Санкт-Петербургский государственный университет

Кафедра системного программирования

Интеграция инструментов оценки тестового покрытия кода в IDE для языков C/C++ на платформе Eclipse

Екатерина Андреевна Целовальникова

Научный руководитель: к.ф.-м.н. Д.В. Луцив, доцент кафедры системного программирования

Консультант: П.А. Бабанов, старший разработчик ООО «Софтком»

Санкт-Петербург
2025

Введение

- Компания ООО «Софтком» разрабатывает интегрированную среду разработки для C/C++ на базе Eclipse
- Eclipse — платформа для разработки ПО
  - предоставляет инструменты для проектирования, отладки и тестирования
  - обладает модульной архитектурой
  - поддерживает интеграцию плагинов
- Анализ покрытия кода повышает качество ПО
  - выявляет непокрытые участки кода
  - повышает надёжность и помогает выявить уязвимости
- Разрабатываемое решение — плагин для интеграции в создаваемую среду

Постановка задачи

Цель: Интегрировать инструменты оценки тестового покрытия кода в IDE для языков C/C++ на платформе Eclipse

Задачи

- Провести обзор существующих плагинов
- Разработать механизм конфигурации запуска анализа
- Разработать GUI
  - реализовать отображение результатов анализа
  - разработать страницу глобальных настроек
  - разработать страницу проектных настроек
- Реализовать подсветку строк кода
- Провести тестирование и апробацию

Требования к решению

- Независимость от CDT
- Поддержка компилятора Clang
- Настройка анализа покрытия
- Отчётность и визуализация

Существующие решения

- Linuxtools GCov [1]
- Testwell CTC++ [2]
- Parasoft C/C++test [3]

Ограничение: Используют CDT в качестве базовой инфраструктуры

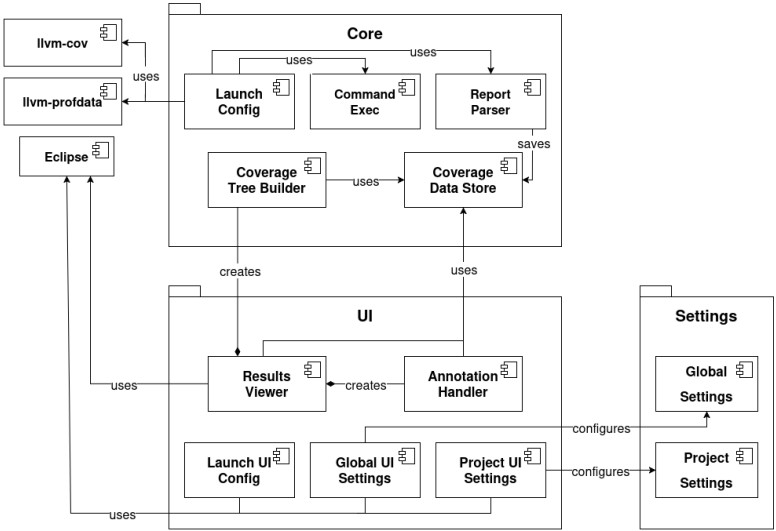
[1] https://projects.eclipse.org/projects/tools.linuxtools
[2] https://www.verifysoft.com/en_ctcpp.html
[3] https://www.parasoft.com/products/parasoft-c-ctest

Анализ существующих решений

Категория	LinuxTools GCov	Testwell CTC++	Parasoft C/C++test
Визуализация результатов	SWT, JFace, BIRT	HTML-отчёты	Parasoft C++test perspective, HTML/PDF/XML-отчёты
Взаимодействие с инструментом	CDT Build Settings (флаги компиляции)	CDT Build Settings (инструментация через ctc)	CDT Build Settings
Конфигурация	CDT Build Settings (параметры компиляции)	CDT Build Settings (настройка ctc)	CDT Build Settings
UI/UX	JFace, Eclipse Action Framework	HTML-отчёты	Parasoft C++test perspective

Вывод: SWT, JFace и Eclipse Action Framework позволяют создать независимое решение с минимальной зависимостью от среды

