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.

«» 2019 г.

Качество и надёжность программного обеспечения

"Поиск ошибок при составлении трассы выполнения алгоритма"

Техническое задание

|  |  |
| --- | --- |
| СОГЛАСОВАНО:  Руководитель работы  доцент кафедры ПОАС  Сычёв О.А.  «» 2019 г. | РАЗРАБОТЧИК:  Студент группы ПрИн-266  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Носкин В.В.  «» 2019 г. |
|  | НОРМОКОНТРОЛЁР:  доцент кафедры ПОАС  Сычёв О.А.  «» 2019 г. |

2019 г.

Содержание

[Введение 3](#_Toc4431358)

[1 Основание для разработки 3](#_Toc4431359)

[2 Назначение разработки 3](#_Toc4431360)

[3 Требования к программе 3](#_Toc4431361)

[3.1 Требования к функциональным характеристикам 3](#_Toc4431362)

[3.2 Требования к надёжности 4](#_Toc4431363)

[3.3 Требования к составу и параметрам технических средств 4](#_Toc4431364)

[3.4 Требования к информационной и программной совместимости 5](#_Toc4431365)

[4 Требования к программной документации 6](#_Toc4431366)

[5 Стадии и этапы разработки 7](#_Toc4431367)

[6 Порядок контроля и приемки 7](#_Toc4431368)

[Приложение 1. Перечень правил, ошибок и сообщений об ошибках 8](#_Toc4431369)

[Приложение 2. Алгоритмические структуры и возможные ошибки файлов 16](#_Toc4431370)

[Приложение 3. Примеры файлов с входными данными 20](#_Toc4431371)

[Приложение 4. Примеры файлов с выходными данными 21](#_Toc4431372)

# Введение

Наименование программы –«Программа поиска ошибок при составлении трассы выполнения алгоритма» (далее «Программа»).

Программа предназначена для организации тренажеров по повышению понимания студентами младших курсов основ программирования.

# 1 Основание для разработки

Разработка проводится на основании задания на курсовой проект по дисциплине «Качество и надёжность программного обеспечения», выданного доцентом кафедры ПОАС Сычёвым О. А. 13 февраля 2019 г.

# 2 Назначение разработки

Разработка может быть использована для организации тренажеров по повышению понимания студентами младших курсов основ программирования.

# 3 Требования к программе

# 3.1 Требования к функциональным характеристикам

Программа должна выполнять следующие функции:

1. определять все ошибки при составлении трассы в соответствии с перечнем правил, находящимся в приложении 1;
2. указывать место нарушения правила;
3. указывать какое правило нарушено;
4. объяснять ошибку на естественном (русском) языке в соответствии с таблицей ошибок вприложении 1;
5. указывать, что трасса корректна, если ошибки не были найдены;
6. генерировать код на языке Си из дерева программы.

Под алгоритмом будем понимать дерево алгоритмических структур, таких как: следование, развилка, выбор, цикл до, цикл пока, цикл для. В качестве трассы будем понимать последовательностьактов выполнения, не имеющих вложенность, начал и концов актов алгоритмических структур,имеющих вложенность. Под актом будем понимать факт выполнения алгоритмической структуры, набора инструкций или оператора.При обнаружении ошибки, программа должна указывать на ошибочный акт, если он находится в алгоритмической структуре, не имеющей вложенность, в противном случае программа указывает на начало и конец ошибочного акта.

# 3.2 Требования к надёжности

В случае возникновения ситуаций ввода некорректных данных программа должна выдать соответствующее ошибке сообщение и корректно завершиться.

Ситуации некорректных входных данных и выдаваемые сообщения об ошибках указаны в приложении 2.

# 3.3 Требования к составу и параметрам технических средств

В состав технических средств должен входить персональный компьютер, включающий в себя:

1. процессорIntelPentiumB980 с тактовой частотой 2.4 ГГц, не менее;
2. оперативную память объемом 4Гб, не менее;
3. операционная система Windows7 или выше.

