

Санкт-Петербургский государственный политехнический
университет
Институт информационных технологий и управления
Кафедра компьютерных систем и программных технологий



ВЫПУСКНАЯ РАБОТА БАКАЛАВРА

Тема: **Разработка модуля расширения статических
проверок для компилятора Clang**

Студент гр. 43501/3 Н.А. Салин

Санкт-Петербургский государственный политехнический
университет
Институт информационных технологий и управления
Кафедра компьютерных систем и программных технологий

Диссертация допущена к защите
зав. кафедрой

_____ В.Ф. Мелехин

«____» _____ 2015 г.

ВЫПУСКНАЯ РАБОТА БАКАЛАВРА

Тема: **Разработка модуля расширения статических
проверок для компилятора Clang**

Направление: 230100 – Информатика и вычислительная техника

Выполнил студент гр. 43501/3

_____ Н.А. Салин

Научный руководитель,
м.т.т., аспирант

_____ М.Х. Ахин

Рецензент,
к. т. н., доц.

_____ Р.Е. Цензент

Консультант по нормоконтролю,
к. т. н., доц.

_____ Р.Е. Вьюер

Эта страница специально оставлена пустой.

РЕФЕРАТ

Отчет, 19 стр., 1 прил.

CLANG, AST, СТАТИЧЕСКИЙ АНАЛИЗ, PLUGIN

Что тут должно быть?

ABSTRACT

Report, 19 pages, 1 appendicies

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1. СТАТИЧЕСКИЕ АНАЛИЗАТОРЫ	9
1.1. Описание и основные задачи статических анализаторов	9
1.2. Существующие статические анализаторы	9
1.2.1. PVS-Studio	9
1.2.2. Coverity	9
1.2.3. Flint	9
1.2.4. Clang Static Analyzer	9
1.2.5. Анализатор в Clang	9
2. КОМПИЛЯТОР CLANG И МОДУЛИ ДЛЯ НЕГО	11
2.1. О Clang	11
2.2. AST	11
2.3. Plugin	11
2.4. AST Matcher	11
3. РАЗРАБОТКА МОДУЛЯ ДЛЯ CLANG	13
3.1. Список доступных проверок	13
3.2. Диаграмма классов	13
3.3. Динамическое подключение проверок	13
3.4. Нахождение ошибки	13
3.5. Добавление новых проверок	13
4. ТЕСТИРОВАНИЕ И АНАЛИЗ РАЗРАБОТАННОГО МОДУЛЯ	15
4.1. Тестирование	15
4.2. Анализ	15
ЗАКЛЮЧЕНИЕ	17
ПРИЛОЖЕНИЕ А. ЛИСТИНГИ	19

ВВЕДЕНИЕ

Описать что такое ошибки в коде, какие существуют средства нахождения ошибок. Рассказать что в данной работе будет использоваться статический анализ.

1. СТАТИЧЕСКИЕ АНАЛИЗАТОРЫ

1.1. Описание и основные задачи статических анализаторов

Тут будет о том что такое статический анализатор, для чего нужен и какие ошибки может находить.

1.2. Существующие статические анализаторы

Существует множество статических анализаторов кода, у каждого есть свои плюсы и минусы. Ниже будут рассмотрены наиболее популярные.

1.2.1. PVS-Studio

1.2.2. Coverity

1.2.3. Flint

1.2.4. Clang Static Analyzer

1.2.5. Анализатор в Clang

2. КОМПИЛЯТОР CLANG И МОДУЛИ ДЛЯ НЕГО

2.1. О Clang

Тут будет описанно что такое Clang и почему он был выбран

2.2. AST

Написать что это такое, рассказать об Clang AST

2.3. Plugin

Что это такое, как подключается и как происходит обход AST.

2.4. AST Matcher

Рассказать что это, и пример сравнения нахождения ошибки с помощью простого Visit* и ASTMatcher

3. РАЗРАБОТКА МОДУЛЯ ДЛЯ CLANG

3.1. Список доступных проверок

Тут будет описание проверок, которые делает анализатор и примеры неверного кода

3.2. Диаграмма классов

Тут будет приведена диаграмма классов и описано как классы взаимодействуют

3.3. Динамическое подключение проверок

Тут написать, что можно включать только необходимые проверки и делается это с помощью флагов для плагина.

3.4. Нахождение ошибки

Тут будет описание того, что каждый класс при вызове функции `addToFinder` добавляет необходимый `astmather` и `callback` для обработки результата.

3.5. Добавление новых проверок

Тут рассказать, что для добавления новой проверки нужно создать новый класс, написать `astMather` и не забыть добавить в `matchFinder`

4. ТЕСТИРОВАНИЕ И АНАЛИЗ РАЗРАБОТАННОГО МОДУЛЯ

4.1. Тестирование

Показать нахождение ошибок. Показать в каких случаях происходит ложное срабатывание и когда наоборот не происходит нахождения ошибки.

4.2. Анализ

Тут будет сравнение скорости компиляции с плагином, без него и с частично отключенными проверками.

ЗАКЛЮЧЕНИЕ

Заключение

ПРИЛОЖЕНИЕ А

ЛИСТИНГИ