Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа № 6

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамические структуры данных. Односвязные списки»

Выполнил:

Кулешов Артём

Студент 1 курса 8 группы

Преподаватель: асс. Андронова М.В.

Минск, 2024

4. В соответствии со своим вариантом разработать программу с использованием ***односвязного списка*** по данным, представленным в таблице ниже.

Программа должна содержать меню с пунктами: добавление элемента, удаление элемента, поиск элемента, вывод списка в консольное окно, запись списка в файл, считывание списка из файла.

Вариант №5

Создать список, содержащий элементы целого типа. Найти сумму положительных двухзначных элементов или выдать сообщение, что таких элементов нет.

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct list {

int num;

list \*next;

};

list\* arrlist = nullptr; // указатель на начало списка

list\* arr; // указатель на текущий элемент

void addElement();

void deleteElement();

void viewAll();

void sum\_poz\_el();

void write\_to\_file();

void read\_from\_file();

int main()

{

int choice;

do {

cout << "Menu " << endl;

cout << "1. Add element to list." << endl;

cout << "2. Delete element." << endl;

cout << "3. View all elements." << endl;

cout << "4. Sum of positive two-digit elements." << endl;

cout << "5. Write list to file." << endl;

cout << "6. Read list from file." << endl;

cout << "7. Exit." << endl;

cin >> choice;

switch (choice) {

case 1:

addElement();

break;

case 2:

deleteElement();

break;

case 3:

viewAll();

break;

case 4:

sum\_poz\_el();

break;

case 5:

write\_to\_file();

break;

case 6:

read\_from\_file();

break;

case 7:

cout << "Exit." << endl;

break;

default:

cout << "Error." << endl;

break;

}

} while (choice != 7);

return 0;

}

//Функция для ввода элемента

void addElement()

{

arr = new list; //Выделение памяти

arr->next = arrlist;

arrlist = arr;

cout << "Enter number:";

cin >> arr->num;

}

//Функция для удаления элемента

void deleteElement()

{

int value;

cin >> value;

list\* temp = arrlist, \* temparrList = nullptr;

arr = arrlist;

while (arr)

{

if (arr->num != value)

{

temp = new list;

temp->next = temparrList;

temparrList = temp;

temp->num = arr->num;

}

arr = arr->next;

}

arrlist = temparrList;

while (temp)

{

arr = temp;

temp = temp->next;

}

delete temp;

}

//Функция для отображения всех элементов списка

void viewAll()

{

arr = arrlist;

while (arr)

{

cout << arr-> num << ' ';

arr = arr-> next;

}

cout << endl;

}

void sum\_poz\_el()

{

arr = arrlist;

int sum = 0;

while (arr)

{

if (arr->num >= 10 && arr->num <= 99)

sum += arr->num;

arr = arr->next;

}

if (sum == 0)

cout << "Element search error." << endl;

else cout << "Sum = " << sum << endl;

}

//Функция для записи в файл

void write\_to\_file()

{

ofstream file("file.txt");

arr = arrlist;

while (arr)

{

file << arr->num;

arr = arr->next;

file << ' ';

}

file << endl << endl;

file.close();

}

//Функция для чтения из файла

void read\_from\_file()

{

ifstream file("file.txt");

string a;

cout << endl;

getline(file, a);

