

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский
политехнический университет»**

Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
направление подготовки: 09.03.01– «Информатика и вычислительная техника»

**Лабораторная работа № 16
по дисциплине
«Информатика»
на тему
«Поиски»**

Выполнил студент гр. ИВТ-23-16

Южаков Федор Алексеевич

Проверил:

доцент кафедры ИТАС

Денис Владимирович Яруллин

(оценка)

(подпись)

(дата)

г. Пермь, 2024

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1 Вариант задания

Постановка задачи.

В массиве целых чисел найти любой элемент с заданным значением.

Заполнение массива организовать любым удобным способом. При этом учесть, что не все поиски работают с неотсортированными массивами.

Поиск выполнить тремя видами:

1. Линейный;
2. Бинарный;
3. Интерполяционный.

Для удобства создать меню с выбором типа поиска.

2 Анализ задачи

В данной задаче основная часть программы находится в функции меню, которая вызывается из основного тела программы. В функцию меню передается массив и его размер. Сама функция состоит из текстового меню, запроса от пользователя номера необходимого пункта меню и оператора множественного выбора, где цифры от одного до трёх – различные виды поисков, а 0 – выход из программы, default – проверка на дурака, после которой снова вызывается меню. В каждом кейсе получаем от пользователя способ заполнения массива вручную или рандомно, границы случайных чисел если пользователь выбрал рандом, значения всех элементов массива если вручную. Если поиск не работает с неотсортированным массивом, то сортируем массив сортировкой Шелла. После заполнения и сортировки (если нужно) выводим массив. Спрашиваем пользователя, по какому значению ему нужно найти массив. Затем вызываем необходимый поиск и выводим индекс, по которому был найден элемент, если элемента нет в массиве выводим соответствующую надпись. После работы снова вызываем меню, чтобы пользователь мог запустить другие поиски или же тот же самый, но с другими данными.

Алгоритм сортировки Шелла:

1. Выбрать размер окна
2. Разделить массив на несколько меньших частей, каждая должна нацело делиться на размер окна
3. Отсортировать каждую из частей при помощи простой сортировки вставкой
4. Продолжать с шага 1 до тех пор, пока не отсортируется весь массив

Алгоритмы поисков

Линейный поиск работает очень просто, он идет с начала массива и до тех пор, пока не встретит элемент с нужным значением, после этого он выдает индекс, на котором был найден элемент. Если элемента не было в массиве, будет выдана соответствующая надпись.

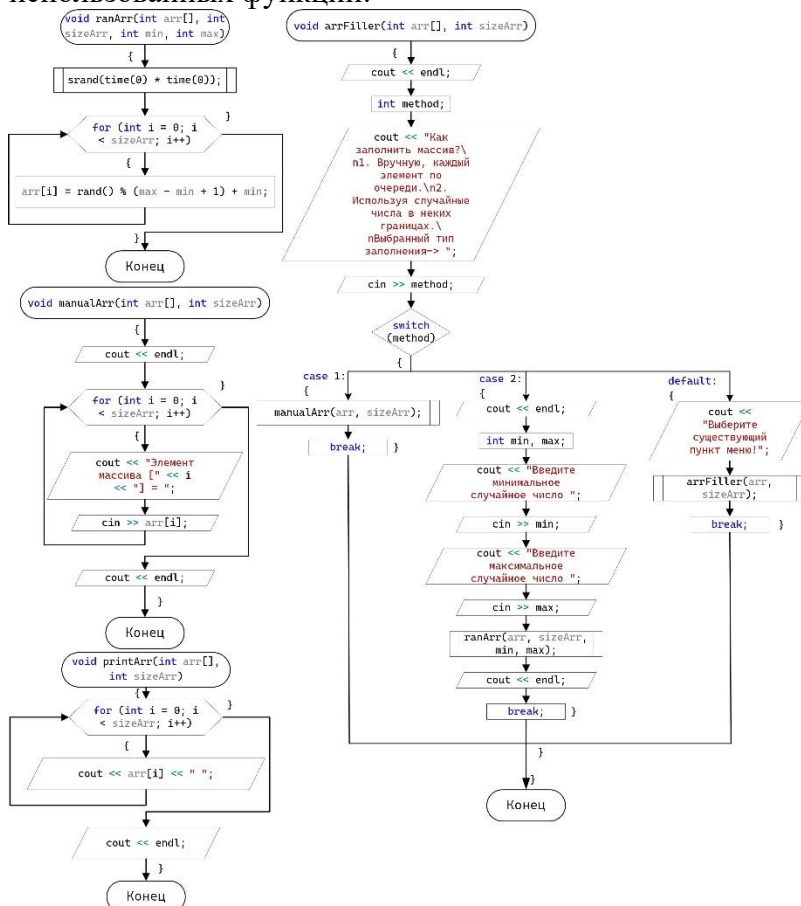
Бинарный поиск работает только с отсортированным массивом. Он ищет середину массива, и смотрит, равен ли этот элемент искомому, если нет, то проверяет, в левой или правой части находится нужный элемент, после этого границы массива становятся границами нужной половины, далее в этих границах снова ищется середина, если элемент снова не найден, снова берется нужная половина, и т.д. пока не найдем элемент. Если же элемент не найден и правая граница равна левой, значит в массиве нет элемента с таким значением и нужно выдать соответствующую надпись.

