**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе № 1**

**по дисциплине «Программирование»**

Тема: Обработка строки

|  |  |  |
| --- | --- | --- |
| Студент гр. 2309 |  | Савин П.А. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2023

**Формулировка задания.**

Дана строка-предложение, слова в котором разделяются одним или несколькими пробелами. Найти в этом предложении последнее слово, которое начинается на заданный символ.

**Анализ задания.**

В целом, слова не обязаны быть осмысленными, это может быть просто последовательность символов, ограниченная пробелами. Также стоит уточнить, что:

1) Предложение остается строкой, то есть, находится в пределах одной линии, все остальное не является частью предложения

2) Заданный символ не может по определению быть пробелом

**Математическая** **постановка.**

Дано:

1) Строка — последовательность символов; часть из них является пробелами, остальные могут образовывать непрерывную подпоследовательность, ограниченную пробелами — слова.

2) Ключевой не пробельный символ, по которому будет осуществляться поиск

Найти:

Среди образующихся слов и начинающихся на ключевой символ найти последнее.

Решение:

Поскольку строка — последовательность символов, ее можно представить, как массив. На примере строки «find , me» (рис. 1):



рис. 1 — строка в представлении «массив символов»

В таком представлении сразу видны пробелы, поэтому от этого вида переходим в двумерный массив слов (рис. 2):

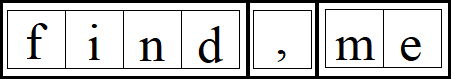


рис. 2 — представление массива слов

где жирными границами обособлены элементы основного массива, тонкими — элементы слова.

Далее остается лишь пройтись по первому символу каждого слова и проверять, равен ли он ключевому. Если проход идет слева направо, то фиксируем по индексу очередное подходящее слово, а последний зафиксированный индекс и будет нужным. В целях оптимизации можно организовать проход справа налево, в таком случае проход будет обрываться на первом найденном слове, однако у такого алгоритма тоже есть худшие случаи, когда ответ находится в первой половине массива или его нет совсем.

Ответ: одно из исходных слов целиком или сообщение, что ни одного подходящего слова не было найдено.

**Контрольный пример.**

За контрольный пример было взято следующее предложение:

«Structures in c are real structures; their only purpose is to compose different types called fields, while structures in c++ are prototypes of class functionality: they chain fields and functions, that process them, into whole composition, also providing base concepts of OOP (inheritance, encapsulation and polymorphism).»;

и заданный символ «c».

Ответ на пример: concepts

**Особенности реализации задания на компьютере.**

Задание разбито на 2 части, обе представляют собой разные реализации одной и той же задачи. Тем не менее, оба варианта предполагают два режима обработки входного файла: по заданному стоп-символу и по заданному количеству символов на чтение.

Для обоих режимов чтения характерны общие проблемы обработки: преждевременное окончание файла, неправильное оформление и пустое предложение (то есть, возможно, строка и считалась верно, но после разбиения слов в ней не оказалось).

При этом, если все предложение оказалось меньше заданного числа символов, ситуация остается корректной, обработка продолжается.

1 часть: модельное представление слова — массив символов, оканчивающийся маркером. Такая реализация является стандартной для C (нуль-терминированная строка), но в рамках этой работы требуется разработать эту модель самому и применять в работе со словами.

Связать две переменные вызывается структура (ключевое слово struct) — в ней будут описаны два поля — сам текст (набор символов, образующих слово) и маркер. Выглядит это так (рис. 3):



рис. 3 — описание структуры Word

Такое описание данных образует новый тип — Word, но при этом не указываются инструменты взаимодействия с ним, поэтому вся работа со структурой открыта (обращения к полям и их изменение). Таким же образом для удобства были собраны и файловые переменные в одну структуру Files. Опять же, без инструментов взаимодействия, вся работа с экземпляром типа Files тоже открыта.

2 часть: модельное представление слова — массив символов, имеющий определенную длину. Эта реализация является нестандартной, принципы изменения слова поменялись, но не изменились принципы обхода слова, что очень удобно для этой задачи.

