МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

им. В.И. Ульянова (Ленина)»

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Пояснительная записка

к курсовой работе

на тему «Электронная картотека»

по дисциплине «Программирование»

Выполнил студент гр. 9308: Семенов А.И.

Проверил к. т. н. доцент: Сискович Т. И.

Санкт-Петербург

2020

**С**одержание

[Введение 4](#_Toc28068394)

[1. Задание 4](#_Toc28068395)

[2. Уточнение задания 4](#_Toc28068396)

[3. Контрольные примеры 6](#_Toc28068397)

[4. Краткое описание алгоритма 7](#_Toc28068398)

[5. Структура вызова функций 8](#_Toc28068399)

[6. Функции 9](#_Toc28068400)

[6.1 Главная функция 9](#_Toc28068401)

[6.2 Функция Menu 11](#_Toc28068402)

[6.3 Функция InputData 12](#_Toc28068403)

[6.4 Функция CheckData 13](#_Toc28068404)

[6.5 Функция Help 14](#_Toc28068405)

[6.6 Функция GenerateDelimiterArray 15](#_Toc28068406)

[6.7 Функция InputKeyWord 17](#_Toc28068407)

[6.8 Функция InputTextConsole 19](#_Toc28068408)

[6.9 Функция InputTextFile 21](#_Toc28068409)

[6.10 Функция ShowDelimiters 25](#_Toc28068410)

[6.11 Функция ShowKeyWord 26](#_Toc28068411)

[6.12 Функция ShowTextConsole 27](#_Toc28068412)

[6.13 Функция SaveResult 28](#_Toc28068413)

[6.14 Функция ShowResult 30](#_Toc28068414)

[6.15 Функция free\_str 32](#_Toc28068415)

[6.16 Функция free\_text 33](#_Toc28068416)

[6.17 Функция AmountOfWords 33](#_Toc28068417)

[6.18 Функция IsDelimiter 35](#_Toc28068418)

[6.19 Функция ProcessText 36](#_Toc28068419)

[7. Текст программы 39](#_Toc28068420)

[8. Результаты тестирования программы 58](#_Toc28068421)

[9. Инструкция пользователю 61](#_Toc28068422)

[Заключение 62](#_Toc28068423)

# Введение

Целью курсовой работы является законченное поэтапное решение содержательной задачи (постановка задачи, спецификация, выбор структур данных и разработка алгоритма, программная реализация, тестирование).

# Задание

Создать электронную картотеку, хранящуюся на диске. С картотекой возможны следующие действия: занесение данных, внесение изменений(исключение, корректировка, добавление), поиск данных, сортировка, отбор по определенным условиям. Вывод результаты работы и сделать возможность сохранения на диске.

# Уточнение задания

При выполнении задания следует учесть:

1. Максимальный размер строки - 80 символов, т. к. стандартный размер строки в командной оболочке – 80.
2. Нужно построить меню таким образом, чтобы пользователь не смог начать изменять картотеку, пока он не введет ее.
3. Количество критериев для обработки исходных данных – 2: рейтинг выше 7 и цена на 98 бензин ниже 54
4. Из подменю должна быть возможность выйти.

Меню программы имеет следующую иерархию:

1 – Справка

2 – Ввод данных

2.1 – Добавить в начало

2.2 – Добавить в конец

2.3 – Добавить на определенное место

2.4 – Ввод полностью заново

2.5 – Получить данные из файла

2.6 - Назад

3 – Контрольный вывод данных

3.1 – Вывод в виде таблицы

3.2 – Вывод в виде блоков построчно

3.3 – Вывод в виде таблицы, начиная с конца

3.4 – Вывод в виде блоков построчно. начиная с конца

3.5 – Вывод в файл

3.6 - Назад

4 – Действия над картотекой

4.1 – Узнать количество элементов картотеки

4.2 – Удалить элемент картотеки

4.3 – Поменять местами 2 элемента картотеки

4.4 – Сортировка картотеки

4.5 – Поиск элементов по заданным параметрам

4.6 – Изменить данные элемента картотеки

4.7 – Назад

5 – Обработка данных

6 – Вывод результата

6.1 – Вывод в виде таблицы

6.2 – Вывод в виде блоков построчно

6.3 – Вывод в виде таблицы, начиная с конца

6.4 – Вывод в виде блоков построчно. начиная с конца

6.5 – Вывод в файл

6.6 - Назад

0 – выход.

