**Отчет по лабораторной работе № 25-26** по курсу "Вычислительные Системы"

Студентка группы М8О-105Б-20 Агеева Алиса, № по списку 02

Контакты e-mail: alisa.ageewa2003@gmail.com

Работа выполнена: « 22 » мая 2022 г.

Преподаватель: доц. каф. 806 Никулин Сергей Петрович

Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан « » \_\_\_ 2020 г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Тема:** Абстрактные типы данных. Рекурсия. Модульное программирование на языке Си. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Автоматизация сборки программ модульной структуры на языке Си с использованием утилиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

make\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Цель работы:** Составить и отладить модуль определений и модуль реализации по заданной\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

схеме модуля определений для абстрактного(пользовательского) типа данных. Составить\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

программный модуль, сортирующий экземпляр указанного абстрактного типа данных заданным\_\_\_\_\_\_\_\_\_\_\_\_\_\_

методом, используя только операции, импортированные из модуля UDT. Автоматизировать\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

сборку программ модульной структуры с использованием утилиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ make\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Задание** (*вариант № 2* )**:** Пользовательский тип:

- АТД: Линейный список

- Процедура: поиск в списке двух элементов, идущих подряд, первый из которых больше второго.Если такие элементы найдены, их перестановка.

- Метод: сортировка методом пузырька(по значению)

Автоматизировать сборку программ этих модульных структур, используя утилиту make\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Оборудование** :

ЭВМ ustation10, процессор AMD Ryzen 5 3550-H с ОП 16 Гб,

НМД 512 Гб. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор AMD E2 7110 с ОП 4 Гб, НМД 500 Гб. Монитор встроенный\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Другие устройства не использовались\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Программное обеспечение:**

Операционная система семейства GNU\Linux, наименование Ubuntu версия 14.04 интерпретатор команд bash версия 4.3.11(1)-release

Система программирования GNU(gcc)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия 8.1\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Редактор текстов vim версия 7.4

Утилиты операционной системы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства GNU\Linux, наименование Ubuntu версия 18.04 интерпретатор команд bash версия 4.4.19(1)-release  
Система программирования GNU(gcc)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия 8.1\_\_\_\_\_\_\_\_\_\_\_\_\_   
Редактор текстов microsoft word 2019\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утилиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных

1. **Идея, метод, алгоритм**  решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)  
     
   Написать модуль, описывающий АДТ, отвечающий за объявление структуры и декларацию

функций.

Реализовать взаимодействие с пользователем через меню.

Полученные модули программы собрать с использованием утилиты make.

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

**list.h**  
  
#ifndef LIST\_H

#define LIST\_H

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#define MAX\_SIZE 64

typedef struct Data {

int key;

int value;

} Data;

typedef struct Iter {

Data \*node;

} Iter;

typedef struct List {

Data arr[MAX\_SIZE];

Data \*begin;

Data \*end;

int size;

} List;

List \*list\_create();

bool list\_is\_empty(List \*list);

Data \*Next(Iter \*cur);

Data \*begin(List \*list);

Data \*end(List \*list);

void list\_insert(List \*list, Data data, int index);

void list\_erase(List \*list, const int index);

Data list\_front(List \*list);

Data list\_back(List \*list);

int list\_size(List \*list);

void list\_print(List \*list);

void list\_copy(List \*list, List \*new\_list);

#endif  
**list.c**

#include "list.h"

#include "string.h"

List \*list\_create() {

List \*list;

list = (List \*) malloc(sizeof(List));

list->begin = &(list->arr[0]);

list->end = &(list->arr[0]);

list->size = 0;

return list;

}

Data list\_front(List \*list) {

return \*(list->begin);

}

Data list\_back(List \*list) {

return \*(list->end - 1);

}

Data \*Next(Iter \*cur) {

return cur->node + 1;

}

Data \*begin(List \*list) {

return list->begin;

}

Data \*end(List \*list) {

return list->end;

}

int list\_size(List \*list) {

return list->size;

}

bool list\_is\_empty(List \*list) {

return list\_size(list) == 0;

}

void list\_copy(List \*list, List \*new\_list) {

for (int i = 0; i < MAX\_SIZE; ++i) {

list->arr[i] = new\_list->arr[i];