Однако на этот раз требуется связать переменные с функциями, которые с ними работают. Для этого используется класс (ключевое слово class) – в нем будут описаны и поля, и методы (функции-обработчики и функции-интерфейсы). Поскольку в задаче рассматривается предложение из слов, в решении будет реализовано два класса слов — Word (базовый, в нем описан интерфейс взаимодействия с отдельным словом) и Words (массив слов, в нем описаны обработчики, которые решают задачу).

Для Word получилось следующее описание (рис. 4):

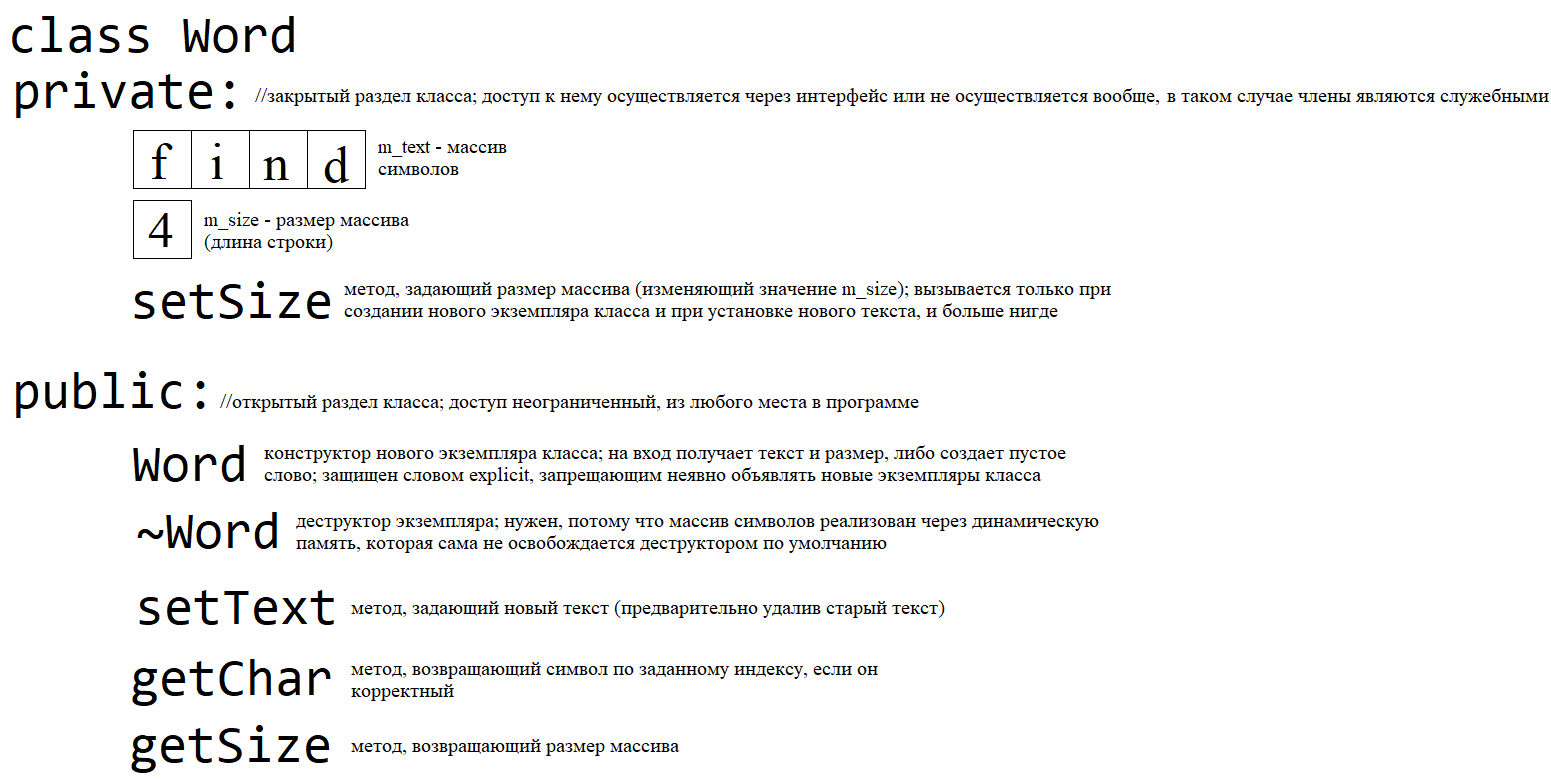


рис. 4 — описание класса Word

Для Words (рис. 5):

