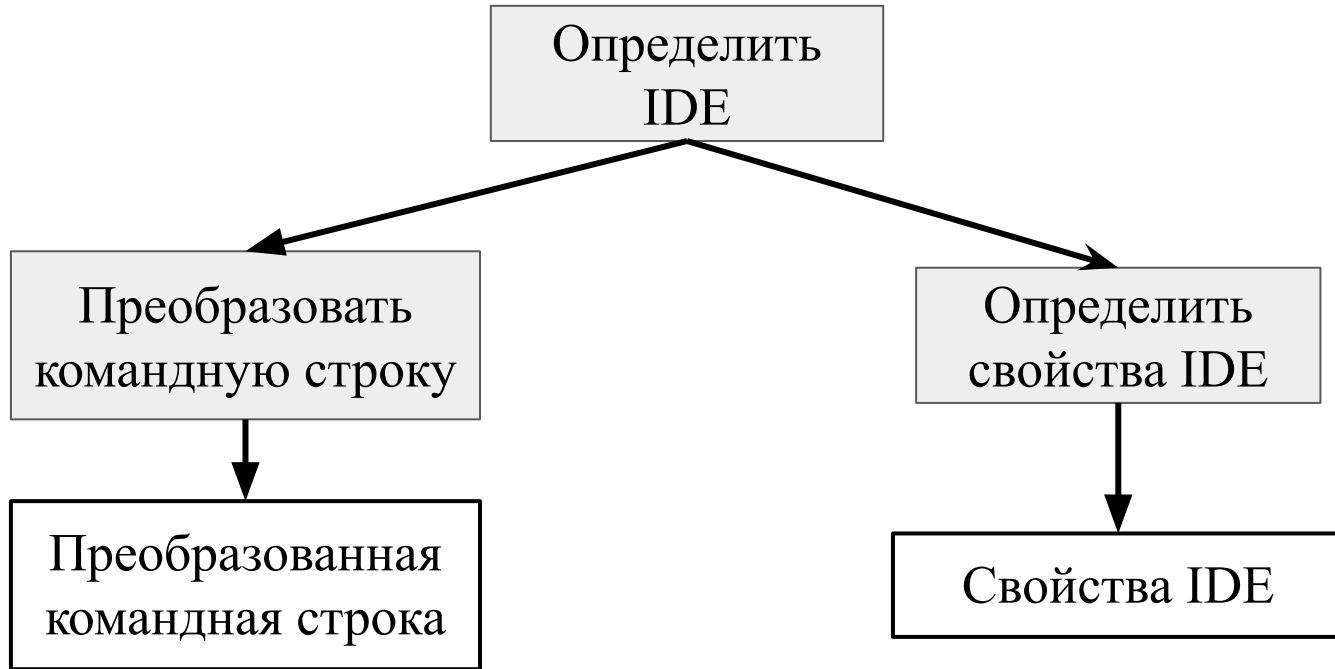
- Некоторые IDE позволяют прописывать путь встроенного анализатора кода
 - ReSharper C++
 - CLion
- В других IDE такой возможности нет
 - Visual Studio

