Отчёт по лабораторной работе №25-26 по курсу Практикум на ЭВМ

Студент группы М8О-111Б Тимирчев Янис Федорович, № по списку 23

Контакты www, e-mail, icq, skype yanis.timirchev@yandex.ru

Работа выполнена: «26» апреля 2022 г.

Преподаватель: доцент каф.806 Никулин С.П.

Входной контроль знаний с оценкой _____

Отчёт сдан « » _____201__ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Абстрактные типы данных. Рекурсия. Модульное программирование на языке СИ

2. **Цель работы:** Составить и отладить модуль определений и модуль реализации по заданной схеме модуля определений для абстрактного (пользовательского) типа данных (стека, очереди, списка или дека, в зависимости от варианта задания).

3. **Задание** (*вариант №*S1): вариант задания – Стек; процедура и метод – поиск и удаление максимального элемента (сортировка линейным выбором)

4. **Оборудование(лабораторное)**:
ЭВМ _____, процессор _____, имя узла сети_____ с ОП _____ Мб,
НМД_____Мб. Терминал_____адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:
Процессор Intel Core i5-7300HQ с ОП 16Гб Мб, НМД 1 ТБ Мб. Монитор 1920x1080 Full HD
Другие устройства _____

5. **Программное обеспечение(лабораторное):**
Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____
Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:
Операционная система семейства Linux, наименование Ubuntu версия 20.04.3
интерпретатор команд bash версия 5.0.17 (1).
Система программирования _____ версия _____
Редактор текстов _____ версия 26.2
Утилиты операционной системы _____
Прикладные системы и программы emacs
Местонахождение и имена файлов программ и данных на домашнем компьютере _____

6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

25: Проанализируем файловую структуру модуля. На ее основе создадим Makefile с зависимостями программных файлов модуля.

26: Отдельно реализуем модули дека на языке Си и реализуем его методы. Модуль имеет операции добавления в начало, удаления из начала, печати стека, сортировки.

7. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

1.makefile
2.Тесты с пустым деком
3.Проверка функций
4.Попытка “ переполнить ” стек

*Пункты 1-7 отчета составляются **строго до** начала лабораторной работы.*

Допущен к выполнению работы. Подпись преподавателя _____

8. **Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

sort.c:

```
#include "sort.h"

void udtSwap(Udt *udt1, Udt *udt2)
{
Udt tmp;

tmp = *udt1;
*udt1 = *udt2;
*udt2 = tmp;
}

void udtSelectionSort(Udt *udt)
{
const int cap = udtCapacity(udt); Udt tmp;
UDT_TYPE tempItem;

if (cap < 2) return;
udtCreate(&tmp, cap); while (!udtEmpty(udt))
{
tempItem = udtTop(udt); udtPop(udt);
while (!udtEmpty(&tmp) && udtTop(&tmp).key > tempItem.key)
{
udtPush(udt, udtTop(&tmp)); udtPop(&tmp);
}
udtPush(&tmp, tempItem);
}
udtSwap(&tmp, udt);
}
```

sort.h:

```
#ifndef UDT_SORT_H
#define UDT_SORT_H

#include "udt.h"

void udtSwap(Udt *udt1, Udt *udt2); void udtSelectionSort(Udt *udt);

#endif
```

udt.c:

```
#include "udt.h"

void udtCreate(Udt *udt, const int capacity)
{
int i;
UDT_TYPE item;

item.key = 0.0f; item.str[0] = '\0';

if (capacity <= 0) return;
udt->data = (UDT_TYPE *)malloc(sizeof(UDT_TYPE) * capacity);
for (i = 0; i < capacity; i++)
udt->data[i] = item;

udt->capacity = capacity; udt->size = 0;
}
```

```
int udtPush(Udt *udt, const UDT_TYPE value)
{
if (udt->size == udt->capacity) return 0;

udt->data[udt->size++] = value;

return 1;
}

void udtPop(Udt *udt)
{
if (udt->size == 0) return;

udt->size--;
}

UDT_TYPE udtTop(const Udt *udt)
{
return udt->data[udt->size - 1];
}

int udtSize(const Udt *udt)
{
return udt->size;
}

int udtEmpty(const Udt *udt)
{
return udt->size == 0;
}

void udtPrint(Udt *udt)

{
int i;
Item item;
printf("Номер\tКлюч\t\tСтрока\n"); for (i = 0; i < udtSize(udt); i++)
{
item = udt->data[i];

printf("%d\t%.3f\t\t%s\n", i + 1, item.key, item.str);
}
}

void udtDestroy(Udt *udt)
{
if (udt->data != NULL)
{
free(udt->data);

udt->data = NULL;
}

udt->capacity = 0;
udt->size = 0;
}
int udtCapacity(const Udt *udt)
{
return udt->capacity;
}
```