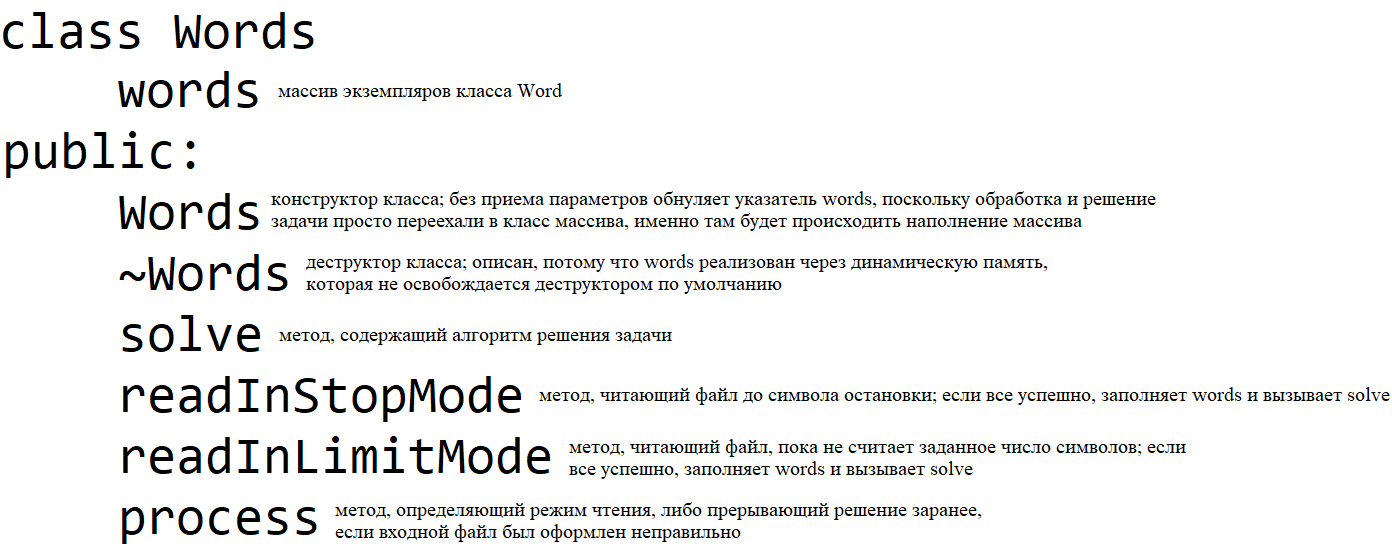


рис. 5 — описание класса Words

Как и в случае со структурами, Word и Words также образуют новые типы, со своим описанием. Также было решено дополнить структуру Files до класса, дописав к ней конструктор и деструктор — стандартные методы не могут открывать и закрывать файлы.

**Разработка интерфейса пользователя.**

Входной файл (режим стоп-символа):

I1: цифра 1

I2: c (кроме пробела)

I3: c (кроме пробела и указанного в I3 символа)

I4: набор символов

Входной файл (режим кол-ва):

I1: цифра 2

I2: +dddd

I3: c (кроме пробела)

I4: набор символов

Лог-файл:

O1: Программа начала свою работу. Производится попытка определить параметр чтения.

O2: Считанный параметр не соответствует шаблонам (либо символ '1', либо '2'). Программа завершила свою работу. (шаблон I1 нарушен)

O3: Входной файл не найден. Программа завершила свою работу. (файл не найден)

Режим стоп-символа:

O4: Режим чтения: по стоп-символу. Производится попытка определить стоп-символ.

O5: Определен стоп-символ – (символ из I2). Производится попытка определить ключевой символ.

O6: Определен ключевой символ – (символ из I3). Производится попытка считать строку.

