МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

САНКТ**-**ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**,** МЕХАНИКИ И ОПТИКИ

Кафедра Информатики и прикладной математики

Лабораторная работа №2 (8 вариант)

По предмету «Верификация моделей программ»

Выполнил:

студент гр. № Р4115

Назукин Д. Е.

Проверил :

Кореньков Ю. Д.

Санкт-Петербург

2017

**Цели**

Освоение верификации алгоритмической последовательности модели программы.

**Задачи**

Используя анализатор от первого задания реализовать программу, принимающую в качестве входных аргументов командной строки список имен N входных файлов и путь к выходной директории. Для заданных входных файлов с текстами на соответствующем варианту языке программа должна формировать в выходной директории выходные файлы, содержащие представление графа потока управления, описывающего последовательность выполнения операций, заданных инструкциями и выражениями соответствующей части исходного текста. В той же директории должен формироваться дополнительный файл, содержащий представление графа вызовов для всей совокупности входных файлов, описывающее отношения между функциями/методами входных файлов в плане обращению их друг к другу по именам.

**Описание работы**

Граф потока управления ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) control flow graph, CFG) — в [теории компиляции](https://ru.wikipedia.org/w/index.php?title=%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%BA%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%86%D0%B8%D0%B8&action=edit&redlink=1) — множество всех возможных путей исполнения программы, представленное в виде [графa](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)" \o "Граф (математика)).

В графе потока управления каждый узел (вершина) графа соответствует [базовому блоку](https://ru.wikipedia.org/wiki/Basic_block) — прямолинейному участку кода, не содержащему в себе ни операций передачи управления, ни точек, на которые управление передается из других частей программы. Имеется лишь два исключения:

* точка, на которую выполняется переход, является первой инструкцией в базовом блоке;
* базовый блок завершается инструкцией перехода.

Направленные дуги используются в графе для представления инструкций перехода. Также, в большинстве реализаций добавлено два специализированных блока:

* входной блок, через который управление входит в граф;
* выходной блок, который завершает все пути в данном графе.

**Аспекты реализации**

Граф потока управления строится по абстрактному синтаксическому дереву. Для обхода дерева был реализован паттерн Посетитель (англ. visitor). В графе есть либо обычные узлы, либо узлы в которых происходит разветвление потока. Разветвление начинается в следующих случаях:

* Условие if
* Цикл while
* Цикл Do while
* Цикл Do until

**Результаты**

Вход программы файл с текстом:

function calc(A as long, B as long, Op as char)  
 if (Op == '+') then  
 calc = sum(A, B);  
 end if  
 if (Op == '-') then  
 calc = minus(A,B);  
 end if  
end function  
  
function minus(A as long, B as long)  
 minus = (A-B);  
end function  
  
function sum(A as long, B as long)  
 sum = A+B;  
end function

Результат:

Файл calc.txt

start -> 0  
0 t-> 1  
1 -> 2  
2 t-> 3  
3 -> end  
2 f-> end  
0 f-> 2  
  
0:(Op=='+')  
1:calc=sum(A,B)  
2:(Op=='-')  
3:calc=minus(A,B)

Файл sum.txt

start -> 0  
0 -> end  
  
0:sum=A+B

Файл minus.txt

start -> 0  
0 -> end  
  
0:minus=(A-B)

Файл funcsCFG.txt

calc->sum  
calc->minus

**Вывод**

В ходе работы реализовано построение графа потока управления для каждой функции, а также графа вызовов для всей совокупности входных файлов, описывающее отношения между функциями/методами входных файлов в плане обращению их друг к другу по именам.