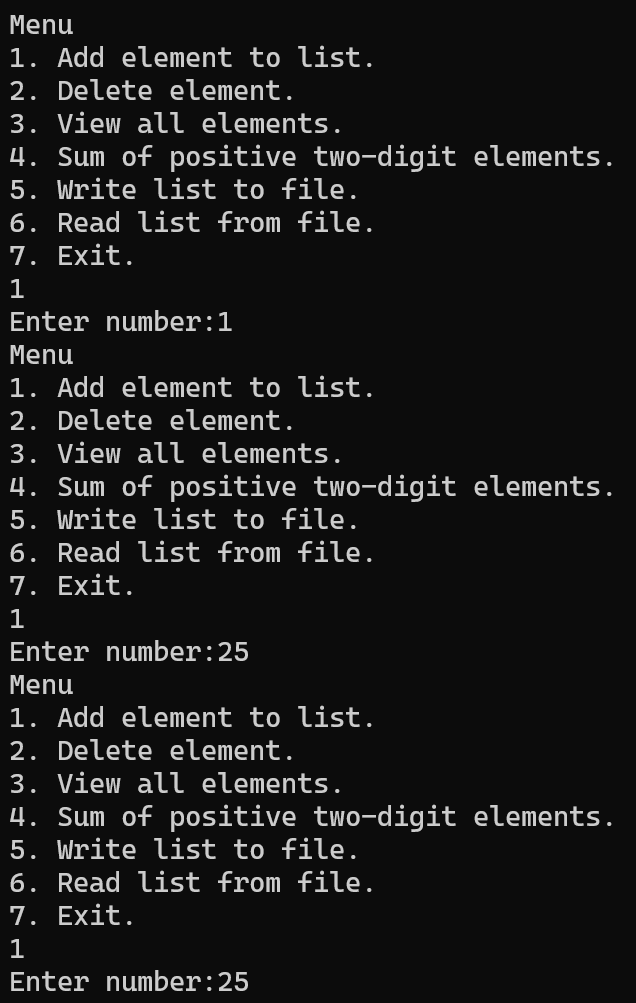
cout << endl << a;

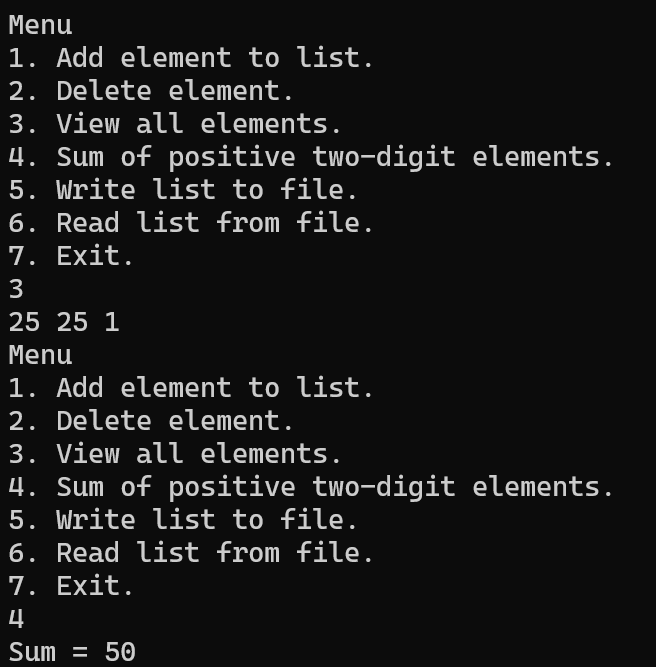
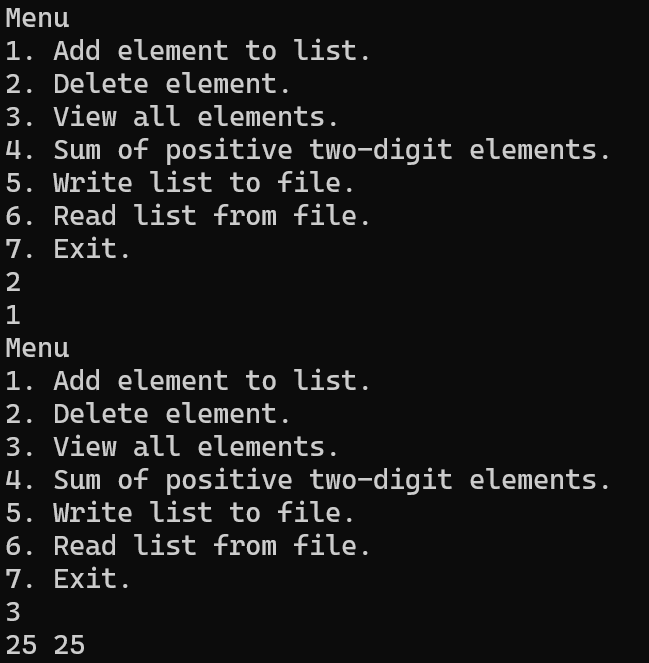
getline(file, a);