O7: Файл преждевременно достиг своего конца. Программа завершила свою работу. (если файл закончился сразу после O4 или O5)

O8: Считанный стоп-символ является пробельным символом - недопустимый символ. Программа завершила свою работу. (если нарушен шаблон I2)

O9: Считанный ключевой символ является пробельным символом - недопустимый символ. Программа завершила свою работу. (если нарушен шаблон I3)

O10: Считанный ключевой символ ранее уже был зарезервирован под стоп-символ - ошибка. Программа завершила свою работу. (если был нарушен шаблон I3)

O11: Файл закончился до стоп-символа, однако предложение все равно будет обработано.

O12: Предложение дошло до перехода на новую строку; обработке подлежит только часть до перехода.

O13: В файле не было обнаружено ни одного слова. Ответ не получен. Программа завершила свою работу.

Режим кол-ва символов:

O4: Режим чтения: по заданному числу символов. Производится попытка определить число символов.

O5: Определенно заданное кол-во символов – (число из I2). Производится попытка считать ключевой символ.

O6: Определен ключевой символ – (символ из I3). Производится попытка считать строку.

O7: Файл преждевременно достиг своего конца. Программа завершила свою работу. (если файл закончился сразу после O4 или O5)

O8: Считанное число символов не является положительным - ошибка. Программа завершила свою работу. (если нарушен шаблон I2)

O9: Считанный ключевой символ является пробельным символом - недопустимый символ. Программа завершила свою работу. (если нарушен шаблон I3)

O10: До конца предложения было считано меньше символов, чем задано. Программа обработает все, что было считано.

O11: Кол-во корректно считанных символов не соответствует заданному. Ответ не получен. Программа завершила свою работу.

O14: В массив были записаны следующие слова:

O15: последовательность строк

O16: Производится попытка решить задачу - найти в предложении последнее слово, начинающееся на ключевой символ.

O17: Найдено слово-ответ на задачу: (слово-ответ). Программа завершила свою работу.

O18: В предложении не было обнаружено ни одного слова, начинающегося на заданный символ. Ответ не получен. Программа завершила свою работу.

Выходной файл:

O1: Ошибка: входной файл не найден.

O2: Ошибка: указан некорректный параметр на чтение. Нужно указать либо символ '1', либо символ '2'. (если нарушен шаблон I1)

Режим стоп-символа:

O3: Ошибка: файл преждевременно достиг своего конца. (попытка определить стоп-символ)

O4: Ошибка: в качестве стоп-символа был указан пробельный символ - недопустимый символ!

O5: Ошибка: файл преждевременно достиг своего конца. (попытка определить ключевой символ)

O6: Ошибка: в качестве ключевого символа был указан пробельный символ - недопустимый символ!

O7: Ошибка: стоп-символ и ключевой символ совпадают.

O8: Ответ не получен: в файле не было обнаружено ни одного слова. (слово - непрерывная последовательность символов, ограниченная пробелами)

Режим кол-ва символов:

O3: Ошибка: файл преждевременно достиг своего конца. (попытка определить число символов)

O4: Ошибка: считанное число символов не является положительным - недопустимое значение!

O5: Ошибка: файл преждевременно достиг своего конца. (попытка определить ключевой символ)

O6: Ошибка: в качестве ключевого символа был указан пробельный символ - недопустимый символ!

O7: Ответ не получен: кол-во корректно считанных символов не соответствует заданному. Строка читается до перехода на новую строку.

O9: Ответ не получен: в предолжении нет слова, начинающегося на заданный символ.

O10: Последнее слово в предложении, оканчивающееся на символ (символ из I3) – (слово-ответ).

**Описание используемых данных.**

Структура Word:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Имя | Модификатор | Назначение |
| char | mark | public | хранение маркера |
| char\* | text | посимвольное хранение слова с маркером |

Структура Files:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Имя | Модификатор | Назначение |
| std::ifstream | inFile | public | входной файл |
| std::ofstream | outFile | выходной файл |
| logFile | лог-файл |

Класс Word:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Имя | Модификатор | Назначение |
| char\* | m\_text | private | посимвольное хранение слова |
| int | m\_size | хранение длины слова |

Класс Words:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Имя | Модификатор | Назначение |
| Word\* | words | private | хранение слов |

Класс Files по данным повторяет аналогичную структуру, исключая дополнительный функционал класса.