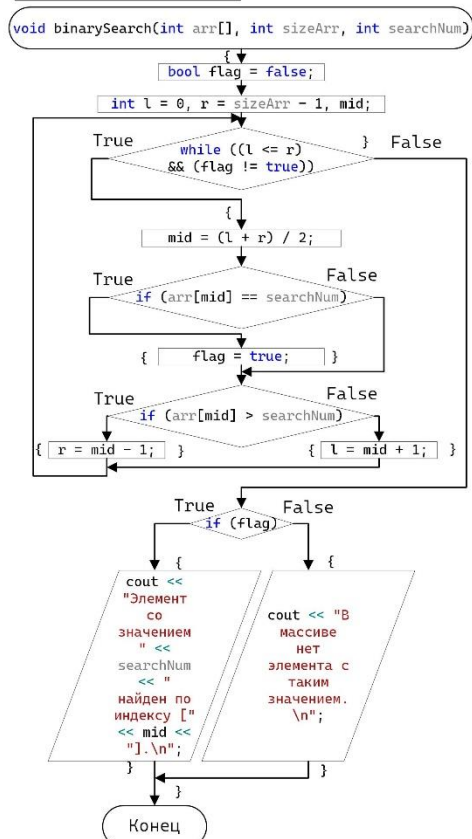
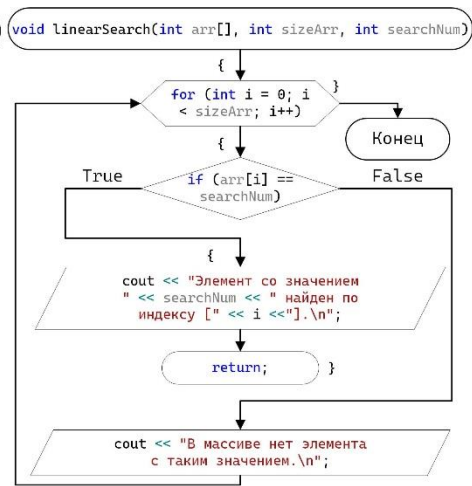
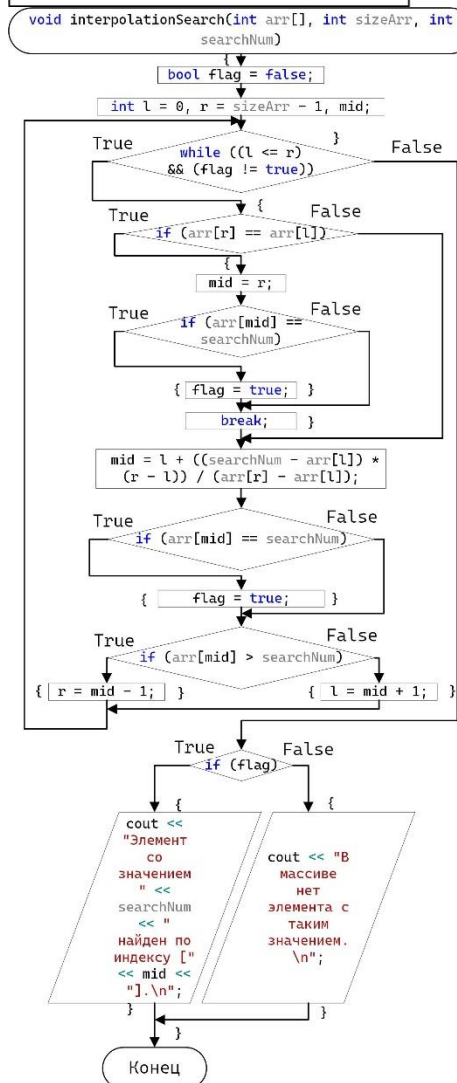
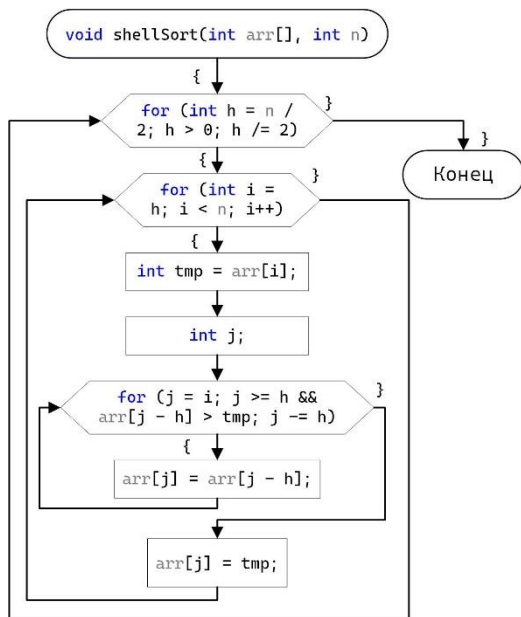
Интерполяционный поиск работает точно также, как и бинарный, но для вычисления середины используется формула: $mid = l + ((searchNum - arr[l]) * (r - l)) / (arr[r] - arr[l])$, где l и r – правая и левая граница, $searchNum$ – значение искомого элемента, arr – массив, в котором производится поиск.