}

list->begin = new\_list->begin;

list->end = new\_list->end;

list->size = new\_list->size;

}

void list\_insert(List \*list, Data val, int index) {

if (list\_size(list) == MAX\_SIZE) {

printf("The list is full!\n");

return;

}

if (index > list\_size(list)) {

printf("Cannot insert!\n");

return;

}

if (index == list\_size(list)) {

list->arr[index] = val;

list->end++;

list->size++;

return;

}

Data \*from = begin(list) + index;

memmove(from + 1, from, sizeof(Data) \* (end(list) - from));

list->arr[index] = val;

list->end++;

list->size++;

}

void list\_erase(List \*list, const int index) {

if (index >= list\_size(list)) {

printf("Out of bound!\n");

return;

}

if (index == list\_size(list) - 1) {

list->end--;

list->size--;

return;

}

Data \*src = begin(list) + index + 1;

int n\_bytes = sizeof(Data) \* (end(list) - src);

memmove(src - 1, src, n\_bytes);

list->end--;

list->size--;

}

void list\_print(List \*list) {

Iter \*cur = (Iter \*) (malloc(sizeof(Iter)));

cur->node = begin(list);

int i = 1;

for (; cur->node < end(list); cur->node = Next(cur)) {

printf("%d. value:%d key:%d\n", i, cur->node->value, cur->node->key);

i++;

}

}

**sort.h**

#ifndef LIST\_SORT\_H

#define LIST\_SORT\_H

#include "list.h"

int list\_sort(List \*list);

bool list\_procedure(List \*list);

#endif

**sort.c**

#include "sort.h"

bool list\_procedure(List \*list) {

bool isFound = false;

List \*new\_list = list\_create();

while (!list\_is\_empty(list)) {

Data elem = list\_front(list);

list\_erase(list, 0);

Data d = list\_back(new\_list);

if (!isFound && !list\_is\_empty(new\_list) && d.key > elem.key) {

Data tmp = list\_back(new\_list);

list\_erase(new\_list, list\_size(new\_list) - 1);

list\_insert(new\_list, elem, list\_size(new\_list));

list\_insert(new\_list, tmp, list\_size(new\_list));

isFound = true;

}

else{

list\_insert(new\_list, elem, list\_size(new\_list));

}

}

list\_copy(list, new\_list);

return isFound;

}

int list\_sort(List \*list) {

int i = 0;

while (list\_procedure(list)) {

i++;

}

return i;

}  
  
**main.c**

#include <stdio.h>

#include "sort.h"

#define MENU "1 - print list\n2 - insert\n3 - erase\n4 - get size\n5 - bubble sort\n"

int main() {

List \*list = NULL;

list = list\_create();

int s;

printf(MENU);

while (scanf("%d", &s) != EOF) {

switch (s) {

case 1: {

printf("\n");

if (list\_is\_empty(list)) {

printf("The list is empty!\n\n");

break;

}

list\_print(list);

printf("\n");

break;

}

case 2: {

Data d;

printf("Enter value and key:\n");

scanf("%d %d", &d.value, &d.key);

int index;

printf("Enter index to insert:\n");

scanf("%d", &index);

index--;

list\_insert(list, d, index);

break;

}

case 3: {

if (list\_is\_empty(list)) {

printf("The list is empty!\n");

break;

}

int index;

printf("Enter index:\n");

scanf("%d", &index);

index--;

list\_erase(list, index);

break;

}

case 4: {

printf("\n");

if (list\_is\_empty(list)) {

printf("The list is empty!\n\n");

break;

}

printf("The size is %d\n", list\_size(list));

printf("\n");

break;

}

case 5 : {

if (list\_is\_empty(list)) {

printf("The list is empty!\n");

break;

}

list\_sort(list);

printf("The list has been sorted!\n");

break;

}

default: {

printf("Incorrect command!\n");

}

}

printf(MENU);

}

return 0;

}

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

*Допущен к выполнению работы.*  **Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ ls*

*Makefile list.c list.h main.c sort.c sort.h*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ cat Makefile*