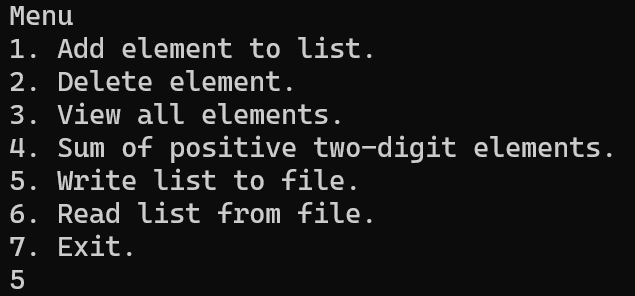
cout << endl << a << endl;

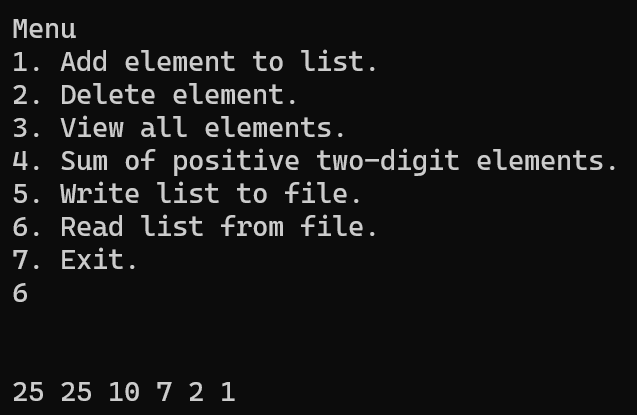
file.close();

}







Вариант №1

Создать список, содержащий элементы целого типа. Найти сумму положительных элементов или выдать сообщение, что положительных элементов нет.

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct list {

int num;

list \*next;

};

list\* arrlist = nullptr; // указатель на начало списка

list\* arr; // указатель на текущий элемент

void addElement();

void deleteElement();

void viewAll();

void poz();

void write\_to\_file();

void read\_from\_file();

int main()

{

int choice;

do {

cout << "Menu " << endl;

cout << "1. Add element to list." << endl;

cout << "2. Delete element." << endl;

cout << "3. View all elements." << endl;

cout << "4. Sum of positive elements." << endl;

cout << "5. Write list to file." << endl;

cout << "6. Read list from file." << endl;

cout << "7. Exit." << endl;

cin >> choice;

switch (choice) {

case 1:

addElement();

break;

case 2:

deleteElement();

break;

case 3:

viewAll();

break;

case 4:

poz();

break;

case 5:

write\_to\_file();

break;

case 6:

read\_from\_file();

break;

case 7:

cout << "Exit." << endl;

break;

default:

cout << "Error." << endl;

break;

}

} while (choice != 7);

return 0;

}

//Функция для ввода элемента

void addElement()

{

arr = new list; //Выделение памяти

arr->next = arrlist;

arrlist = arr;

cout << "Enter number:";

cin >> arr->num;

}

//Функция для удаления элемента

void deleteElement()

{

int value;

cin >> value;

list\* temp = arrlist, \* temparrList = nullptr;

arr = arrlist;

while (arr)

{

if (arr->num != value)

{

temp = new list;

temp->next = temparrList;

temparrList = temp;

temp->num = arr->num;

}

arr = arr->next;

}

arrlist = temparrList;

while (temp)

{

arr = temp;

temp = temp->next;

}

delete temp;

}

//Функция для отображения всех элементов списка

void viewAll()

{

arr = arrlist;

while (arr)

{

cout << arr-> num << ' ';

arr = arr-> next;

}

cout << endl;

}

void poz()

{

arr = arrlist;

int sum = 0;

while (arr)

{

if (arr->num > 0)

sum += arr->num;

arr = arr->next;

}

if (sum == 0)

cout << "Element search error." << endl;

else cout << "Sum = " << sum << endl;

}

//Функция для записи в файл

void write\_to\_file()

{

ofstream file("file.txt");

arr = arrlist;

while (arr)

{

file << arr->num;

arr = arr->next;

file << ' ';

}

file << endl << endl;

file.close();