udt.h

```
#ifndef UDT_H
#define UDT_H

#include <stdio.h> #include <stdlib.h>

typedef struct
{
float key; char str[31];
} Item;
typedef Item UDT_TYPE; typedef struct
{
UDT_TYPE *data; int capacity;
int size;
} Udt;

void udtCreate(Udt *udt, int capacity); int udtPush(Udt *udt, UDT_TYPE value); void udtPop(Udt *udt);

UDT_TYPE udtTop(const Udt *udt); int udtSize(const Udt *udt);
int udtEmpty(const Udt *udt); void udtPrint(Udt *udt);
void udtDestroy(Udt *udt);
int udtCapacity(const Udt *udt); #endif
```

main.c

```
#include <stdio.h> #include "sort.h"
void getLine(char *str, int size); int main()
{
const int N = 10; int action;
char tmpCh; Udt udt; Item item;

udtCreate(&udt, N);

do
{
printf("Меню\n");
printf("1) Добавить элемент\t2) Удалить элемент\n"); printf("3) Размер стека\t\t4) Сортировка\n"); printf("5) Печать\t\t6) Выход\n");
printf("Выберите действие: "); scanf("%d", &action);
switch (action)
{
case 1:
{
printf("Введите ключ: "); scanf("%f", &item.key);
scanf("%c", &tmpCh); printf("Введите Строку: "); getLine(item.str, sizeof(item.str));




item.str);

}

if (udtPush(&udt, item))
printf("Элемент с ключом %f и строкой '%s' добавлен успешно\n", item.key,

else
printf("Стек полон\n"); break;

case 2:
{
if (udtSize(&udt) > 0)
{
```

```
item = udtTop(&udt); udtPop(&udt);
printf("Элемент с ключом %f и строкой '%s' удален успешно\n", item.key,
item.str);
}
else
printf("Стек пуст\n");
}
break;

case 3:
{
printf("Размер стека: %d (Реальный размер: %d)\n", udtSize(&udt), N);
}
break;

case 4:
{
if (udtSize(&udt) > 1) udtSelectionSort(&udt);
else
printf("Должно быть хотя бы 2 элемента\n");

break;
}

case 5:
{
if (udtSize(&udt) > 0)
{
printf("Стек:\n");

udtPrint(&udt);
}
else
printf("Стек пуст\n");
}
break;

case 6: break;

default:
{
printf("Ошибка. Такого пункта меню не существует\n");
}
break;
}
}
while (action != 6); udtDestroy(&udt);
return 0;
}

void getLine(char *str, const int size)
{
int cnt = 0, ch;

while ((ch = getchar()) != '\n' && cnt < size - 1) str[cnt++] = ch;

str[cnt] = '\0';
}
```

Тесты:

```
ianis@dev1l:~/l26$ ls
main.c makefile sort.c sort.h udt.c udt.h
ianis@dev1l:~/l26$ cat makefile
prog: main.o sort.o udt.o
cc -o prog main.o sort.o udt.o main.o: main.c
cc -c main.c sort.o: sort.c sort.h
cc -c sort.c udt.o: udt.c udt.h
cc -c udt.c
ianis@dev1l:~/l26$ make cc -c main.c
cc -c sort.c cc -c udt.c
cc -o prog main.o sort.o udt.o
ianis@dev1l:~/l26$ ls
main.c main.o makefile prog sort.c sort.h sort.o udt.c udt.h udt.o
ianis@dev1l:~/l26$ ls -l
total 68
-rwxrwxr-x 1 ubuntu ubuntu 3545 May 27 18:10 main.c
-rw-rw-r-- 1 ubuntu ubuntu 7048 May 27 18:11 main.o
-rw-rw-r-- 1 ubuntu ubuntu 155 May 27 18:06 makefile
-rwxrwxr-x 1 ubuntu ubuntu 21616 May 27 18:11 prog
-rwxrwxr-x 1 ubuntu ubuntu 831 May 27 16:41 sort.c
-rwxrwxr-x 1 ubuntu ubuntu 158 May 27 16:41 sort.h
-rw-rw-r-- 1 ubuntu ubuntu  3392 May 27 18:08 sort.o
-rwxrwxr-x 1 ubuntu ubuntu  2451 May 27 17:36 udt.c
-rwxrwxr-x 1 ubuntu ubuntu 657 May 27 17:36 udt.h
-rw-rw-r-- 1 ubuntu ubuntu 4936 May 27 18:08 udt.o ianis@dev1l:~/l26$ ./prog
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 6
ianis@dev1l:~/l26$ make make: 'prog' is up to date.
ianis@dev1l:~/l26$ nano main.c

ianis@dev1l:~/l26$ make cc -c main.c
cc -o prog main.o sort.o udt.o ianis@dev1l:~/l26$ ls -l total 68

-rwxrwxr-x 1 ubuntu ubuntu 3561 May 27 18:22 main.c
-rw-rw-r-- 1 ubuntu ubuntu 7048 May 27 18:23 main.o
-rw-rw-r-- 1 ubuntu ubuntu 155 May 27 18:06 makefile
-rwxrwxr-x 1 ubuntu ubuntu 21616 May 27 18:23 prog
-rwxrwxr-x 1 ubuntu ubuntu 831 May 27 16:41 sort.c
-rwxrwxr-x 1 ubuntu ubuntu 158 May 27 16:41 sort.h
-rw-rw-r-- 1 ubuntu ubuntu  3392 May 27 18:08 sort.o
-rwxrwxr-x 1 ubuntu ubuntu  2451 May 27 17:36 udt.c
-rwxrwxr-x 1 ubuntu ubuntu 657 May 27 17:36 udt.h
-rw-rw-r-- 1 ubuntu ubuntu 4936 May 27 18:08 udt.o ianis@dev1l:~/l26$ ./prog
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 2
Стек пуст Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 4
Должно быть хотя бы 2 элемента Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 5
Стек пуст Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 3
Размер стека: 0 (Реальный размер: 10)
```

Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 1 Введите ключ: 999
Введите Строку: 123345
Элемент с ключом 999.000000 и строкой '123345' добавлен успешно Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 1 Введите ключ: 50.50 Введите Строку: qwerty
Элемент с ключом 50.500000 и строкой 'qwerty' добавлен успешно Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка

5) Печать 6) Выход Выберите действие: 1 Введите ключ: 100.100 Введите Строку: ioi21312
Элемент с ключом 100.099998 и строкой 'ioi21312' добавлен успешно Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 5
Стек:
Номер Ключ Строка
1 999.000 123345
2 50.500 qwerty
3 100.100 ioi21312
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 4
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 5
Стек:
Номер Ключ Строка 1 50.500 qwerty
2 100.100 ioi21312
3 999.000 123345
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 3
Размер стека: 3 (Реальный размер: 10) Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 1 Введите ключ: 1312 Введите Строку: 234234wer
Элемент с ключом 1312.000000 и строкой '234234wer' добавлен успешно
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 1 Введите ключ: 123
Введите Строку: tyuew
Элемент с ключом 123.000000 и строкой 'tyuew' добавлен успешно Меню
1) Добавить элемент 2) Удалить элемент

3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 1 Введите ключ: 10
Введите Строку: 12323
Элемент с ключом 10.000000 и строкой '12323' добавлен успешно Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 1 Введите ключ: 44
Введите Строку: qerrwq
Элемент с ключом 44.000000 и строкой 'qerrwq' добавлен успешно

Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 1 Введите ключ: 2134 Введите Строку: qwewe
Элемент с ключом 2134.000000 и строкой 'qwewe' добавлен успешно Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 1 Введите ключ: 666
Введите Строку: 6666
Элемент с ключом 666.000000 и строкой '6666' добавлен успешно Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 1 Введите ключ: 777
Введите Строку: 777rewte
Элемент с ключом 777.000000 и строкой '777rewte' добавлен успешно Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 3
Размер стека: 10 (Реальный размер: 10) Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 1 Введите ключ: 1223 Введите Строку: 123123 Стек полон
Меню
1) Добавить элемент 2) Удалить элемент

3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 5
Стек:
Номер Ключ Строка 1 50.500 qwerty
2 100.100 ioi21312
3 999.000 123345
4 1312.000 234234wer
5 123.000 tyuew
6 10.000 12323
7 44.000 qerrwq
8 2134.000 qwewe
9 666.000 6666
10 777.000 777rewte
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 4
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 5
Стек:
Номер Ключ Строка 1 10.000 12323
244.000 qerrwq
350.500 qwerty
4 100.100 ioi21312
5 123.000 tyuew
6 666.000 6666
7 777.000 777rewte
8 999.000 123345
9 1312.000 234234wer
10 2134.000 qwewe Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 2
Элемент с ключом 2134.000000 и строкой 'qwewe' удален успешно Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 2
Элемент с ключом 1312.000000 и строкой '234234wer' удален успешно Меню
1) Добавить элемент 2) Удалить элемент

3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 5

Стек:
Номер Ключ Строка
1 10.000 12323
2 44.000 qerrwq
3 50.500 qwerty
4 100.100 ioi21312
5 123.000 tyuew
6 666.000 6666
7 777.000 777rewte
8 999.000 123345
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 2
Элемент с ключом 999.000000 и строкой '123345' удален успешно Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 5
Стек:
Номер Ключ Строка
1 10.000 12323
2 44.000 qerrwq
3 50.500 qwerty
4 100.100 ioi21312
5 123.000 tyuew
6 666.000 6666
7 777.000 777rewte
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 4
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 5
Стек:
Номер Ключ Строка
1 10.000 12323
2 44.000 qerrwq
3 50.500 qwerty
4 100.100 ioi21312
5 123.000 tyuew
6 666.000 6666
7 777.000 777rewte
Меню
1) Добавить элемент 2) Удалить элемент
3) Размер стека 4) Сортировка
5) Печать 6) Выход Выберите действие: 6

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора** по существу работы _____

11. **Выводы**

Научился работать с таким типом данных как стек. Научился реализовывать стек на языке СИ, научился сортировать линейным выбором.

Недочёты при выполнении задания могут быть устранены следующим образом:_____

Подпись студента_____