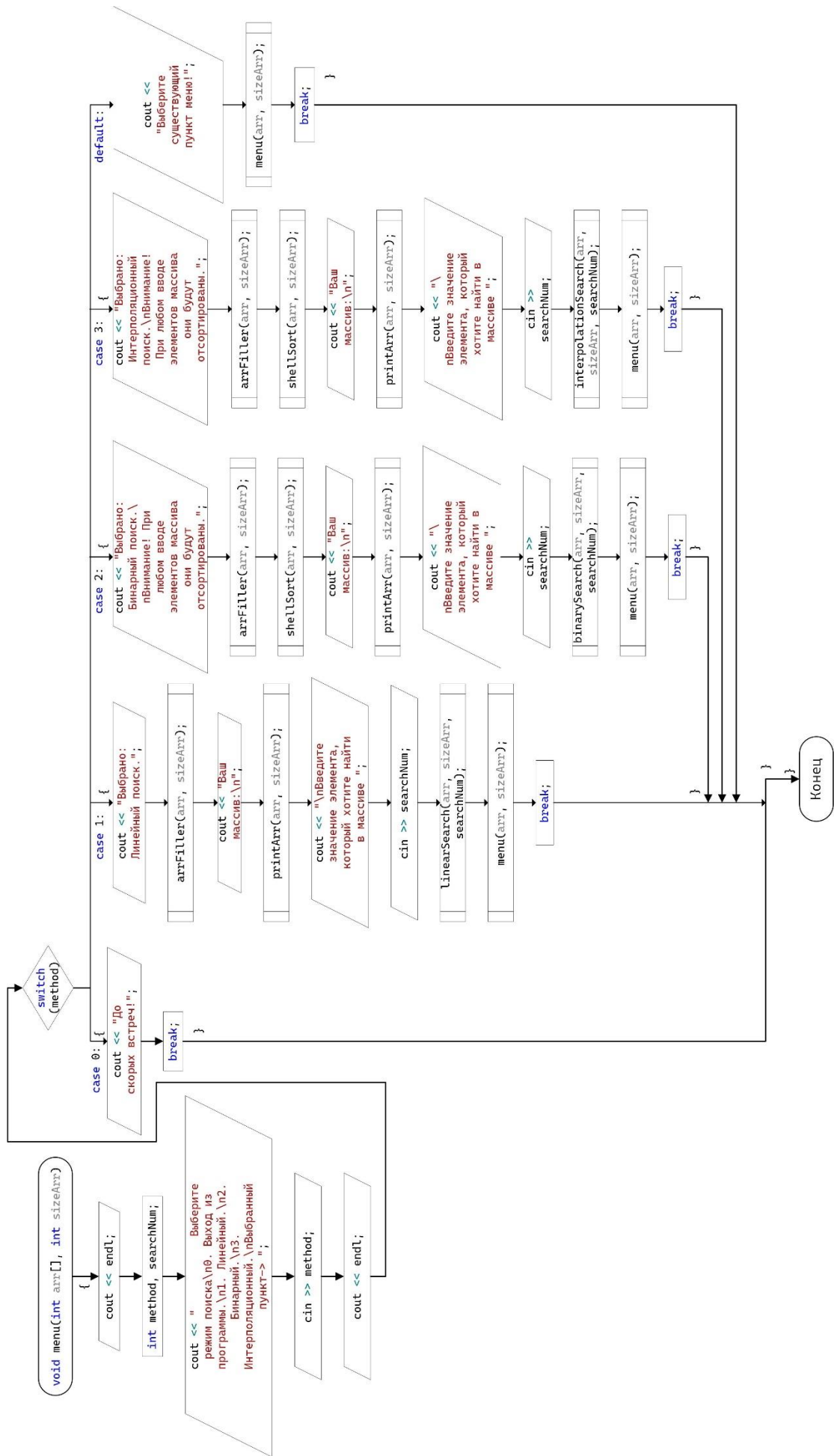
После завершения функции «меню» программа завершается.