Пункты меню 3,4,5,6 недоступны, пока пользователь не введет все исходные данные, также выводится соответствующее сообщение. Пункт 6 недоступен, пока пользователь не выполнит обработку(пункт 5). Если пользователь выберет несуществующий пункт меню, то нужно ему сообщить об этом и предложить выбрать пункт меню заново.

# Контрольные примеры

Контрольные примеры приведены в таблице 1.

Таблица 1. Контрольные примеры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  пр | Исходные данные | | | Результат |
| Разделители | Введённый текст | Слово |
| 1 | ‘; | word1'word2d;fkoo  vbmmq;saop  sssqqsqs'  ioioioioi;  word;word'wword | KRS | word1'KRS'word2d  vbmmq;KRS  sssqqsqs'  ioioioioi;KRS  word;word'wword'KRS |
| 2 | 123 | jjjhhu1klsadkl2klsaklas  dsaio2iosaias2io3io  asdjk  faspopqdqs  f1fff2ffsa3asqqw | KWORD | jjjhhu1klsadkl2KWORD  dsaio2KWORD1iosaias2KWORD1io  asdjk1KWORD  faspopqdqs  f1KWORD1fff2KWORD1ffsa3asqqw1KWORD |
| 3 | /, | Notepad++/notepad--  office,sea,machine  people/who,see,me  dont,see/sea,sooo  soap,ice,cream/bitter | SIXLEN | Notepad++/SIXLEN/notepad--/SIXLEN  office,sea,SIXLEN/machine/SIXLEN  people/who,SIXLEN/see,SIXLEN  dont,see/SIXLEN/sea,SIXLEN |
| 4 | , . | Any text here  Menu,help.  Text.words  ..result is thisline..  input keyword after spacebar | EIGHTLEN | Any EIGHTLEN text  Menu,help.  Text.words EIGHTLEN  ..result is thisline.. |

# Краткое описание алгоритма

Начало программы.

Шаг №1. Вывод меню (функция *Menu*).

Шаг №2. Выбор пользователем пункта меню.

Шаг №3. Переход к пункту, выбранным пользователем:

Пункт 1-ый: Вывод справки для пользователя(функция *Help*)

Пункт 2-ой: Ввод исходных данных(функция *InputMenu*), открытие подменю.

Пункт 2.1-ый: Добавление данных в начало(функция *PushForward*) и переход к шагу 2.

Пункт 2.2-ой: Добавление данных в конец(функция *PushBack*) и переход к шагу 2.

Пункт 2.3-ий: Добавление данных на определенное место(функция *PushAnyPlace*) и переход к шагу 2.

Пункт 2.4-ый: Ввод данных полностью заново(функция *InputStations*) и переход к шагу 2.

Пункт 2.5-ый: Получение данных из файла(функция *InputTextFile*) и переход к шагу 2.

Пункт 2.0-ый: Переход к шагу 1.

Пункт 3-ий: Контрольный вывод данных(функция *OutputMenu*) открытие подменю.

Пункт 3.1-ый: Вывод данных в виде таблицы(функция *OutputGasStationsTable*) и переход к шагу 3.

Пункт 3.2-ой: Вывод данных в виде блоков построчно(функция *OutputGasStationText*) и переход к шагу 3.

Пункт 3.3-ий: Вывод данных в виде таблицы, начиная с конца(функция *OutputGasStationsTableReverse*) и переход к шагу 3.

Пункт 3.4-ый: Вывод данных в виде блоков построчно, начиная с конца(функция *OutputGasStationTextReverse*) и переход к шагу 3.

Пункт 3.0-ый: Переход к шагу 1.

Пункт 4-ый: Вывод подменю действий над картотекой(функция *ListActions*)

Пункт 4.1-ый: Вывод количества данных в картотеке(функция *ListLen*) и переход к шагу 4.