}

//Функция для чтения из файла

void read\_from\_file()

{

ifstream file("file.txt");

string a;

cout << endl;

getline(file, a);

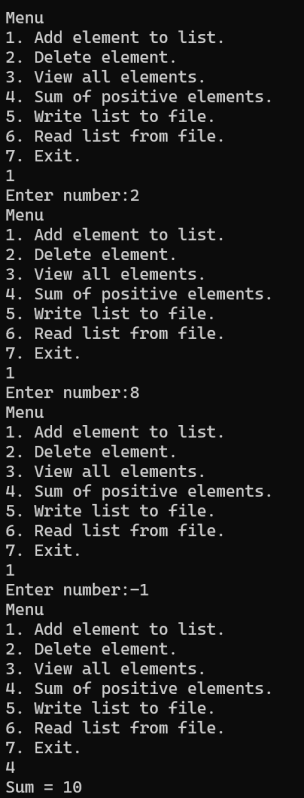
cout << endl << a;

getline(file, a);

cout << endl << a << endl;

file.close();

}



Вариант №6

Создать список, содержащий элементы целого типа. Найти сумму отрицательных двухзначных элементов или выдать сообщение, что таких элементов нет.

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct list {

int num;

list \*next;

};

list\* arrlist = nullptr; // указатель на начало списка

list\* arr; // указатель на текущий элемент

void addElement();

void deleteElement();

void viewAll();

void sum\_negative\_two\_digit();

void write\_to\_file();

void read\_from\_file();

int main()

{

int choice;

do {

cout << "Menu " << endl;

cout << "1. Add element to list." << endl;

cout << "2. Delete element." << endl;

cout << "3. View all elements." << endl;

cout << "4. Sum of negative two-digit elements." << endl;

cout << "5. Write list to file." << endl;

cout << "6. Read list from file." << endl;

cout << "7. Exit." << endl;

cin >> choice;

switch (choice) {

case 1:

addElement();

break;

case 2:

deleteElement();

break;

case 3:

viewAll();

break;

case 4:

sum\_negative\_two\_digit();

break;

case 5:

write\_to\_file();

break;

case 6:

read\_from\_file();

break;

case 7:

cout << "Exit." << endl;

break;

default:

cout << "Error." << endl;

break;

}

} while (choice != 7);

return 0;

}

//Функция для ввода элемента

void addElement()

{

arr = new list; //Выделение памяти

arr->next = arrlist;

arrlist = arr;

cout << "Enter number:";

cin >> arr->num;

}

//Функция для удаления элемента

void deleteElement()

{

int value;

cin >> value;

list\* temp = arrlist, \* temparrList = nullptr;

arr = arrlist;

while (arr)

{

if (arr->num != value)

{

temp = new list;

temp->next = temparrList;

temparrList = temp;

temp->num = arr->num;

}

arr = arr->next;

}

arrlist = temparrList;

while (temp)

{

arr = temp;

temp = temp->next;

}

delete temp;

}

//Функция для отображения всех элементов списка

void viewAll()

{

arr = arrlist;

while (arr)

{

cout << arr-> num << ' ';

arr = arr-> next;

}

cout << endl;

}

void sum\_negative\_two\_digit()

{

arr = arrlist;

int sum = 0;

while (arr)

{

if (arr->num >= -99 && arr->num <= -10)

sum += arr->num;

arr = arr->next;

}

if (sum == 0)

cout << "Element search error." << endl;

else cout << "Sum = " << sum << endl;

}

//Функция для записи в файл

void write\_to\_file()

{

ofstream file("file.txt");

arr = arrlist;

while (arr)

{

file << arr->num;

arr = arr->next;

file << ' ';

}

file << endl << endl;

file.close();

}

//Функция для чтения из файла

void read\_from\_file()

{

ifstream file("file.txt");

string a;

cout << endl;

getline(file, a);

