

# Реализация расширенного препроцессора для проекта PySi

Аникин Егор Георгиевич

7 июня 2020 г.

Научный руководитель: д.ф.-м.н., проф. А.Н. Терехов

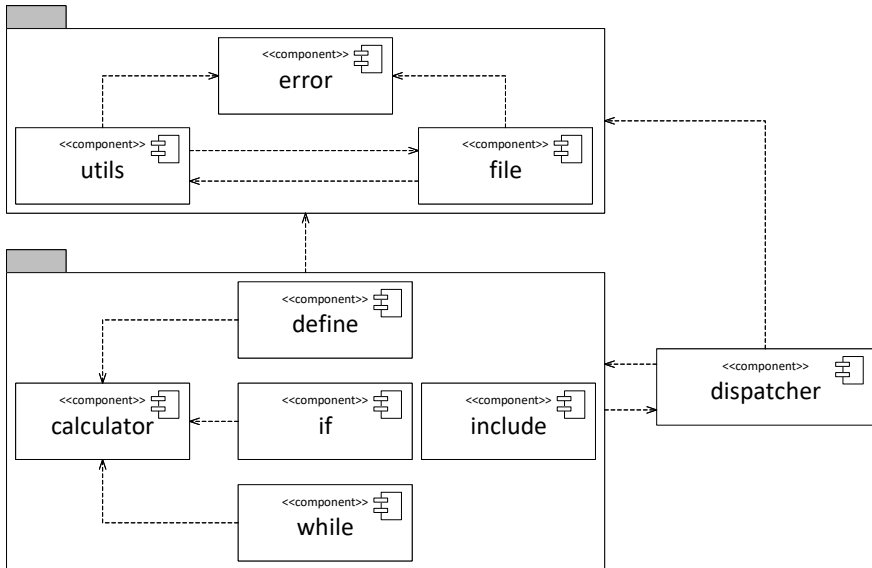
Рецензент: Директор ООО "Новые Мобильные  
Технологии" В. В. Оносовский

Цель работы — создать препроцессор для языка РуСи.

Задачи:

- спроектировать архитектуру
- реализовать базовую и расширенную функциональность препроцессор
- реализовать возможность детектирования ошибок препроцессора
- реализовать возможность добавления файлов
- провести тестирование и апробацию

- #include — добавление файлов
- #define — определение простых макросов, возможно, с параметрами
- #if, #ifdef и #ifndef — директивы условной компиляции
- #undef — удаление макроса
- #macro — определение сложных макросов, возможно, с вложенными директивами
- #eval — счёт на уровне препроцессора
- #while — цикл препроцессора с условием
- #set — переопределение простого макроса



Основные принципы:

- обработка текста итератором
- обработка директив методом рекурсивного спуска
- раскрытие макроса

Основные особенности:

- объединение файлов в один текст
- предварительный проход для поиска заголовочных файлов
- `#include`
- командная строка
- область видимости

Было:

- определение места ошибки
- вывод строки с ошибкой до нужного символа
- сообщение о типе ошибки

Изменилось:

- информация о файле
- пересчёт строк с учётом работы препроцессора
- контроль использования директив
- контроль раскрытия макросов

Как встроить в другой язык:

- изменить вывод ошибок
- изменить формат ввода/вывода

Текущий интерфейс:

- вход: файлы с исходным кодом
- выход: текст с исходным кодом, не содержащий директивы препроцессора, и сопутствующая информация



## Пример «Объявление идентификаторов» до обработки

```
#define A 5
#define B 3
#define C 7
#define case(t) concat(concat(int a,t),;)
#if #eval(A+B) == #eval(C + 1) || A > C
    #define j 0
    #while j < A
        case(#eval(j + B))
        #set j #eval(j+1)
    #endw
    #undef j
#endif
```

## Пример «Объявление идентификаторов» после обработки

```
int a3;  
int a4;  
int a5;  
int a6;  
int a7;
```

```
#macro FACTOR(N)
  #if #eval(N) >= 2
    #eval(N * FACTOR(N - 1))
  #else
    1
  #endif
#endm
```

## Сделано:

- спроектирована архитектура препроцессора
- реализованы все описанные директивы
- реализована возможность добавления файлов
- проведена корректировка вывода ошибок Руси и реализован вывод собственных ошибок препроцессора
- проведено тестирование и апробация