Пункт 4.2-ой: Удаление элемента картотеки(функция *DeleteItem*) и переход к шагу 4.

Пункт 4.3-ий: Смена мест 2-х элементов(функция *Swap*) и переход к шагу 4.

Пункт 4.4-ый: Сортировка картотеки(функция *Sort*) и переход к шагу 4.

Пункт 4.5-ый: Поиск элементов по заданным параметрам(функция *GetItem*) и переход к шагу 4.

Пункт 4.6-ый: Изменение данных элемента(функция *EditItem*) и переход к шагу 4.

Пункт 4.0-ый: Переход к шагу 1.

Пункт 5-ый: Обработка результата(функция *Process*)

Пункт 6-ый: Вывод результата(функция *OutputMenu*), открытие подменю

Пункт 6.1-ый: Вывод данных в виде таблицы(функция *OutputGasStationsTable*) и переход к шагу 6.

Пункт 6.2-ой: Вывод данных в виде блоков построчно(функция *OutputGasStationText*) и переход к шагу 6.

Пункт 6.3-ий: Вывод данных в виде таблицы, начиная с конца(функция *OutputGasStationsTableReverse*) и переход к шагу 6.

Пункт 6.4-ый: Вывод данных в виде блоков построчно, начиная с конца(функция *OutputGasStationTextReverse*) и переход к шагу 3.

Пункт 6.0-ый: Переход к шагу 6.

Пункт 0: Переход к шагу 1.

Пункт 0-ой: Переход к шагу 4.

Шаг №4. Конец программы.

# Структура вызова функций

Структура вызова функций представлена на рисунке 1.

*Рисунок 1. Структура вызова функций*

# Функции

# Текст программы

# Результаты тестирования программы

Тестирование программного обеспечения — это процесс исследования ПО с целью выявления ошибок и определения соответствия между реальным и ожидаемым поведением ПО, осуществляемый на основе набора тестов, выбранных определённым образом.

Для поиска ошибок были выбраны тесты:

1. Тест первый:

Текст:

*inputOanyLtext*

*pressfO*

*OOhelloOO*

*threeLwordsLhere*

*textLnotLwritten*

*OLlettersLO*

Ожидаемый результат:

Пользователь не сможет ввести строки 5,6, так как в 4-й строке содержится кол-во слов, равное максимальному кол-ву, встреченному ранее. должна вывести:

*inputOTETIOanyLTETI*

*pressfO*

*OOhelloOTETIO*

*threeLTETIOwordsLTETI*

Программа отработала правильно при таком тесте. Результат выполнения приведен на рисунке 22.

*Рисунок 22. Результат выполнения программы*

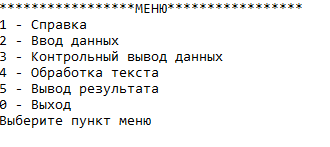
Во время тестирования ошибок в работе программы не обнаружено.

# Инструкция пользователю

При запуске программы Вы увидите меню, такое же, как показано на рисунке 24. Вас просят выбрать пункт меню. При вводе 1, Вы получите справку о том, для чего предназначена программа. При вводе 2, Вам откроется подменю, в котором Вы сможете выбрать, что хотите ввести: разделители, ключевое слово, текст(из консоли или из файла). Текст вводится только после ввода разделителей.

В 3-м пункте меню Вы можете посмотреть, что было введено Вами и проверить правильность ввода.

При выборе пункта 4 программа обработает заданный Вами текст по правилам, указанным в справке(пункт 1 меню). Если Вы не зададите все исходные данные(разделители, слово, текст), то обработка будет недоступна, и Вам будет об этом сообщено. После обработки текста Вы сможете выбрать пункт 5, в котором представится выбор, куда вывести результат обработки: в консоль, в файл. При выборе пункта 0 Вы, соответственно, завершите работу программы.



*Рисунок 24. Меню программы*

# Заключение

При выполнении курсовой работы были получены практические навыки поэтапного решение содержательной задачи, связанной с обработкой текстовой информации.