# 3.4 Требования к информационной и программной совместимости

При запуске программы на вход подается два параметра командной строки:

* имя файла с деревом алгоритма программы;
* имя файла с трассой выполнения программы.

Для передачи дерева алгоритма программы используется обратная польская запись. Дерево в таком формате записывается в текстовый файл в его первой строке.

Примеры входных файлов приведены в приложении 3.

Требования к записи алгоритма программы:

* используются только разрешённые алгоритмические структуры, описанныевприложении 2;
* разделение алгоритмических структур и их элементов между собой осуществляется с помощью пробела;
* имя оператора указывается после символа «#»;
* количество элементов алгоритмической структуры указывается в скобках;
* алгоритмическая структура обязательно должна содержать имя и кол-во элементов (количество элементов не указывается только для«утверждение»);
* имя алгоритмической структуры должно быть уникальным для алгоритма.

Требования к записи трассы выполнения:

* используются только разрешённые алгоритмические структуры, указанные вприложении 2;
* начало цикла, альтернатива, выбора или следования помечается словом «begin»;
* начало цикла, альтернатива, выбора или следования помечается словом «end»;
* цикл, альтернатива, выбор или следование должны иметь как начало, так и конец;
* имя акта указывается после символа «#»;
* номер исполнения акта указывается после символа «@»;
* состояние условия указывается в скобках;
* состояние условия может принимать значение 1 (истинна)или 0 (ложь);
* каждый акт начинается с новой строки;
* трасса не должна содержать актов алгоритмических структур, которые не описаны в алгоритме;
* если алгоритмическая структура имеет nактов, то трасса должна содержать все акты с неповторяющимися номерами исполнения от 1 до n.

На выходе программа предоставляет два файла текстового формата:

* файл с расширением «.txt», содержащий ошибки трассы, если они имеются.
* файлс расширением «.cpp», содержащий код алгоритма на языке Си.

Примеры выходных файлов приведены в приложении 4.

Имя выходного файла с ошибками генерируется из префикса «output\_»и названия файла с трассой программы.

Имя выходного файла с алгоритмом на языке си генерируется из префикса «code\_»и названия файла с алгоритмом программы.

Программа будет написана на языке C++ с использованием стандартных библиотек и библиотеки Qt.

# 4 Требования к программной документации

Программная документация должна быть представлена в бумажной в 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:

- копии всех документов бумажной формы;

- рабочая документация;

- текст программы.

# 5 Стадии и этапы разработки

Разработка должна быть проведена в четыре этапа:

1) техническое задание – 15 марта 2019 года;

2) технический проект– 4 апреля 2019 года;

3) рабочий проект – 10 мая 2019 года;

4) внедрение–24мая 2019 года.

# 6 Порядок контроля и приемки

Программа сдается для проверки преподавателю 24 мая 2019 года. При обнаружении в программе ошибок и недостатков исполнитель устраняет их в недельный срок и предоставляет программу на повторную проверку.

# Приложение 1

Перечень правил, ошибок и сообщений об ошибках

Таблица 1. Перечень правил, ошибок и сообщений об ошибках

|  |  |  |  |
| --- | --- | --- | --- |
| № | Утверждение | Ошибки | Сообщения об ошибках |
| 1 | Если следование А СОДЕРЖИТ оператор Б, то для любого акта А должен существовать акт Б, начало акта А < акт Б < конец акта А | Оператор Б входит в следование А, при этом акт Б < начало акта А | <Акт Б> не может выполняться раньше <начало акта А> потому что <Б> входит в <следование А>. |
| Оператор Б входит в следование А, при этом акт Б > конец акта А | <Акт Б> не может выполняться позже <конец акта А>, потому что <Б> входит в <следование А> |
| Оператор Б входит в следование А, при этом между началом акта А и концом акта А содержится больше одного акта Б | <акт А> не может содержать больше одного <акт Б>, потому что следование выполняет операции внутри по одному разу. |
| Оператор Б входит в следование А, при этом между началом акта А и концом акта А не содержится ни одного акта Б | Между <начло акта А> и <конец акта А> должен выполниться <оператор Б>, потому что <оператор Б> входит в <следование А> |
| 2 | Если существует акт выполнения А, то должно выполняться условие, что начало акта А <= конец акта А | Акт А существует и начало акта А > конец акта А | <начало акта А> не может выполняться позже <конец акта А>. |