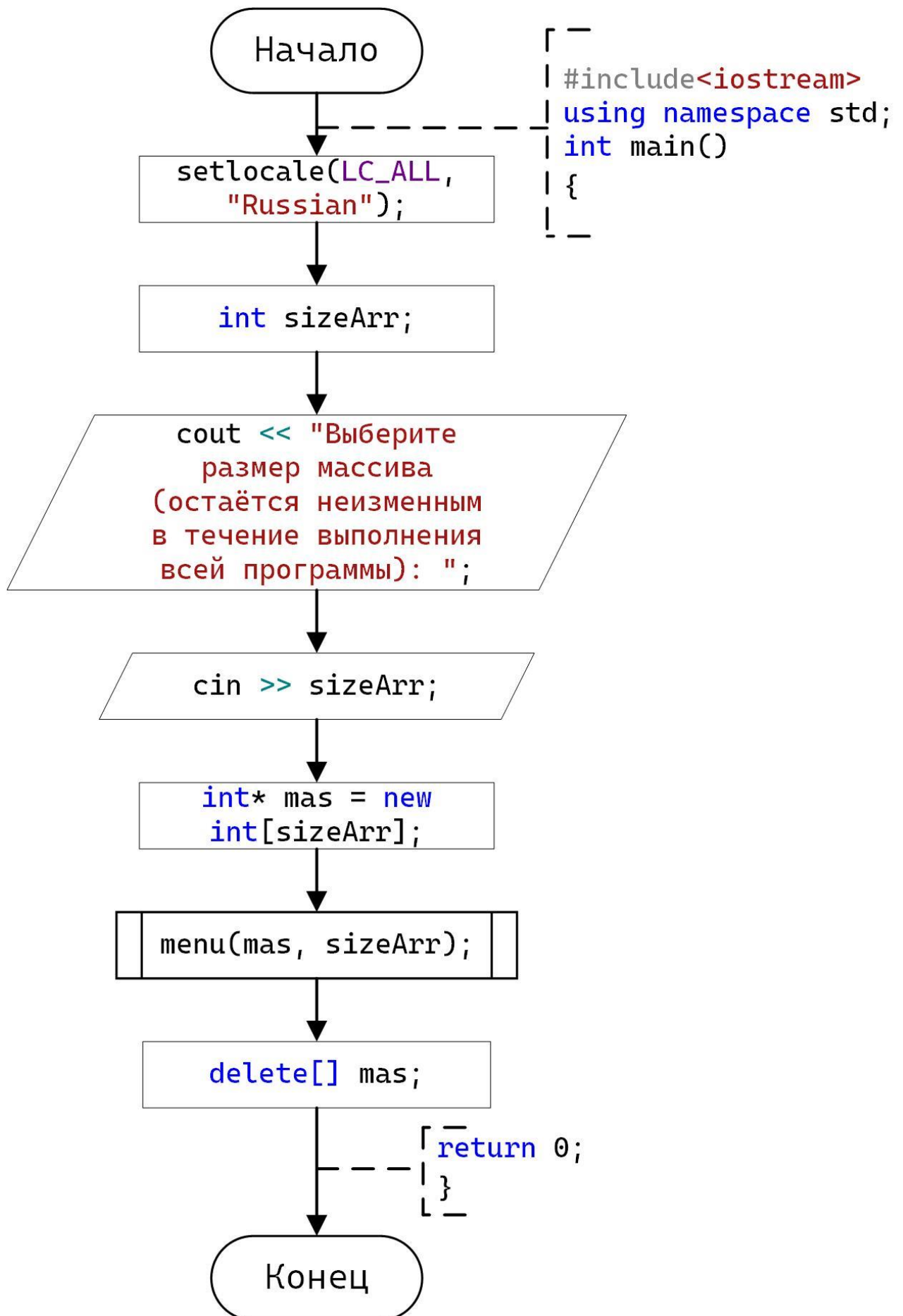
3 Блок схема

Проанализировав задачу, составим подробную блок схему основной программы и использованных функций.









ПРАКТИЧЕСКАЯ ЧАСТЬ

4 Результат решения

4.1 Готовая программа

Исходя из подробных блок схем, составим программу на языке C++.

Таблица 1 – Готовая программа задачи

```
#include <iostream>
using namespace std;
void ranArr(int arr[], int sizeArr, int min, int max)
{
    srand(time(0) * time(0));
    for (int i = 0; i < sizeArr; i++) arr[i] = rand() % (max - min + 1) + min;
}
void manualArr(int arr[], int sizeArr)
{
    cout << endl;
    for (int i = 0; i < sizeArr; i++)
    {
        cout << "Элемент массива [" << i << "] = ";
        cin >> arr[i];
    }
    cout << endl;
}
void arrFiller(int arr[], int sizeArr)
{
    cout << endl;
    int method;
    cout << "Как заполнить массив?\n1. Вручную, каждый элемент по очереди.\n2. Используя
случайные числа в неких границах.\nВыбранный тип заполнения-> ";
    cin >> method;
    switch (method)
    {
        case 1:
        {
            manualArr(arr, sizeArr);
            break;
        }
        case 2:
        {
            cout << endl;
            int min, max;
            cout << "Введите минимальное случайное число ";
            cin >> min;
            cout << "Введите максимальное случайное число ";
            cin >> max;
            ranArr(arr, sizeArr, min, max);
            cout << endl;
            break;
        }
        default:
        {
            cout << "Выберите существующий пункт меню!";
            arrFiller(arr, sizeArr);
            break;
        }
    }
}
void printArr(int arr[], int sizeArr)
{
    for (int i = 0; i < sizeArr; i++) cout << arr[i] << " ";
    cout << endl;
}
void shellSort(int arr[], int n)
{
    for (int h = n / 2; h > 0; h /= 2)
    {
        for (int i = h; i < n; i++)
        {
```

```

        int tmp = arr[i];
        int j;
        for (j = i; j >= h && arr[j - h] > tmp; j -= h)
