Федеральное государственное бюджетное образовательное учреждение высшего образования «Пензенский Государственный университет»

Кафедра «Вычислительная техника»

ОТЧЕТ

**По лабораторной работе №3**

по курсу «Логика и основы алгоритмизации в ИЗ»

на тему «Динамические списки»

Выполнили: ст. гр. 21ВВ1

Сурков А.А

Хорошильцев Е.А

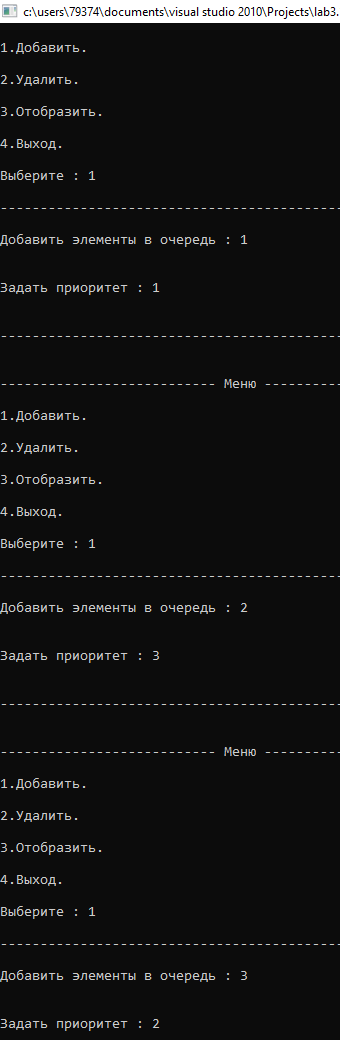
Кривенков И.В

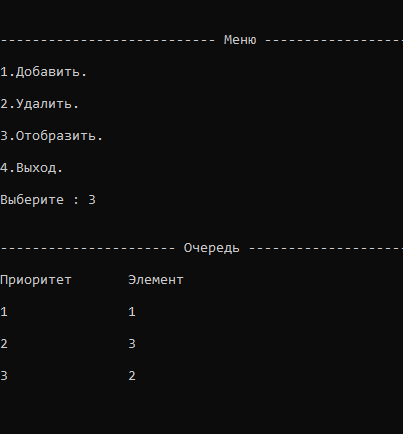
Приняли: Юрова О.В.

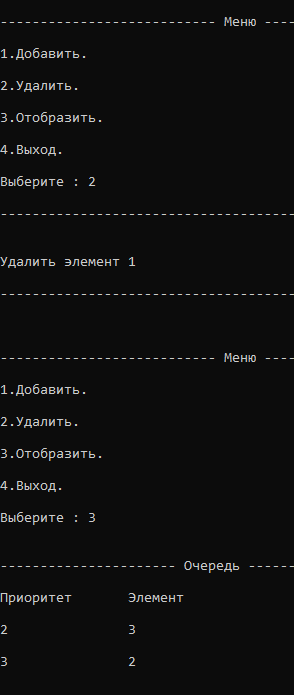
Акифьев И.В.

2022

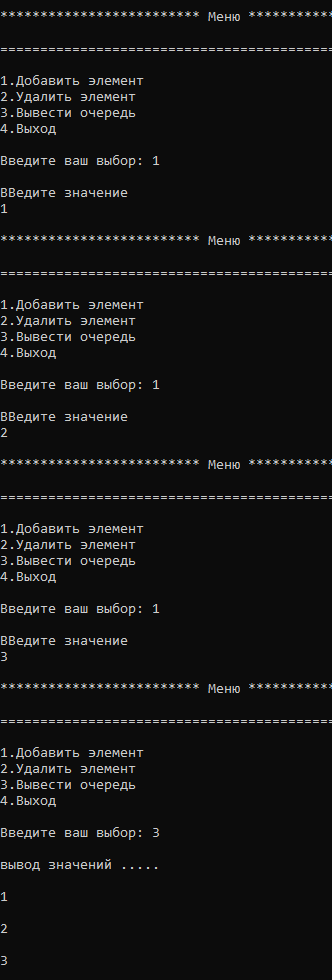
**Задание 1**: Реализовать приоритетную очередь, путём добавления элемента в список в соответствии с приоритетом объекта.

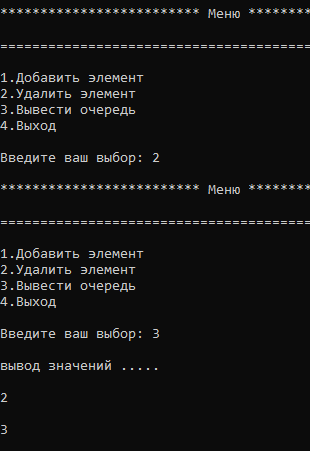






**Задание 2:** Реализовать структуру данных “Очередь”.





**Вывод:** в ходе выполнения данной лабораторной работы мы вспомнили работу со списками, а также научились реализовывать очередь и приоритетную очередь.