Продолжение таблицы 1.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Утверждение | Ошибки | Сообщения об ошибках |
| 3 | Если набор инструкций А содержит наборы инструкций А1..АN, то каждый акт А может содержать акты для части из наборов A1..AN и только их, причём должно выполняться условие, что начало акта А < любой акт из набора А1..AN < конец акта А> | Набор инструкций А содержит наборы инструкций А1..АN, но акт Аi из набора А1..АN < начало акта А | <Акт Аi> не может находится раньше <начало акта А>, потому что <Акт Аi> входит в <набор А> |
| Набор инструкций А содержит наборы инструкций А1..АN, но акт Аi из набора А1..АN > конец акта А | <Акт Аi> не может находится дальше чем <конец акта А>, потому что <Акт Аi> входит в <набор А> |
| 4 | Если А и Б операторы входящие в следование В, и A < Б, то для любого акта В должны существовать акты А и Б, при этом конец Акта А < начало акта Б | А и Б операторы входящие в следование В, и А < Б, и для любого акта В существуют акты А и Б, но конец акта А > начало акта Б | <конец акта А> не может находится позже <начало акта Б>, потому что в <следование В><оператор А> находится перед <оператор Б> |
| 5 | Если А это альтернатива, Б её условие, В -первая ветка, а Г - вторая, то для любого акта А, если Б истинно, то существует акт В и не существует акт Г, если же Б ложно, то существует акт Г и не существует акт В | А является альтернативой, Б её условие, В - первая ветка, а Г - вторая, но для акта А, при истинном Б не существует акта В | Во время выполнения альтернативы <акт А> должна выполниться ветка <В>, потому что условие <Б> истинно |

Продолжение таблицы 1.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Утверждение | Ошибки | Сообщения об ошибках |
|  |  | А является альтернативой, Б её условие, В - первая ветка, а Г - вторая, но для акта А, при истинном Б существует акт Г | Во время выполнения альтернативы <акт А> не должна выполниться ветка <Г>, потому что условие <Б> истинно |
| А является альтернативой, Б её условие, В - первая ветка, а Г - вторая, но для акта А, при ложном Б не существует акта Г | Во время выполнения альтернативы <акт А> должна выполниться ветка <Г>, потому что условие <Б> ложно |
| А является альтернативой, Б её условие, В - первая ветка, а Г - вторая, но для акта А, при ложном Б существует акт В | Во время выполнения альтернативы <акт А> не должна выполниться ветка <В>, потому что условие <Б> ложно |
| 6 | Если А это цикл ПОКА, Б его условие, а В его тело, то для каждого акта А существуют акты В (0..n) расположенные подряд (т.е. Не существует актов Г, что конец акта В(i) < Г < начало акта В(i+1)) | А является циклом пока, Б его условием, В его телом, но для акта А существует такой акт Г, что конец акта В(i) < Г < начало акта В(i+1) | Во время выполнения цикла <акт А><акт Г> не можетвыполняться между <акт Вi> и <акт Вi+1> , потому что тело цикла на каждой итерации должно выполниться сразу друг за другом. |
| 7 | Если А это цикл, Б его условие, а В его тело, то к началу любого акта В (если это не первая итерация цикла ДО) условие Б будет истинным. | А является циклом, Б его условием, а В телом, но к началу акта В условие Б является ложным | Во время выполнения цикла <акт А> на <i-ой> итерации не должен выполниться <акт В>, потому что условие <Бi> ложно |