Остальные переменные:

|  |  |  |
| --- | --- | --- |
| Тип | Имя | Назначение |
| Files\*, Files | files (main, process, readInStopMode, readInLimitMode, solve) | экземпляр типа Files |
| Word\* | words | массив объектов типа Word |
| char | tmp (readInStopMode, readInLimitMode) | универсальный символ-заглушка |
| stop (readInStopMode) | хранение стоп-символа |
| key (readInStopMode, readInLimitMode, solve) | хранение ключевого символа |
| mode (process) | хранение считанного режима чтения |
| char\* | chars (readInLimitMode) | массив для всех считанных символов строки |
| int | startPos (readInStopMode, readInLimitMode) | хранение значения позиции начала строки |
| charsCount (readInStopMode, readInLimitMode) | подсчет непробельных символов; в режиме по кол-ву символов до обработки строки — считанное число символов |
| wordsCount (readInStopMode, readInLimitMode) | подсчет слов в строке |
| curPos (readInStopMode, readInLimitMode) | хранение значения текущей позиции для записи слов |
| curIndex (readInStopMode, readInLimitMode) | подсчет индекса для очередного слова |
| i, j (solve) | итераторы циклов while |

**Описание используемых функций.**

Для функций, обрабатывающих входной файл и решающих задачу, описание одно на обе части, поскольку разница лишь в обращении к данным. Также в этой таблице будут описаны методы, принадлежащие классам (в том числе, конструкторы и деструкторы).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя | Назначение | Параметры | | | |
| входные | выходные | модифицируемые | транзитные |
| process | первичная обработка входного файла, попытка определить режим считывания |  |  |  | files (переходит в readInStopMode, readInLimitMode, solve) |
| readInStopMode | считывание файла и обработка строки в режиме стоп-символа |  |  |  | words, wordsCount, key, files (переходят в solve) |
| readInLimitMode | считывание файла и обработка строки в режиме кол-ва символов |  |  |  | words, wordsCount, key, files (переходят в solve) |
| solve | решение задачи после всех обработок |  |  | files (запись сообщений) |  |
| Word | конструктор Word (выделяет память и заполняет ее) | text (по умолчанию пустая строка) |  |  | size (по умолчанию 1) (переходит в setSize) |
| ~Word | деструктор Word (освобождает память) |  |  |  |  |
| setSize (Word) | задание размера объекту класса Word | size |  |  |  |
| setText (Word) | задание содержимого объекту класса Word | text |  |  | size (переходит в setSize) |
| getChar (Word) | возврат символа по индексу | index |  |  |  |
| getSize (Word) | возврат размера |  |  |  |  |
| Words | конструктор Words (на деле заглушка для инициализации указателя) |  |  |  |  |
| ~Words | деструктор Words (освобождает память) |  |  |  |  |
| Files | конструктор Files (открывает все файлы) | in (по умолчанию in.txt), out (по умолчанию out.txt), log (по умолчанию log.txt) |  |  |  |
| ~Files | деструктор Files (закрывает все файлы) |  |  |  |  |

**Организация ввода-вывода.**

|  |  |  |  |
| --- | --- | --- | --- |
| Библиотека | Имя | Команда | Назначение |
| fstream | inFile | >> | считывание с входного файла |
| eof | проверка, достигнут ли EOF |
| unsetf | снятие заданного флага считывания |
| is\_open | проверка, открыт ли файл |
| clear | снятие всех флагов |
| seekg | перевод указателя на заданную позицию |
| tellg | возврат текущей позиции указателя |
| inFile, outFile, logFile | close | закрытие файла |
| open | открытие файла |
| outFile, logFile | << | вывод в файл строки |

**Представление алгоритма решения задачи.**

При открытии файла считываем цифру 1 или 2, режим считывания. Если там нет цифры 1 или 2, или там что-то другое, или файл обрывается на этом моменте, прерываем обработку. На цифру 1 назначено считывание в режиме стоп-символа, 2 — по кол-ву символов.