Разработка алгоритма работы инструмента



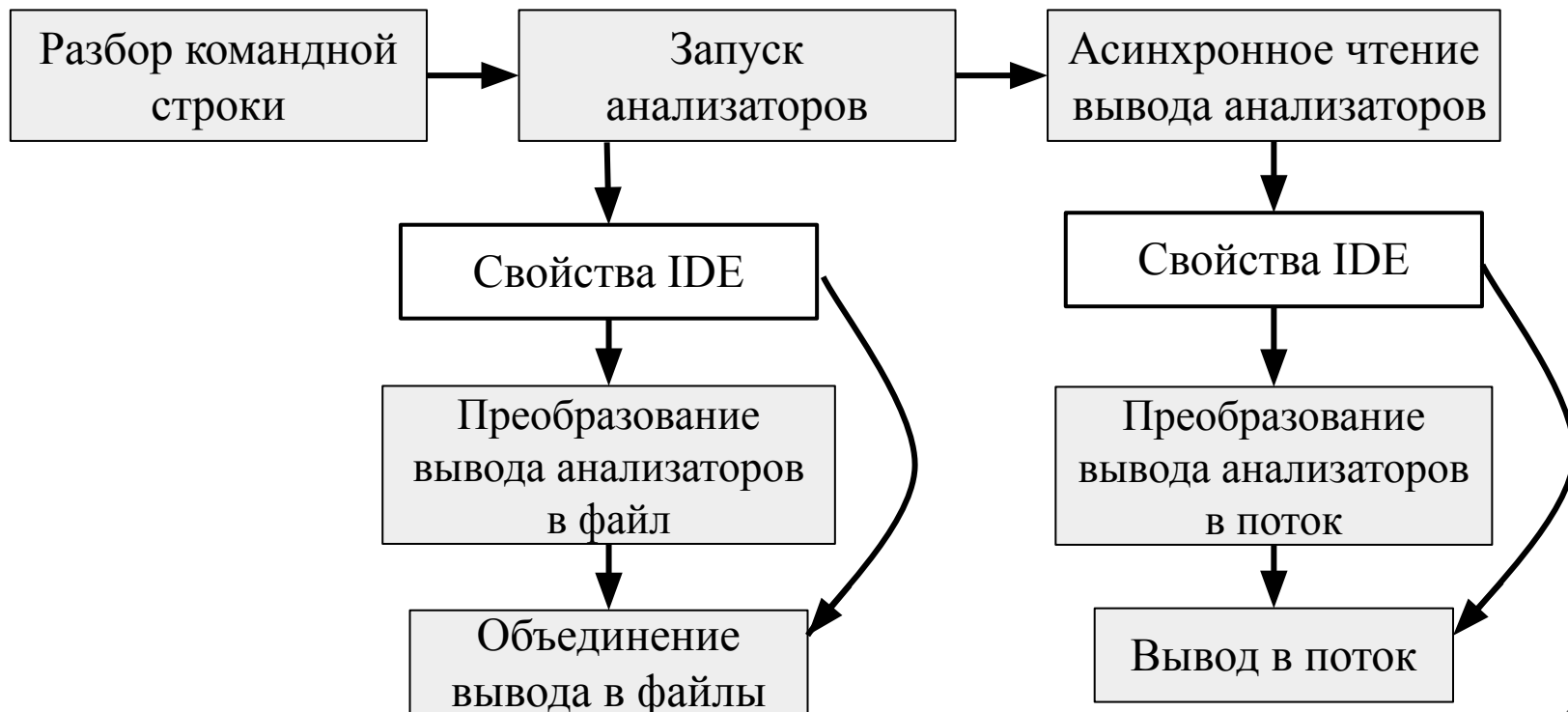
Разработка алгоритма работы инструмента

- Алгоритм работы обработчика командной строки



Разработка алгоритма работы инструмента

- Алгоритм работы ядра cpr-lint-combine



Объединение результата работы статических анализаторов

```
std::vector< SomeClass > vector;  
for ( int i = 0; i < 5; ++i )  
    for( int j = 0; j < 10; ++j )  
        vector.push_back( SomeClass( std::rand() ) );
```

Clang-tidy

5 is a magic number; consider replacing it with a named constant

Rand() has limited randomness; use C++11 random library instead

use `emplace_back` instead of `push_back`

statement should be inside braces

Clazy

Vector reserve candidate

Container inside loop causes unneeded allocations

Разработка способа настройки разработанного инструмента

Реализован пользовательский конфигурационный файл с возможностью указать следующие настройки:

- Имя IDE (CLion, ReSharper, BareMSVC)
- Путь до cpp-lint-combine
- Путь до clazy-standalone
- Путь до clang-tidy
- Список проверок clazy-standalone
- Дополнительные параметры для компилятора clang
- Используемые статические анализаторы

Апробация работы

1. Ссылка на github-репозиторий с разработанным инструментом - <https://github.com/WaveAccessEgorGabov/cpp-lint-combine/tree/dev/elop>
2. Доклад к научно-техническому семинару студентов и преподавателей каф. МО ЭВМ 2021
3. Публикация в [сборнике докладов студентов и аспирантов на научно-техническом семинаре кафедры МО ЭВМ](#), 2021 год.

Результаты

- Разработана идея для интеграции
- Разработан алгоритм работы инструмента
- Объединены результаты работы нескольких статических анализаторов кода
- Добавлена возможность настройки инструмента пользователем

Направления дальнейшей разработки

- Реализация интеграции в новые IDE
- Добавление новых анализаторов для объединения