Архитектура разрабатываемого плагина



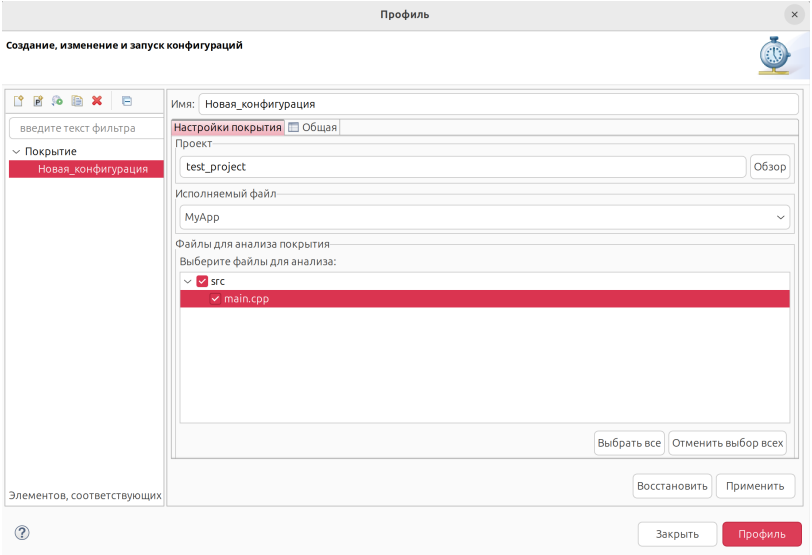
Создание конфигурации запуска [1/2]

Конфигурация запуска — это набор параметров и настроек для выполнения тестов и запуска анализа покрытия кода

Функции делегата запуска

- Определяет ресурсы для анализа покрытия
- Собирает проект с флагами для сбора данных покрытия
- Выполняет тесты с сохранением сырых данных покрытия
- Создаёт промежуточный профиль покрытия из сырых данных
- Формирует отчёт с использованием бинарного файла и промежуточного профиля
  - Формат — `.lcov`

Создание конфигурации запуска [2/2]



Пример отчёта

```
SF:src/sample.cpp     % SF: путь к файлу
FN:15,_Z3sumii        % FN: функция и номер строки
FNDA:1,_Z3sumii       % FNDA: количество вызовов функции
FNF:1                 % FNF: всего функций
FNH:1                 % FNH: покрыто функций
DA:15,4               % DA: покрытие строки
DA:16,0               % (номер строки, кол-во выполнений)
BRDA:16,0,0,1         % BRDA: покрытие ветвления
BRDA:16,0,1,0         % (строка, блок, ветвь, кол-во выполнений)
BRF:2                 % BRF: всего ветвлений
BRH:1                 % BRH: покрыто ветвлений
LF:2                  % LF: всего строк
LH:1                  % LH: покрыто строк
end_of_record         % конец записи
```

Отображение результатов анализа покрытия

Консоль | Результаты покрытия

Элемент	Покрытие	Покрытые строки	Пропущенные строки	Всего строк
src	100,00%	22	0	22
main.cpp	100,00%	22	0	22
add	100,00%	3	0	3
subtract	100,00%	3	0	3
factorial	100,00%	9	0	9
main	100,00%	7	0	7

Консоль | Результаты покрытия

- Счетчики строк
- Счетчики ветвлений
- Счетчики функций

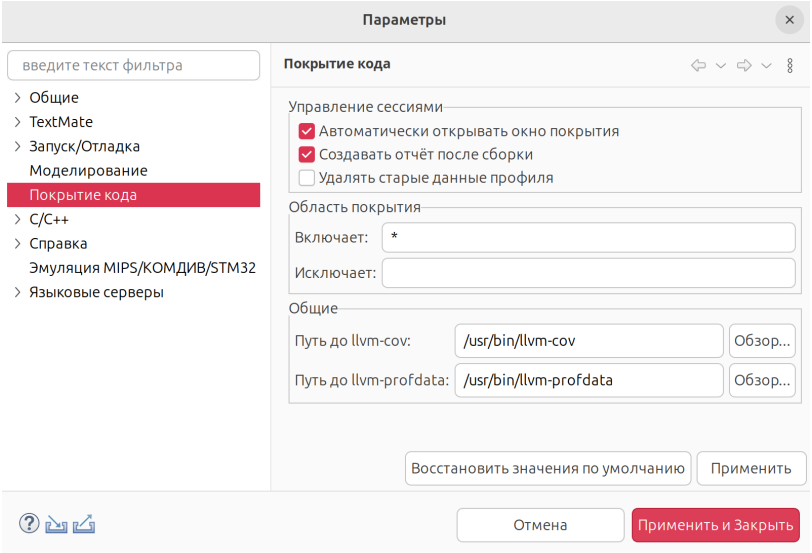
Элемент	Покрытие	Покрытые строки	Пропущенные строк	
src	100,00%	22	0	
main.cpp	100,00%	22	0	
add	100,00%	3	0	3
subtract	100,00%	3	0	3
factorial	100,00%	9	0	9
main	100,00%	7	0	7