*GCCFLAGS=-w*

*run:all*

*./main*

*all:list.o sort.o main.o*

*gcc $(GCCFLAGS) list.o sort.o main.o -o main*

*list.o:*

*gcc $(GCCFLAGS) -c list.c*

*sort.o:*

*gcc $(GCCFLAGS) -c sort.c*

*main.o:*

*gcc $(GCCFLAGS) -c main.c*

*clean:*

*rm -rf \*.o main*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ ls*

*Makefile list.c list.h main.c sort.c sort.h*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ make sort.o*

*gcc -w -c sort.c*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ ls*

*Makefile list.c list.h main.c sort.c sort.h sort.o*

*alice@alice-VirtualBox:~$/mnt/c/Users/alice/Desktop/labs/26$ make main.o*

*gcc -w -c main.c*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ ls*

*Makefile list.c list.h main.c main.o sort.c sort.h sort.o*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ make all*

*gcc -w -c list.c*

*gcc -w list.o sort.o main.o -o main*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ ls*

*Makefile list.c list.h list.o main main.c main.o sort.c sort.h sort.o*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ make clean*

*rm -rf \*.o main*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ ls*

*Makefile list.c list.h main.c sort.c sort.h*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ make all*

*gcc -w -c list.c*

*gcc -w -c sort.c*

*gcc -w -c main.c*

*gcc -w list.o sort.o main.o -o main*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ make all*

*gcc -w list.o sort.o main.o -o main*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ ls*

*Makefile list.c list.h list.o main main.c main.o sort.c sort.h sort.o*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ make sort.o*

*make: 'sort.o' is up to date.*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ ls*

*Makefile list.c list.h list.o main main.c main.o sort.c sort.h sort.o*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ make clean*

*rm -rf \*.o main*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ ls*

*Makefile list.c list.h main.c sort.c sort.h*

*alice@alice-VirtualBox:~$c/Users/alice/Desktop/labs/26$ make run*

*gcc -w -c list.c*

*gcc -w -c sort.c*

*gcc -w -c main.c*

*gcc -w list.o sort.o main.o -o main*

./main

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

The list is empty!

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

2

Enter value and key:

1234 567

Enter index to insert:

102342

Cannot insert!

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

2

Enter value and key:

13 1

Enter index to insert:

1

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:13 key:1

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

2

Enter value and key:

235 70

Enter index to insert:

2

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:13 key: 1

2. value:235 key:70

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

2

Enter value and key:

66 66

Enter index to insert:

1

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:66 key:66

2. value:13 key: 1

3. value:235 key:70

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

2

Enter value and key:

1 17

Enter index to insert:

2

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:66 key:66

2. value:1 key:17

3. value:13 key:1

4. value:235 key:70

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

3

Enter index:

4

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:66 key:66

2. value:1 key:17

3. value:13 key:1

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

3

Enter index:

1

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:1 key:17

2. value:13 key:1

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

3

Enter index:

2

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:1 key:17

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

2

Enter value and key:

10 190

Enter index to insert:

1

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

2

Enter value and key:

77 77

Enter index to insert:

3

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:10 key:190

2. value:1 key:17

3. value:77 key:77

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

4

The size is 3

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

2

Enter value and key:

1000 1000

Enter index to insert:

4

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

4

The size is 4

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:10 key:190

2. value:1 key:17

3. value:77 key:77

4. value:1000 key:1000

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

2

Enter value and key:

32 1

Enter index to insert:

3

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:10 key:190

2. value:200 key:300

3. value:32 key:1

4. value:77 key:77

5. value:1000 key:1000

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

4

The size is 5

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

5

The list has been sorted!

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:32 key:1

2. value:77 key:77

3. value:10 key:190

4. value:200 key:300

5. value:1000 key:1000

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

3

Enter index:

3

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:32 key:1

2. value:77 key:77

3. value:200 key:300

4. value:1000 key:1000

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

3

Enter index:

4

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:32 key:1

2. value:77 key:77

3. value:200 key:300

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

3

Enter index:

2

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1

1. value:32 key:1

2. value:200 key:300

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

342342

Incorrect command!

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

4

The size is 2

1 - print list

2 - insert

3 - erase

4 - get size

5 - bubble sort

1. **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 |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 11. **Выводы**

В ходе данной лабораторной работы мною были изучены абстрактные типы данных, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

рекурсия, модульное программирование на языке Си, а также автоматизация сборки программ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

модульной структуры на языке Си с использованием утилиты make\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_