Продолжение таблицы 1.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Утверждение | Ошибки | Сообщения об ошибках |
| 8 | Если А это цикл ДО или ПОКА, Б его условие, а В его тело, то для каждого акта А после выполнения i-го акта В, если условие Б истинно, существует акт В i+1 | А это цикл ДО или ПОКА, Б его условие, а В его тело, то для акта А после выполнения i-го акта В при истинном условии Бi, не существует акт Вi+1 | Во время выполнения цикла <акт А> на <i-ой> итерации должно выполниться тело <В>, потому что условие <Бi> истинно |
| А это цикл ДО или ПОКА, Б его условие, а В его тело, то для акта А после выполнения i-го акта В при ложном условии Бi, существует акт Вi+1 | Во время выполнения цикла <акт А> на очередной итерации не должно выполниться тело <В>, потому что условие <Б> ложно |
| 9 | Если А это цикл ДО, Б его условие, а В его тело, то для каждого акта А существуют акты В (1..n) расположенные подряд (т.е. Не существует актов Г, что конец акта В(i) < Г < начало акта В(i+1)) | А это цикл ДО, Б его условие, В его тело, но для акта А не существует ни одного акта В | Во время выполнения цикла <акта А> типа ДО его тело <В> должно выполниться хотя бы один раз |
| А является циклом ДО, Б его условием, В его телом, но для акта А существует такой акт Г, что конец акта В(i) < Г < начало акта В(i+1) | Во время выполнения цикла <акт А><акт Г> не должно выполняться между <акт Вi> и <акт Вi+1> , потому что тело цикла на каждой итерации должно выполниться сразу за выполнением на предыдущей итерации |

Продолжение таблицы 1.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Утверждение | Ошибки | Сообщения об ошибках |
| 10 | Если А это цикл ДЛЯ, а Б его инициализация, то для любого акта А первым (наименьшим) будет акт Б (т.е. не существует актов Г, что начало акта А < акт Г < начало акта Б); это будет единственный акт Б для этого акта А | А это цикл ДЛЯ, Б его инициализация, но для акта А существует такой акт Г, что начало акта А < акт Г < начало акта Б | Во время выполнения цикла <А><акт Г> не должен выполняться между <начало акт А> и <акт Б>, потому что для цикла ДЛЯ сначала должна выполниться его инициализация |
| А это цикл ДЛЯ, Б его инициализация, но для акта А существует несколько актов Б | Во время выполнения цикла <А> типа ДЛЯ не должно выполниться <акт Бi>, потому что инициализация цикла ДЛЯ выполняется 1 раз |
| 11 | Если А это цикл ДЛЯ, Б его тело, а В - переход для каждого акта А будут существовать акты Б и В (0..n) такие, что для любого целого числа i Б(i) < В(i) < Б(i+1) < В(i+1) и т.д., причём пара актов Б и В расположены подряд (т.е. если существуют акты Б(i+1) и В(i + 1) при актах Б(i) и В(i), то не существует актов Г, что конец акта В(i) < акт Г < начало акта Б(i+1) | А это цикл ДЛЯ, Б его тело, В - переход для каждого акта А, но для акта А для некоторого целого числа i акт В(i) < акт Б(i) | Во время выполнения цикла <акт А> на каждой итерации его переход<В> должно выполняться после тела<Б> |
| А это цикл ДЛЯ, Б его тело, В - переход для каждого акта А, но для акта А существует такой акт Г, что конец акта В(i) < акт Г < начало акта Б(i+1) | Во время выполнения цикла <акт А><акт Г> не должен выполняться между <конец акта Вi> и <начало акта Бi+1> , потому что тело следующей итерации должно выполниться сразу за переходом текущей итерации |

