Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

УТВЕРЖДАЮ:

Зав. кафедрой ПОАС

\_\_\_\_\_\_\_\_\_\_ Орлова Ю.А.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Качество и надёжность программного обеспечения

«Генерация объяснения выражения»

Техническое задание

СОГЛАСОВАНО: Разработчик:

Руководитель работы: Студент ПрИн-266

Доцент кафедры ПОАС \_\_\_\_\_\_\_\_\_\_ Чибриков А.А.

\_\_\_\_\_\_\_\_\_\_ Сычев О.А. «16»\_мая\_\_\_ 2020 г.

«\_\_\_»\_\_\_\_\_\_\_ 2020 г.

Нормоконтролер:

Преподаватель кафедры ПОАС

\_\_\_\_\_\_\_\_\_\_ Матюшечкин Д.С.

«\_\_\_»\_\_\_\_\_\_\_ 2020 г.

2020

Содержание

|  |  |  |  |
| --- | --- | --- | --- |
| Введение..................................................................................................................... | | | 3 |
| 1 | Основание для разработки .................................................................................... | | 3 |
| 2 | Назначение для разработки................................................................................... | | 3 |
| 3 | Требования к программе ....................................................................................... | | 3 |
| 3.1 | | Требования к функциональным характеристикам .......................................... | 3 |
| 3.2 | | Требования к надежности .................................................................................. | 5 |
| 3.3 | | Требования к составу и параметрам технических средств............................. | 5 |
| 3.4 | | Требования к информационной и программной совместимости................... | 6 |
| 4 | Требования к программной документации ......................................................... | | 6 |
| 5 | Стадии и этапы разработки .................................................................................... | | 7 |
| 6 | Порядок контроля и приемки................................................................................. | | 7 |
| Приложение 1. Коды возможных ошибок, с расшифровкой................................. | | | 8 |
| Приложение 2. Пример файла с деревом разбора выражения (JSON)................ | | | 9 |
| Приложение 3. Пример файла с деревом разбора выражения (XML)................ | | | 10 |
| Приложение 4. Пример файла с описанием элементов......................................... | | | 11 |
| Приложение 5. Пример файла с описанием функций........................................... | | | 12 |
| Приложение 6. Пример работы программы........................................................... | | | 13 |

Введение

Техническое задание описывает требования к программе генерации объяснения выражения (в дальнейшем – Expression Describer).

1 Основание для разработки

Программа разрабатывается на основании задания на курсовой проект по дисциплине «Надежность и качество программного обеспечения», выданного доцентом кафедры ПОАС Сычевым О.А. 11 февраля 2020 года.

2 Назначение для разработки

Программа предназначена для создания описания заданного выражения.

3 Требования к программе

3.1 Требования к функциональным характеристикам

Программа должна составлять описание выражения на русском языке. Программа должна принимать на вход дерево разбора выражения и строки, описывающие, что обозначают переменные и пользовательские типы, а также описания функций. Программа должна склонять по падежам описания базовых операций и приближенно склонять введенные описания элементов, опираясь на производимые над ними операции пример работы в приложении 6.

Программа должна поддерживать следующие бинарные операции:

* сложение (“+”);
* вычитание (“-”);
* умножение (“\*”);
* деление (“/”);
* остаток от деления (“%”);
* обращение к элементу массива (“[]”);
* побитовый сдвиг влево (“<<”);
* побитовый сдвиг вправо (“>>”);
* побитовое И (“&”);
* побитовое ИЛИ (“|”);
* побитовое исключающее ИЛИ (“^”);
* равенство (“==”);
* неравенство (“!=”);
* больше (“>”);
* меньше (“<”);
* больше или равно (“>=”);
* меньше или равно (“<=”);
* логическое И (“&&”);
* логическое ИЛИ (“||”);
* присвоение (“=”);
* присвоение с умножением (“\*=”);
* присвоение с делением (“/=”);
* присвоение со сложением (“+=”);
* присвоение с вычитанием (“-=”).

Программа должна поддерживать следующие унарные операторы:

* инкремент (“++”);
* декремент (“--”);
* логическое отрицание (“!”);
* побитовое отрицание (“~”);
* непрямое обращение (“\*”);
* взятие адреса (“&”);
* непрямое обращение к члену структуры (“->”);
* прямое обращение к члену структуры (“.”).

При отсутствии описания для какого-либо элемента выражения, программа должна выводить имя этого элемента, вместо его описания.

3.2 Требования к надежности

Программа должна обрабатывать следующий перечень ошибок:

* некорректное дерево разбора выражения;
* ошибки чтения файла с деревом разбора выражения;
* ошибки чтения файлов с описанием элементов.

При ошибке, программа должна немедленно прекратить работу и вывести сообщение с кодом ошибки. Подробное описание кодов возможных ошибок приведены в Приложении 1.