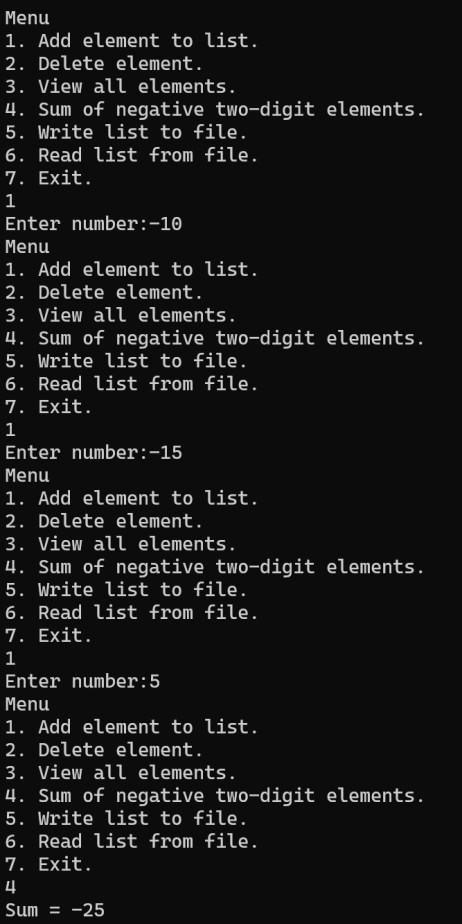
cout << endl << a;

getline(file, a);

cout << endl << a << endl;

file.close();

}



Вариант № 14

Создать список, содержащий элементы целого типа. Найти сумму отрицательных элементов, кратных 5, или выдать сообщение, что таких элементов нет.

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct list {

int num;

list \*next;

};

list\* arrlist = nullptr; // указатель на начало списка

list\* arr; // указатель на текущий элемент

void addElement();

void deleteElement();

void viewAll();

void sum\_negative\_5();

void write\_to\_file();

void read\_from\_file();

int main()

{

int choice;

do {

cout << "Menu " << endl;

cout << "1. Add element to list." << endl;

cout << "2. Delete element." << endl;

cout << "3. View all elements." << endl;

cout << "4. Sum of negative elements divisible by 5." << endl;

cout << "5. Write list to file." << endl;

cout << "6. Read list from file." << endl;

cout << "7. Exit." << endl;

cin >> choice;

switch (choice) {

case 1:

addElement();

break;

case 2:

deleteElement();

break;

case 3:

viewAll();

break;

case 4:

sum\_negative\_5();

break;

case 5:

write\_to\_file();

break;

case 6:

read\_from\_file();

break;

case 7:

cout << "Exit." << endl;

break;

default:

cout << "Error." << endl;

break;

}

} while (choice != 7);

return 0;

}

//Функция для ввода элемента

void addElement()

{

arr = new list; //Выделение памяти

arr->next = arrlist;

arrlist = arr;

cout << "Enter number:";

cin >> arr->num;

}

//Функция для удаления элемента

void deleteElement()

{

int value;

cin >> value;

list\* temp = arrlist, \* temparrList = nullptr;

arr = arrlist;

while (arr)

{

if (arr->num != value)

{

temp = new list;

temp->next = temparrList;

temparrList = temp;

temp->num = arr->num;

}

arr = arr->next;

}

arrlist = temparrList;

while (temp)

{

arr = temp;

temp = temp->next;

}

delete temp;

}

//Функция для отображения всех элементов списка

void viewAll()

{

arr = arrlist;

while (arr)

{

cout << arr-> num << ' ';

arr = arr-> next;

}

cout << endl;

}

void sum\_negative\_5()

{

arr = arrlist;

int sum = 0;

while (arr)

{

if (arr->num % 5 == 0 && arr->num < 0)

sum += arr->num;

arr = arr->next;

}

if (sum == 0)

cout << "Element search error." << endl;

else cout << "Sum = " << sum << endl;

}

//Функция для записи в файл

void write\_to\_file()

{

ofstream file("file.txt");

arr = arrlist;

while (arr)

{

file << arr->num;

arr = arr->next;

file << ' ';

}

file << endl << endl;

file.close();

}

//Функция для чтения из файла

void read\_from\_file()

{

ifstream file("file.txt");

string a;

cout << endl;

getline(file, a);

cout << endl << a;

getline(file, a);

cout << endl << a << endl;

file.close();

}