        {
            arr[j] = arr[j - h];
        }
        arr[j] = tmp;
    }
}

void linearSearch(int arr[], int sizeArr, int searchNum)
{
    for (int i = 0; i < sizeArr; i++)
    {
        if (arr[i] == searchNum)
        {
            cout << "Элемент со значением " << searchNum << " найден по индексу [" << i
<< "].\n";
            return;
        }
    }
    cout << "В массиве нет элемента с таким значением.\n";
}

void binarySearch(int arr[], int sizeArr, int searchNum)
{
    bool flag = false;
    int l = 0, r = sizeArr - 1, mid;
    while ((l <= r) && (flag != true))
    {
        mid = (l + r) / 2;
        if (arr[mid] == searchNum) flag = true;
        if (arr[mid] > searchNum) r = mid - 1;
        else l = mid + 1;
    }
    if (flag) cout << "Элемент со значением " << searchNum << " найден по индексу [" << mid <<
    "].\n";
    else cout << "В массиве нет элемента с таким значением.\n";
}

void interpolationSearch(int arr[], int sizeArr, int searchNum)
{
    bool flag = false;
    int l = 0, r = sizeArr - 1, mid;
    while ((l <= r) && (flag != true))
    {
        if (arr[r] == arr[l])
        {
            mid = r;
            if (arr[mid] == searchNum) flag = true;
            break;
        }
        mid = l + ((searchNum - arr[l]) * (r - l)) / (arr[r] - arr[l]);
        if (arr[mid] == searchNum) flag = true;
        if (arr[mid] > searchNum)
        {
            r = mid - 1;
        }
        else
        {
            l = mid + 1;
        }
    }
    if (flag) cout << "Элемент со значением " << searchNum << " найден по индексу [" << mid <<
    "].\n";
    else cout << "В массиве нет элемента с таким значением.\n";
}

void menu(int arr[], int sizeArr)
{
    cout << endl;
    int method, searchNum;
    cout << "                Выберите режим поиска\n0. Выход из программы.\n1. Линейный.\n2.
Бинарный.\n3. Интерполяционный.\nВыбранный пункт-> ";
    cin >> method;
    cout << endl;