Продолжение таблицы 1.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Утверждение | Ошибки | Сообщения об ошибках |
|  |  | А это цикл ДЛЯ, Б его тело, В - переход для каждого акта А, но для акта А существует такой акт Г, что конец акта Б(i) < акт Г < начало акта В(i) | Во время выполнения цикла <акт А><акт Г> не должен выполняться между <конец акта Бi> и <начало акта Вi> , потому что переход должен выполниться сразу за телом цикла |
| 12 | Если А это цикл ДЛЯ, Б его условие, В его тело, а Г - переход для каждого акта А после выполнения i-го акта Г, если условие Б истинно, существует акт Вi+1 и акт Гi+1 | А это цикл ДЛЯ, Б его условие, В его тело, а Г переход, но для акта А после выполнения i-го акта Г, при истинном условии Б не существует акта Вi+1 | Во время выполнения<акт А> должен выполниться<акт Вi+1> после выполнения <акт Гi>, потому что условие <Б> истинно |
| А это цикл ДЛЯ, Б его условие, В его тело, а Г переход, но для акта А после выполнения i-го акта Г, при истинном условии Б не существует акта Гi+1 | Во время выполнения<акт А> должен выполниться<акт Гi+1> после выполнения <акт Гi>, потому что условие <Б> истинно |
| А это цикл ДЛЯ, Б его условие, В его тело, а Г переход, но для акта А после выполнения i-го акта Г, при ложном условии Б существует акт Вi+1 | Во время выполнения<акт А> не может выполняться<акт Вi+1> после выполнения <акт Гi>, потому что условие <Б> ложно |

Продолжение таблицы 1.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Утверждение | Ошибки | Сообщения об ошибках |
|  |  | А это цикл ДЛЯ, Б его условие, В его тело, а Г переход, но для акта А после выполнения i-го акта Г, при ложном условии Б существует акт Гi+1 | Во время выполнения<акт А> не может выполняться<акт Гi+1> после выполнения <акт Гi>, потому что условие <Б> ложно |
| 13 | Если А это цикл ДЛЯ, Б его условие, В его тело, а Г переход, то после каждого акта Вi должен существовать акт Гi | А это цикл ДЛЯ, Б его условие, В его тело, а Г переход, но после акта Вi не существует акта Гi | Во время выполнения<акт А> после выполнения <акт Вi> должен выполниться<акт Гi>, потому что в цикле для переход выполняется после тела цикла. |
| 14 | Кратность выполнения тела для программы, функции и следования - 1 | А это алгоритмическая структура, Б первая содержащая её программа, функция или следование, но в акте Б существует несколько актов А | Во время выполнения <акт А>не может выполняться <акт Бi, i>1>,потому что тело<вид А>выполняется ровно один раз |
| 15 | Кратность выполнения веток для альтернативы - 0..1 | А это альтернатива, Б его ветка, но за один акт А выполняется более одного акта Б | ч |
| 16 | Для любого акта А не существует ни одного такого акта Б, что акт Б не <(вложен) акта А и начало акта А <акт Б <конец акта А | Для акта А существует акт Б, что акт Б не<(вложен) акта А и начало акта А <акт Б < конец акта А | Во время выполнения<акт А> не может выполняться<акт Б>, потому что <Б> не является вложенным в <А> |

