**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Алгоритмы и структуры данных»**

Тема: Стек, очередь, дек

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Виноградов К.А. |
| Преподаватель |  | Балтрашевич В.Э. |

Санкт-Петербург

2017

**Цель работы.**

Знакомство с такими структурами данных, как стек, очередь и дек, реализация и построение алгоритмов, использующих их.

**Текст задания.**

Содержимое заданного текстового файла F, разделенного на строки, переписать в текстовый файл G, выписывая литеры каждой строки в обрат¬ном порядке. В решении задачи использовать стек.

**Организация стека.**

Для решения поставленной задачи используем стек. Стек представляет собой структуру типа «LIFO», где элемент, попавший в стек последним, извлекается оттуда раньше всех. В программе стек реализован как однонаправленный список структур с двумя полями: данных и указателем на следующую структуру. Кроме этого нам понадобятся две функции для работы со стеком – Push и Pop. Функция Push позволяет поместить элемент на верхушку стека, а Pop – извлечь верхнее значение. Реализация стека и функций, связанных с ним представлена в приложении А.

**Ход работы.**

В данном отчете представлены лишь функции, написанные для решения задачи. Все иные, автоматически сгенерированные файлы, в отчет не включены.

Сначала написан код, выполняющийся открытие отдельного окна выбора файла по нажатию вкладки «Open file», созданной с помощью встроенного в Qt редактора. В данном окне из специального поля «treeView», которое также создано с помощью редактора, считывается путь к файлу с исходными данными. Если по указанному пути файл не найден, то выводится соответствующее сообщение об ошибке. Если же файл найден, то эта строка преобразуется в формат QString для того, чтобы ее можно было использовать для открытия файла. Код данной функции представлен в приложении А.

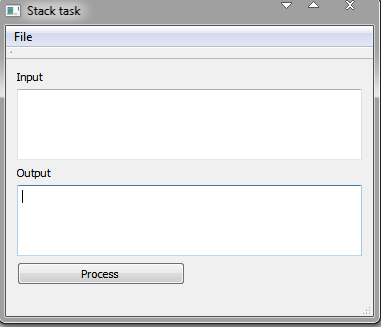
Данные считываются из файла, после чего создается стек путем выделения памяти под каждую его единицу, помещения символа в поле для данных и соединения стековых единиц в единое целое с помощью указателей. Затем стековые единицы по очереди вытаскиваются из стека, и значения поля данных каждой переносится на поле «textEdit» пользовательского интерфейса. Код программы представлен в приложении Б.

На основании результата работы программы в поле «textEdit» выводится результат, после чего файл закрывается и программа ожидает повторного запроса или завершения.

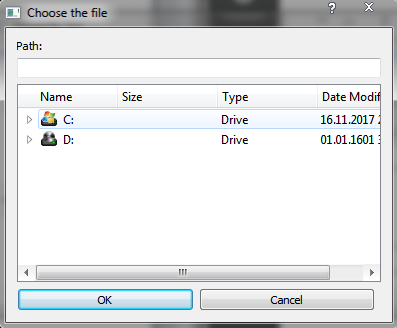
**Пользовательский интерфейс.**

Для более удобной работы программы был создан пользовательский интерфейс:

* основное окно программы



* окно выбора файла



**Тестирование программы.**

Таблица 1. Результаты тестирования программы

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| abcdefghij | jihgfedcba |
| РаБоТа Со СтЕкОм | мОкЕтС оС аТоБаР |
| Doing reports is boooring zzz....... | .......zzz gnirooob si stroper gnioD |
| Empty ex(nope) | )epon(xe ytpmE |
| \0 | The file is empty |

**Вывод.**

В результате выполнения данной лабораторной работы были построены функции для обработки строки с использованием стека, а также созданапрограмма, использующая данные функции.

**ПРИЛОЖЕНИЕ А**

**Реализация стека и функций для работы с ним**

typedef struct elem

{

QChar liter;

elem\* next;

}elem;

elem\* **pop**(elem\* head)

{

if(head==NULL)

return head;

elem\* tmp=new elem;

if((head->next==NULL)){

tmp->liter=head->liter;

head->liter='\0';

}

else

{

while(head->next->next)

head=head->next;

tmp=head->next;

head->next=NULL;

}

return tmp;

}

void **push**(elem\* head, QChar lit)

{

while(head->next)

head=head->next;

elem\* tmp=new elem;

tmp->liter=lit;

head->next=tmp;

tmp->next=NULL;

}

**ПРИЛОЖЕНИЕ Б**

**Функция обработки стека**

void MainWindow::**on\_Enter\_clicked**()

{

if(buff.isEmpty())

{

ui->textEdit->setText("The file is empty");

return;

}

QFile output("output.txt");

if(!output.*open*(QFile::WriteOnly|QFile::Text|QFile::Append))

{

QMessageBox::information(this, "Error", "Path is not correct!");

return;

}

QTextStream writestream(&output);

int i=0;

ui->textEdit->clear();

while(buff[i]!='\0'){

elem\* head=new elem;

elem\* tmp=new elem;

head->liter=buff[i];

head->next=tmp;

tmp->next=NULL;

i++;

while((buff[i+1]!='\n')&&(buff[i+1]!='\0'))

{

elem\* tmp\_next=new elem;

tmp->liter=buff[i];

tmp->next=tmp\_next;

tmp=tmp\_next;

tmp\_next->next=NULL;

tmp->next=NULL;

i++;

}

tmp->liter=buff[i];

tmp->next=NULL;

i+=2;

while(head->liter!='\0')

{

QChar c=pop(head)->liter;

ui->textEdit->insertPlainText(c);

writestream<<c;

}

ui->textEdit->insertPlainText("\n");

writestream<<"\n";

}

output.*close*();

}