3.3 Требования к составу и параметрам технических средств

Минимальные требования к оборудованию:

* устройство ввода (клавиатура и мышь);
* не менее 10 МБ свободной памяти на жестком диске;
* не менее 200 МБ оперативной памяти.

3.4 Требования к информационной и программной совместимости

Для передачи дерева разбора выражения, описания элементов используются файлы в текстовом формате. Дерево разбора выражения хранится в JSON или XML представлении. Допустимые расширения файлов – xml, json. Для XML: каждая вершина дерева – тэг с именем node и атрибутом name, обозначающим элемент или операцию. Для JSON: каждая вершина дерева – объект со свойствами name, обозначающей элемент или операцию, и child, содержащей дочерние вершины. Описания элементов вводятся в именительном падеже. На одной строке указывается обозначение элемента, на следующей - его описание. В качестве описания функции – описание результата этой функции и описание ее аргументов. Количество аргументов записывается через пробел после имени функции на первой строке, на следующих строках указываются описание результата функции и названия аргументов, если описание для аргумента отсутствует, указывается прочерк.

Конечный результат выводится в консоль.

Входные файлы задаются в программу последовательно в виде аргументов командной строки: файл с деревом разбора выражения, файл с описанием элементов, файл с описанием функций. Примеры входных файлов смотри в приложении 2, 3, 4, 5.

Программа будет написана на языке C++ с использованием его стандартных библиотек, библиотеки Qt и библиотеки для склонения по падежам Morpher.

4 Требования к программной документации

Программная документация должна быть представлена в бумажной форме в виде:

* техническое задание;
* технический проект в виде описания программы;
* программа и методика испытаний;
* руководство программиста.

Программная документация должна быть представлена в электронной форме в виде:

* техническое задание;
* рабочая документация;
* текст программы.

5 Стадии и этапы разработки

Этапы разработки подразделяются на:

1. техническое задание – 14 марта 2020 года;
2. технический проект – 4 апреля 2020 года;
3. рабочий проект – 16 мая 2020 года;
4. внедрение – 30 мая 2020 года.

6 Порядок контроля и приемки

Программа сдается для проверки преподавателю не позднее чем за 3 недели до конца семестра (5 мая 2020 года). При обнаружении в программе ошибок и недостатков исполнитель устраняет их в недельный срок и предоставляет программу на повторную проверку.

Приложение 1

Коды возможных ошибок, с расшифровкой

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Описание ошибки | Пример, приводящий к ошибке | Пример сообщения об  ошибке |
| 1 | Недопустимый формат файла с деревом | /tree.txt | Недопустимый формат файла с деревом |
| 2 | Ошибка открытия файла с деревом | /tree1.json (Отсутствует) | Ошибка открытия файла с деревом: Не удается найти указанный файл. |
| 3 | Ошибка разбора файла с деревом | <node name="z" 12></node> | Ошибка разбора файла с деревом: error occurred while parsing element on 4:20 |
| "name":"+",12 | Ошибка разбора файла с деревом: unterminated object on 19 |
| 4 | Ошибка открытия файла с описанием | /desc1.txt (Отсутствует) | Ошибка открытия файла с описанием: Не удается найти указанный файл. |
| 5 | Отсутствует атрибут name у элемента | <node > | Ошибка разбора файла с деревом: Отсутствует аттрибут name |
| {“child”:[]} |

Приложение 2

Пример файла с деревом разбора выражения (JSON)

{

"name":"+",

"child":[

{

"name":"\*",

"child":[{"content":"z"},{"content":"y"}]

},{

"name":"\*",

"child":[{

"name":"[]",

"child":[{"name":"array"},{"name":"x"}]

}

,{"name":"x"}]

}]}

Приложение 3

Пример файла с деревом разбора выражения (XML)

<?xml version="1.0" encoding="UTF-8"?>

<node name="+">

<node name="\*">

<node name="z"></node>

<node name="y"></node>

</node>

<node name="\*">

<node name="[]">

<node name="array"></node>

<node name="x"></node>

</node>

<node name="x"></node>

</node>

</node>

Приложение 4

Пример файла с описанием элементов

x

абсцисса

y

ордината

z

аппликата

Приложение 5

Пример файла с описанием функций

log 2

логарифм

основание

-

Приложение 6

Примеры работы программы

|  |  |  |  |
| --- | --- | --- | --- |
| Выражение | Описание элементов | Описание функций | Результат |
| x + y | x  абсцисса  y  ордината |  | Сумма абсциссы и ординаты |
| x\*sin(y) | x  абсцисса  y  ордината | sin 1  синус  - | произведение абсциссы на синус от ординаты |
| a = log(a,b) |  | log 2  логарифм  основание  - | Присвоить ‘a’ значение логарифма от ‘a’, в качестве основания и ‘b’ |