Продолжение таблицы 1.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Утверждение | Ошибки | Сообщения об ошибках |
| 17 | Если А это выбор, Б набор условий, В набор веток, Г альтернативная ветка, то для любого акта А, если Бi истинно, то существует один и только один акт Вi, относящийся к условию Бi и не существует актов Вj при j != i | А это выбор, Б набор условий, В набор веток, то для акта А, при истинном Бi существует несколько актов Вi | Для выбора <акт А> ветка <акт Вi> при истинном условии <Бi> должна выполниться один раз |
| А это выбор, Б набор условий, В набор веток, то для акта А, при истинном Бi и ложных Б0..Бi-1 не существует ни одного акта Вi | При выполнении выбора <акт А> должен выполниться <акт Вi> потому что <Бi> - первое истинное условие из набора веток |
| А это выбор, Б набор условий, В набор веток, то для акта А, при ложном Бi и ложных Б0..Бi-1 существует акт Вi | При выполнении выбора <акт А> не должен существовать <акт Вi>, потому что условие <Бi> является ложным |
| А это выбор, Б набор условий, В набор веток, то для акта А, при истинном Бi и истинном Бj, где j > i существует <акт Вj> | При выполнении <акт А> при истинных условиях <Бi> и <Бj> не может существовать <акт Вi>, потому что истинное условие <Бi>расположено раньше чем <Бj> |

# Приложение 2

Алгоритмические структуры и возможные ошибки файлов.

Таблица 2. Алгоритмические структуры и их элементы.

|  |  |  |  |
| --- | --- | --- | --- |
| Алгоритмические структуры и их элементы | Обозначение | Примеры записи | Си |
| Следование | seq | stmt#s1 stmt#s2 st#s3 seq#s(3) | { //s  s1;  s2;  s3;  } |
| Альтернатива | alt | cond#c1 stmt#s2 stmt#s3 alt#if(3) | if(c1)  s2;  else  s3; |
| Цикл ДЛЯ | for | stmt#s1 cond#c2 stmt#s3 stmt#s4 for#f(4) | for(s1;c2;s3)  s4; |
| Цикл ПОКА | while | cond#c1 stmt#s2 while#w(2) | while(c1)  s2; |
| Цикл ДО | do | cond#c1stmt#s2 do#d(2) | do{  s2;  }while(c1); |

Продолжение таблицы 2.

|  |  |  |  |
| --- | --- | --- | --- |
| Алгоритмические структуры и их элементы | Обозначение | Примеры записи | Си |
| Выбор | sel | stmt#stmt1 cond#cond1 stmt#stmt2 cond#cond2 stmt#stmt3 se1#se(5) | if(cond1)  stmt1;  else if (cond2)  stmt2;  else  stmt3; |
| Утверждение | stmt | stmt#name | name; |
| Условие | cond | cond#name | name |

Таблица 3. Возможные ошибки файла, содержащего дерево алгоритма.

|  |  |  |
| --- | --- | --- |
| Описание ошибки | Пример, приводящий к ошибке | Пример сообщения об ошибке |
| Файл с деревом алгоритма не найден | Отсутствие файла с алгоритмом | Файл с алгоритмом <имя файла> не найден |
| У оператора отсутствует число его операндов | stmt#1 stmt#2 seq#name | У оператора<имя оператора><положение в строке>отсутствует кол-во операндов. |
| У оператора отсутствует имя оператора | stmt#1 stmt#2 seq#(2) | У оператора <имя оператора><положение в строке>отсутствует его имя. |

Продолжение таблицы 3

|  |  |  |
| --- | --- | --- |
| Описание ошибки | Пример, приводящий к ошибке | Пример сообщения об ошибке |
| У оператора отсутствует разделитель между им и его именем. | stmt#1 stmt#2 seqname(2) | У оператора <имя оператора><положение в строке>отсутствует разделитель между им и его именем. |
| У оператора отсутствует разделитель между его именем и кол-вом операндов. | stmt#1 stmt#2 seq#name2 | У оператора <имя оператора><положение в строке>отсутствует разделитель между его именем и кол-вом операндов. |
| У оператора не хватает операндов | stmt#2 seq#name(2) | У оператора <имя оператора><положение в строке>не хватает операторандов |
| Избыток операндов | stmt#0 stmt#1 stmt#2 seq#name(2) | У оператора <имя оператора><положение в строке>лишние операнды. |
| Используются нераспознанные элементы (элементы, которых нет в таблице 2) | opstmt#1 seq#name(2) | Найден неизвестный элемент <положение в строке> |