Считывание в режиме стоп-символа:

Считываем стоп-символ. Если там ничего нет, или файл обрывается на этом моменте, прерываем обработку. Далее считываем ключевой символ. Если происходит описанное выше или ключевой символ совпадает со стоп-символом, также прерываем обработку. Принцип обработки строки:

1) Посимвольное считывание

2) Подсчет непробельных символов

3) Каждый пробельный символ, а также переход на новую строку, стоп-символ и конец файла сбрасывают счетчик

4) Если счетчик был больше 0 на момент сброса, значит, зафиксировано слово

5) После первого считывания выделяется память под слова, курсор переводится в начало строки

6) Второе считывание похоже на первое, только теперь фиксируется начало каждого слова, подсчитывается его размер, после чего выделяется память и заполняется всеми символами

Если никаких слов не было образовано, прервать обработку.

Считывание в режиме кол-ва символов:

Считываем число. Если там ничего нет, или число не положительное, или файл обрывается на этом моменте, прерываем обработку. Далее считываем ключевой символ. Если происходит описанное выше, также прерываем обработку. Принцип обработки строки:

1) Сразу выделяется память размером считанного числа

2) Записываются все символы, которые встретятся в строке до перехода на новую строку или конца файла

3) Если в массиве до остановки не нашлось ни одного символа, прервать обработку

Определение слов и их запись повторяет алгоритм, описанный выше.

Решение задачи:

Начать сканирование справа налево. Если первый символ очередного слова совпадает с ключевым, вывести его и прекратить решение задачи. Если ни одного слова не было найдено, сообщить об этом.

**Блок-схема.**

|  |
| --- |
| main |
| C:\Users\DOM\Desktop\main_struct.png |
| process |
| C:\Users\DOM\Desktop\process_struct.png |
| readInStopMode |
| C:\Users\DOM\Desktop\readstop.png |
| readInLimitMode |
| C:\Users\DOM\Desktop\readlimit.png |
| solve |
| C:\Users\DOM\Desktop\solve.png |

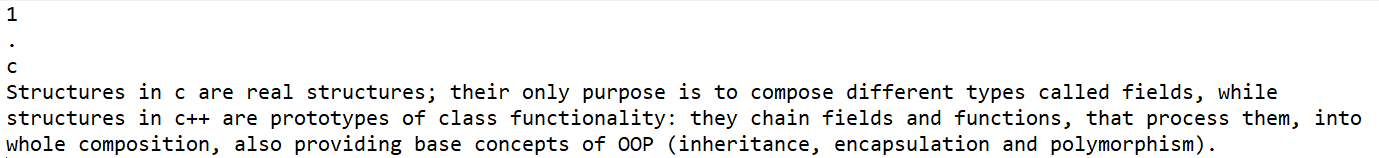
Как и в случае описания функций, блок-схемы представлены и для 1 части, и для 2.

