Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Волгоградский государственный технический университет»

УТВЕРЖДАЮ:  
 заведующий кафедрой ПОАС   
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Орлова Ю.А.  
 «\_\_\_»\_\_\_\_\_\_\_\_\_ 2020 г.

**КАЧЕСТВО И НАДЕЖНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**  
«Отображение сложной структуры»  
Описание программы

СОГЛАСОВАНО: Разработчик:  
руководитель работы: студент ПрИн-266  
доцент кафедры ПОАС \_\_\_\_\_\_\_\_\_\_ Пехтелев Е.П.  
\_\_\_\_\_\_\_\_\_\_ Сычев О.А. «\_\_\_»\_\_\_\_\_\_\_\_\_ 2020 г.  
«\_\_\_»\_\_\_\_\_\_\_\_\_ 2020 г.

Нормоконтроллер:  
 Преподаватель кафедры ПОАС  
 \_\_\_\_\_\_\_\_\_\_ Матюшечкин Д.С.  
 «\_\_\_»\_\_\_\_\_\_\_\_\_ 2020 г.

2020

Содержание

[1 Общие сведения 3](#_Toc44088823)

[2 Функциональное назначение 3](#_Toc44088824)

[3 Описание логической структуры 3](#_Toc44088825)

[4 Используемые технические средства 4](#_Toc44088826)

[5 Вызов и загрузка 4](#_Toc44088827)

[6 Входные и выходные данные 4](#_Toc44088828)

[Приложение А. Описание используемых в программе структур данных 5](#_Toc44088829)

[Приложение Б. Описание алгоритмов функций 7](#_Toc44088831)

[Приложение В. Диаграмма вызовов функций 13](#_Toc44088833)

[Приложение Г. Диаграмма потоков данных 14](#_Toc44088835)

# 1 Общие сведения

Программа называется «StructureView». Программа предназначена для нахождения и отображения части структуры, описываемой выражением. Для корректного функционирования программы необходима операционная система Windows XP и выше. Программа написана на языке C++ с использованием его стандартных библиотек и библиотеки Qt.

# 2 Функциональное назначение

Программа может быть использована для нахождения и отображения части структуры, описываемой выражением. Областью применения программы являются разработки кафедры «Программное обеспечение автоматизированных систем» Волгоградского Государственного Технического Университета.

# 3 Описание логической структуры

Работа программы логически разделена на несколько частей:

* считывание входных данных;
* обработка входных данных;
* расчет параметров для изображения;
* создание изображения;
* сохранение результата в файл.

Алгоритм отображения в изображение структуры и его части, заданной выражением можно разделить на три логических блока. Описание используемых в программе структур, данных приведено в приложении А. алгоритмы основных функций — в приложении Б, диаграмма вызовов функций — в приложении В, диаграмма потоковых данных — в приложении Г.

Первый логический блок обрабатывает и хранит в оперативной памяти данные о структуре.

Второй логический блок обрабатывает обратную польскую запись и хранит названия элементов структуры для выделения.

Третий логический блок создает производит отрисовку структуры в виде текста на полотне и выделяет элементы структуры другим цветом.

4 Используемые технические средства

Описание требуемых технических средств содержится в Техническом задании в пункте 3.3

5 Вызов и загрузка

Программа запускается посредством командной строки с указанием 1 параметра – путь к файлу.

Например: StructureView.exe C:/Users/Egor/Desktop/Struct/Primer.txt

6 Входные и выходные данные

Формат входных и выходных файлов описан в пункте 3.4 технического задания

# Приложение А

Описание используемых в программе структур данных

Класс DataStruct содержит поля:

QString error – сообщение об ошибке

bool ok = true – успешность выполнения

QVector<QString> elemName – список названий элементов структуры

QVector<QString> elemType – список типов элементов структуры

QVector<int> elemValue – список значений элементов структуры

QVector<bool> elemIsStatic –список

QJsonDocument json – документ json, получаемый из входных данных

QString structName – название структуры

QString objectName – название объекта

Класс Expression содержит поля:

enum operation {Add,

Sub,

Mul,

Div,

Inc,

Dec,

Reference,

ReferenceToTheStatic} – операции для обработки выражения

QString error –сообщение об ошибке

bool ok = true – успешность выполнения

QString express – выражение

QVector<QString> stack – стек для обработки

QVector<QString> result – список названий элементов, к которым совершено обращение

DataStruct\* curStruct –ссылка на данные о структуре

# Приложение Б

# Описание алгоритмов функций

Главная функция программы:

int main(int argc, char \*argv[])

Входные данные:

argc - количество переданных аргументов командной строки

argv[] – аргрументы командной строки:

argv[1] - путь к файлу

Выходные данные

Если

Алгоритм функции:

1. Считать данные с файлов

2. Обработать данные о структуре

3. Обработать выражение

4. Создать изображение

5. Отрисовать структуру

6. Сохранить изображение в файл

Функция обработки данных о структуре

void DataStruct::parceStruct()

Алгоритм функции:

1. Получить из json документа объект

2. Если в json объекте присутствует поле "structName"

2.1 Запоминаем значение поля

3. Иначе

3.1 Сообщаем о отсутствии названия структуры

4. Если в json объекте присутствует поле "objectName "

4.1 Запоминаем значение поля

5. Иначе

5.1 Сообщаем о отсутствии названия объекта

6. Если json объект имеет внутри себя объект с названием "properties"

6.1 Получаем объект с названием "properties"

6.2 По количеству записей в объекте

6.2.1 Запомнить тип переменной

6.2.2 Запомнить имя переменной

6.2.3 Запомнить значение переменной

6.2.4 Запомнить является ли переменная статической

7. Если json объект имеет внутри себя объект с названием "methods"

7.1 Получаем объект с названием "methods"

7.2 По количеству записей в объекте

7.2.1 Запомнить тип возвращаемого значения метода

7.2.2 Запомнить имя метода

7.2.3 Запомнить является ли метод статическим

Функция обработки выражения

bool Expression::parceExp(DataStruct\* curDataStruct)

Входные данные:

ссылка на данные о структуре

Выходные данные:

true если выполнено успешно, иначе false

Алгоритм функции:

1. Разделить выражение на операнды

2. Пока операнды не закончились и нет ошибок

2.1. Берем операнд

2.2. Если операнд сложение

2.2.1. Если есть два числа в стеке

2.2.1.1. Если оба операнда являются числами

2.2.1.1.1. Складываем

2.2.1.1.2. Возвращаем в стек

2.2.1.2. Иначе

2.2.1.2.1 Сообщаем что встречены неизвестные переменные

2.2.2. Иначе

2.2.2.1. Сообщаем, что недостаточно операндов

2.3. Если операнд вычитание

2.3.1. Если есть два числа в стеке

2.3.1.1. Если оба операнда являются числами

2.3.1.1.1. Вычитаем

2.3.1.1.2. Возвращаем в стек

2.3.1.2. Иначе

2.3.1.2.1 Сообщаем что встречены неизвестные переменные

2.3.2. Иначе

2.3.2.1. Сообщаем, что недостаточно операндов

2.4. Если операнд произведение

2.4.1. Если есть два числа в стеке

2.4.1.1. Если оба операнда являются числами

2.4.1.1.1. Умножаем

2.4.1.1.2. Возвращаем в стек

2.4.1.2. Иначе

2.4.1.2.1 Сообщаем что встречены неизвестные переменные

2.4.2. Иначе

2.4.2.1. Сообщаем, что недостаточно операндов

2.5. Если операнд деление

2.5.1. Если есть два числа в стеке

2.5.1.1. Если оба операнда являются числами

2.5.1.1.1. Делим

2.5.1.1.2. Возвращаем в стек

2.5.1.2. Иначе

2.5.1.2.1 Сообщаем что встречены неизвестные переменные

2.5.2. Иначе

2.5.2.1. Сообщаем, что недостаточно операндов

2.6. Если операнд инкремент

2.6.1. Если есть чисто в стеке

2.6.1.1. Если операнд является числом

2.6.1.1.1. Прибавляем 1

2.6.1.1.2. Возвращаем в стек

2.6.1.2. Иначе

2.6.1.2.1 Сообщаем что встречена неизвестная переменная

2.6.2. Иначе

2.6.2.1. Сообщаем, что недостаточно операндов

2.7. Если операнд декремент

2.7.1. Если есть чисто в стеке

2.7.1.1. Если операнд является числом

2.7.1.1.1. Вычитаем 1

2.7.1.1.2. Возвращаем в стек

2.7.1.2. Иначе

2.7.1.2.1 Сообщаем что встречена неизвестная переменная

2.7.2. Иначе

2.7.2.1. Сообщаем, что недостаточно операндов

2.8. Если операнд динамическое обращение

2.8.1. Если есть два значения в стеке

2.8.1.1. Если значение первого операнда не совпадает с названием объекта

2.8.1.1.1. Сообщаем о неверном имени объекта

2.8.1.2. Если значения второго аргумента нет среди элементов структуры

2.8.1.2.1. Сообщаем что аргумент не является элементом структуры

2.8.1.3. Если элемент структуры статический

2.8.1.3.1. Сообщаем о неправильном обращении к элементу структуры

2.8.1.4. Иначе

2.8.1.4.1. Если элемента нет в списке с результатом

2.8.1.4.1.1. Добавляем в список

2.8.1.4.2. Возвращаем в стек число 1

2.8.2. Иначе

2.8.2.1. Сообщаем, что недостаточно операндов

2.9. Если операнд статическое обращение

2.9.1. Если есть два значения в стеке

2.9.1.1. Если значение первого операнда не совпадает с названием структуры

2.9.1.1.1. Сообщаем о неверном имени структуры

2.9.1.2. Если значения второго аргумента нет среди элементов структуры

2.9.1.2.1. Сообщаем что аргумент не является элементом структуры

2.9.1.3. Если элемент структуры динамический

2.9.1.3.1. Сообщаем о неправильном обращении к элементу структуры

2.9.1.4. Иначе

2.9.1.4.1. Если элемента нет в списке с результатом

2.9.1.4.1.1. Добавляем в список

2.9.1.4.2. Возвращаем в стек число 1

2.9.2. Иначе

2.9.2.1. Сообщаем, что недостаточно операндов

2.10. Если операнд число

2.10.1 Добавляем в стек

3. Возвращаем признак успешного выполнения

# Приложение В

# Диаграмма вызовов функций

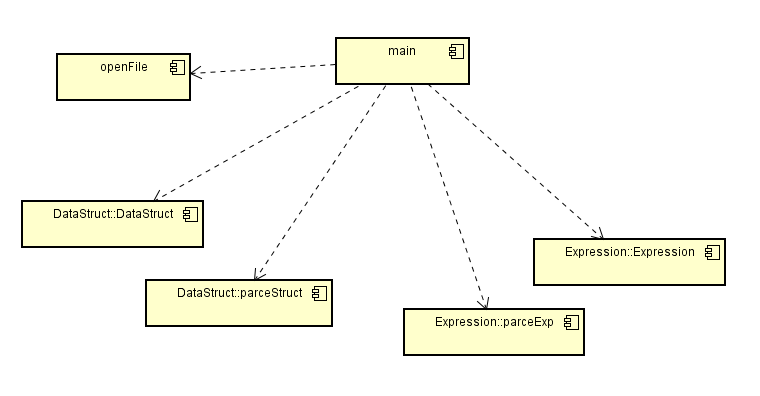


Рисунок 1. Диаграмма вызовов функций

# Приложение Г

# Диаграмма потоков данных

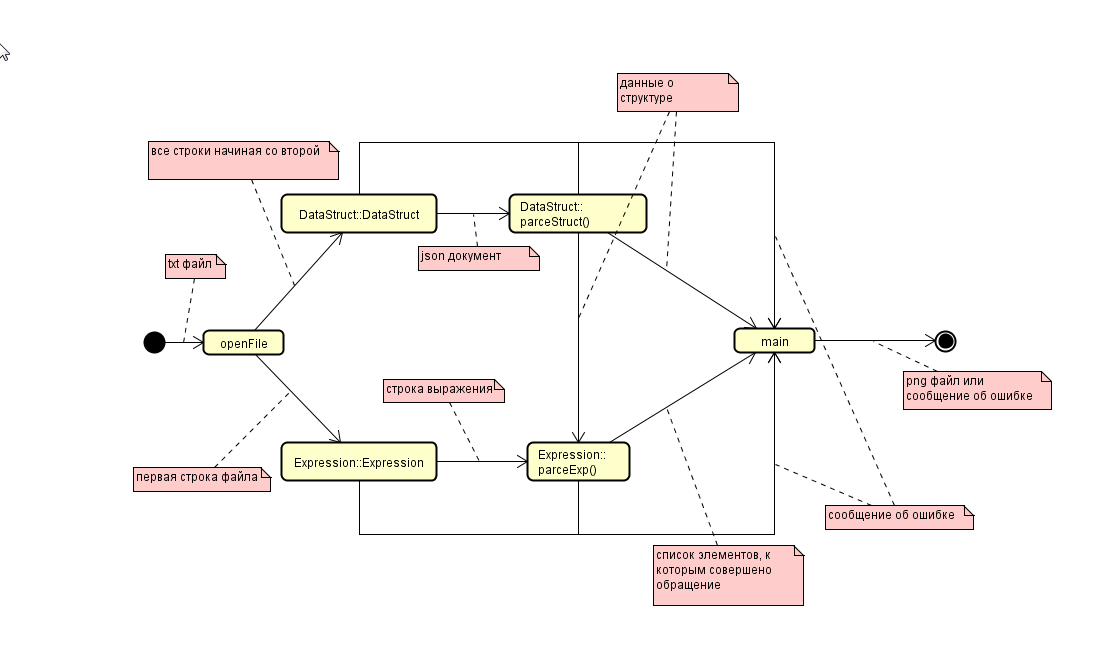


Рисунок 2. Диаграмма потоков данных