**Листинг:**

**Задание 1:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <conio.h>

#include <locale.h>

void insert(); //добавить

void del(); // удалить

void display(); //отобразить

struct node{

int priority;

int info;

struct node \*link;

}\*front = NULL;

int main(){

setlocale(LC\_ALL, "Rus");

int choice;

while(1){

printf("\n\n--------------------------- Меню -------------------------------------\n\n");

printf("1.Добавить.\n\n");

printf("2.Удалить.\n\n");

printf("3.Отобразить.\n\n");

printf("4.Выход.\n\n");

printf("Выберите : ");

scanf("%d", &choice);

switch(choice){

case 1:

insert();

break;

case 2:

del();

break;

case 3:

display();

break;

case 4:

exit(1);

default :

printf("Плохо, пробуй ещё...\n");

}

}

}

void insert(){

setlocale(LC\_ALL, "Rus");

struct node \*tmp,\*q;

int added\_item,item\_priority;

tmp = (struct node \*)malloc(sizeof(struct node));

printf("\n------------------------------------------------------------------------------\n");

printf("\nДобавить элементы в очередь : ");

scanf("%d",&added\_item);

printf("\n\nЗадать приоритет : ");

scanf("%d",&item\_priority);

printf("\n\n------------------------------------------------------------------------------\n");

tmp->info = added\_item;

tmp->priority = item\_priority;

if( front == NULL || item\_priority < front->priority ){

tmp->link = front;

front = tmp;

}

else{

q = front;

while( q->link != NULL && q->link->priority <= item\_priority ) q=q->link;

tmp->link = q->link;

q->link = tmp;

}

}

void del(){

struct node \*tmp;

if(front == NULL){

printf("\n--------------------------------------------------------------------");

printf("\n\nНедостаточный поток очередей\n");

printf("\n------------------------------------------------------------------------");

}

else{

tmp = front;

printf("\n----------------------------------------------------------------------------\n");

printf("\n\nУдалить элемент %d\n",tmp->info);

printf("\n-----------------------------------------------------------------------------\n\n");

front = front->link;

free(tmp);

}

}

void display(){

setlocale(LC\_ALL, "Rus");

struct node \*ptr;

ptr = front;

if(front == NULL){

printf("\n-----------------------------------------------------------------------");

printf("\n\nОчередь пуста\n");

printf("\n--------------------------------------------------------------------------");

}

else{

printf("\n\n---------------------- Очередь --------------------------------\n\n");

printf("Приоритет\tЭлемент\n\n");

while(ptr != NULL){

printf("%d \t\t%d\n\n",ptr->priority,ptr->info);

ptr = ptr->link;

}

}

}

**Задание 2:**

#include<stdio.h>

#include<stdlib.h>

#include <locale.h>

struct node

{

int data;

struct node \*next;

};

void insert();

void del();

void display();

struct node \*front;

struct node \*rear;

void main () {

setlocale(LC\_ALL, "Rus");

int choice = 1;

while(choice != 4)

{

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Меню \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n=================================================================\n");

printf("\n1.Добавить элемент\n2.Удалить элемент\n3.Вывести очередь\n4.Выход\n");

printf("\nВведите ваш выбор: ");

scanf("%d",& choice);

switch(choice)

{

case 1:

insert();

break;

case 2:

del();

break;

case 3:

display();

break;

case 4:

exit(0);

break;

default:

printf("\nПлохо, пробуй ещё...\n");

}

}

}

void insert() {

setlocale(LC\_ALL, "Rus");

struct node \*ptr;

int item;

ptr = (struct node \*) malloc (sizeof(struct node));

if(ptr == NULL) {

printf("\nПереполнение\n");

return;

}

else {

printf("\nВВедите значение\n");

scanf("%d",&item);

ptr -> data = item;

if(front == NULL) {

front = ptr;

rear = ptr;

front -> next = NULL;

rear -> next = NULL;

}

else {

rear -> next = ptr;

rear = ptr;

rear->next = NULL;

}

}

}

void del () {

setlocale(LC\_ALL, "Rus");

struct node \*ptr;

if(front == NULL)

{

printf("\nНЕДОСТАТОЧНЫЙ ПОТОК\n");

return;

}

else

{

ptr = front;

front = front -> next;

free(ptr);

}

}

void display()

{

setlocale(LC\_ALL, "Rus");

struct node \*ptr;

ptr = front;

if(front == NULL)

{

printf("\nОчередь пуста\n");

}

else

{ printf("\nвывод значений .....\n");

while(ptr != NULL)

{

printf("\n%d\n",ptr -> data);

ptr = ptr -> next;

}

}

}