Подсветка строк

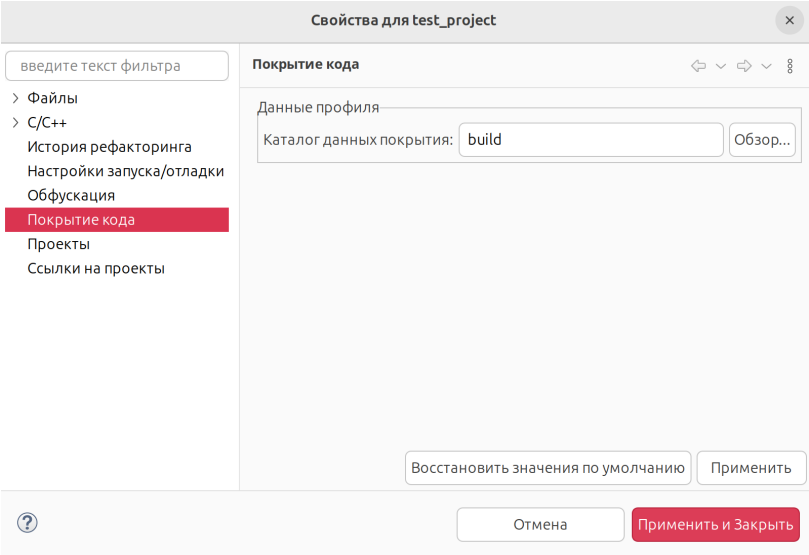
main.cpp

```cpp
#include <iostream>
#include <stdexcept>

class Math {
public:
    int add(int a, int b) const {
        return a + b;
    }

    int subtract(int a, int b) const {
        return a - b;
    }

    int factorial(int n) const {
        if (n < 0) throw std::invalid_argument("Negative input for factorial is not allowed");
        if (n == 0) return 1;
        int result = 1;
        for (int i = 1; i <= n; ++i) {
            result *= i;
        }
        return result;
    }
};

#ifndef
int main() {
    Math math;
    std::cout << "5 + 3 = " << math.add(5, 3) << std::endl;
    std::cout << "5 - 3 = " << math.subtract(5, 3) << std::endl;
    std::cout << "Factorial of 5 = " << math.factorial(5) << std::endl;
    return 0;
}
#endif
```

Страница глобальных настроек



Страница проектных настроек



Тестирование

Проведено ручное тестирование 10 открытых проектов на C/C++ с GitHub для оценки анализа покрытия кода и интеграции с Eclipse

- Критерии выбора проектов
  - сборка с использованием CMake
  - компиляция компилятором Clang
  - наличие тестов для анализа покрытия
- Сценарии тестирования и результаты
  - корректность конфигурации запуска: 95,1% успешно
  - стабильность обработки профильных данных: 93,4% успешно
  - корректность отображения дерева результатов: 91,8% успешно
  - корректность подсветки строк в редакторе: 89,3% успешно
  - работоспособность глобальных и проектных настроек: 100% успешно
- Итог: 93.9% тестов успешны

Апробация

- Проведена презентация в офисе «Софтком»
  - получены рекомендации по улучшению GUI
  - отмечены интуитивность интерфейса и наглядность результатов
- Протестировано консультантом на закрытых проектах заказчика

Результаты

В процессе работы были выполнены следующие задачи

- Проведён обзор существующих плагинов
- Разработан механизм конфигурации запуска анализа
- Реализован GUI
  - реализовано отображение результатов анализа
  - разработана страница глобальных настроек
  - разработана страница проектных настроек
- Реализована подсветка строк кода
- Проведены тестирование и апробация

Программный код выполненной работы доступен на GitHub [4]

[4] https://github.com/softcom-su/CLDT-extensions/pull/2