**Текст программы.**

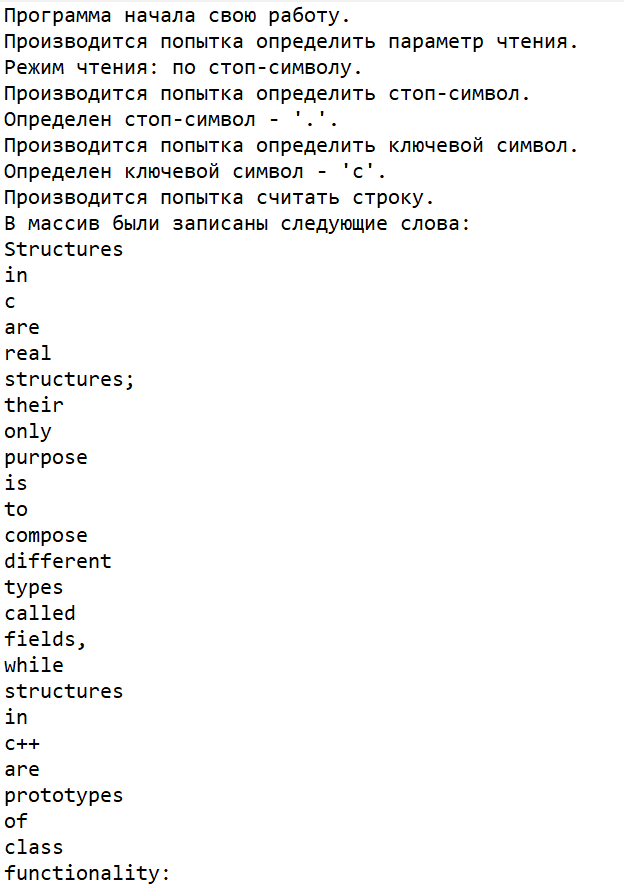
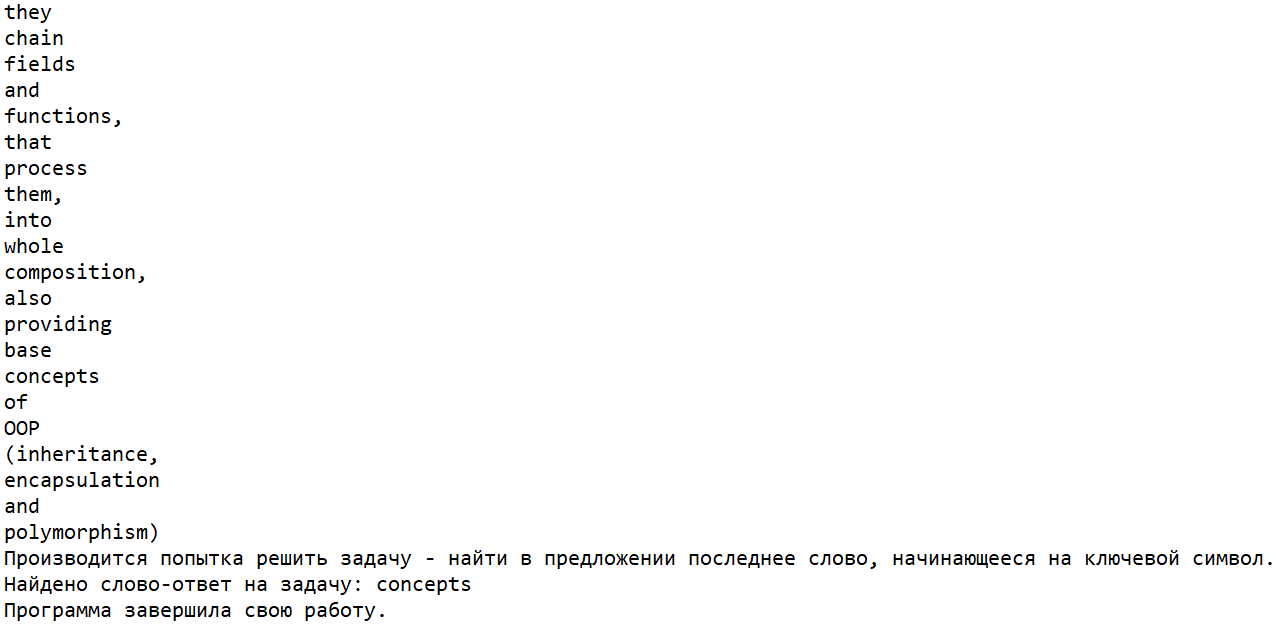
|  |
| --- |
| main (слева 1 часть, справа 2) |
|  |
| process |
|  |
| readInStopMode |
|  |
| readInLimitMode |
|  |
| solve |
|  |
| структуры |
|  |
| классы |
|  |
| методы классов (помимо методов-обработчиков) |
|  |

**Результаты работы программы.**

На примере следующего входного файла:

получены следующие результаты:

Лог-файл

Выходной файл:



**Вывод о проделанной работе.**

Изучены основные модельные представления строк, освоены на практике (а именно при создании разных реализаций одной и той же задачи) базовые концепции ООП.