Таблица4. Возможные ошибки файла, содержащего трассу выполнения алгоритма.

|  |  |  |
| --- | --- | --- |
| Описание ошибки | Пример, приводящий к ошибке | Пример сообщения об ошибке |
| Файл с трассой не найден | Отсутствие файла с трассой | Файл с трассой<имя файла> не найден |
| Используются нераспознанные элементы (элементы, которых нет в таблице 2) | op stmt#1 seq#name(2) | Найден неизвестный элемент <положение в строке> |

Продолжение таблицы 4.

|  |  |  |
| --- | --- | --- |
| Описание ошибки | Пример, приводящий к ошибке | Пример сообщения об ошибке |
| У акта отсутствует номер его выполнения | stmt#s1@ | В строке <номер строки>отсутствует номер выполнения акта. |
| У акта отсутствует его имя | stmt#@2 | В строке <номер строки>отсутствует имя акта. |
| У акта отсутствует разделитель между актом и его именем | sts1@2 | В строке <номер строки>отсутствует разделитель между актом и его именем. |
| У акта отсутствует разделитель между его именем и номером выполнения | stmt#s12 | В строке <номер строки>отсутствуетразделитель между актом и его именем. |
| У актов вида цикл, следование, альтернатива или выбор отсутствует начало | stmt#s2@1  endseq#main@1 | В строке <номер строки>отсутствуетначало акта вида цикл, следование, альтернатива или выбор |
| У актов вида цикл, следование, альтернатива или выбор отсутствует конец | begin seq#main@1  stmt#s2@1 | В строке <номер строки>отсутствуетконец акта вида цикл, следование, альтернатива или выбор |

# Приложение 3

Примеры файлов с входными данными.

Пример файла с алгоритмом

Пример дерева алгоритма в обратной польской записи см. рис. 1.



Рисунок 1. Пример дерева алгоритма в текстовом редакторе «Notepad++»

Описание примера: представление некоторого алгоритма см. рис. 2, используя обратную польскую запись.

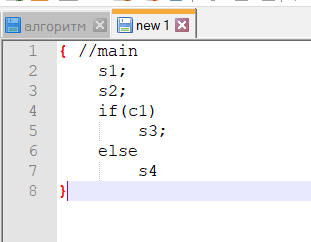


Рисунок 2. Пример алгоритма в текстовом редакторе «Notepad++»

Пример файла с трассой

Примертрассы выполнения алгоритма см. рис. 3.

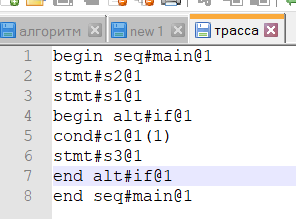


Рисунок 3. Пример файла с трассой выполнения алгоритма в текстовом редакторе «Notepad++»

# Приложение 4

Примеры файлов с выходными данными.

Пример выходного файла с ошибками в текстовом редакторе«Notepad++»на основе входных данных примера, описанного в приложении 3, см. рис. 4.

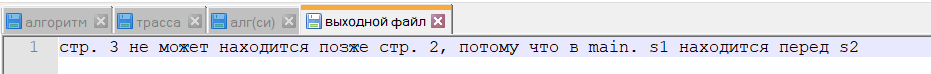


Рисунок 4. Пример выходного файла с ошибками в текстовом редакторе «Notepad++»

Пример выходного файла, содержащего алгоритм на языке СИ в текстовом редакторе «Notepad++» на основе входных данных примера, описанного в приложении 3, см. рис. 5.

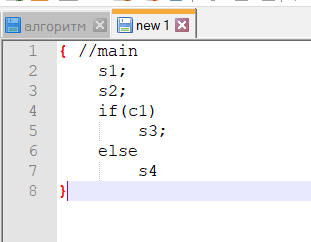


Рисунок 5. Пример выходного файла с алгоритмом на языке СИ в текстовом редакторе «Notepad++»