```



```

switch (method)
{
case 0:
{
    cout << "До скорых встреч!";
    break;
}
case 1:
{
    cout << "Выбрано: Линейный поиск.";
    arrFiller(arr, sizeArr);
    cout << "Ваш массив:\n";
    printArr(arr, sizeArr);
    cout << "\nВведите значение элемента, который хотите найти в массиве ";
    cin >> searchNum;
    linearSearch(arr, sizeArr, searchNum);
    menu(arr, sizeArr);
    break;
}
case 2:
{
    cout << "Выбрано: Бинарный поиск.\nВнимание! При любом вводе элементов массива они
будут отсортированы.";
    arrFiller(arr, sizeArr);
    shellSort(arr, sizeArr);
    cout << "Ваш массив:\n";
    printArr(arr, sizeArr);
    cout << "\nВведите значение элемента, который хотите найти в массиве ";
    cin >> searchNum;
    binarySearch(arr, sizeArr, searchNum);
    menu(arr, sizeArr);
    break;
}
case 3:
{
    cout << "Выбрано: Интерполяционный поиск.\nВнимание! При любом вводе элементов массива
они будут отсортированы.";
    arrFiller(arr, sizeArr);
    shellSort(arr, sizeArr);
    cout << "Ваш массив:\n";
    printArr(arr, sizeArr);
    cout << "\nВведите значение элемента, который хотите найти в массиве ";
    cin >> searchNum;
    interpolationSearch(arr, sizeArr, searchNum);
    menu(arr, sizeArr);
    break;
}
default:
{
    cout << "Выберите существующий пункт меню!";
    menu(arr, sizeArr);
    break;
}
}
}

int main()
{
    setlocale(LC_ALL, "Russian");
    int sizeArr;
    cout << "Выберите размер массива (остаётся неизменным в течение выполнения всей
программы): ";
    cin >> sizeArr;
    int* mas = new int[sizeArr];
    menu(mas, sizeArr);
    delete[] mas;
    return 0;
}

```

4.2 Скриншоты

Запустим программу, чтобы убедиться, что она работает верно.

```
Консоль отладки Microsoft V x + v
Выберите размер массива (остаётся неизменным в течение выполнения всей программы): 25

    Выберите режим поиска
0. Выход из программы.
1. Линейный.
2. Бинарный.
3. Интерполяционный.
Выбранный пункт-> 44

Выберите существующий пункт меню!
    Выберите режим поиска
0. Выход из программы.
1. Линейный.
2. Бинарный.
3. Интерполяционный.
Выбранный пункт-> 2

Выбрано: Бинарный поиск.
Внимание! При любом вводе элементов массива они будут отсортированы.
Как заполнить массив?
1. Вручную, каждый элемент по очереди.
2. Используя случайные числа в неких границах.
Выбранный тип заполнения-> 1

Элемент массива [0] = 234
Элемент массива [1] = 876
Элемент массива [2] = 12
Элемент массива [3] = 987
Элемент массива [4] = 23
Элемент массива [5] = 12
Элемент массива [6] = 0
Элемент массива [7] = -76
Элемент массива [8] = -65
Элемент массива [9] = 2
Элемент массива [10] = 6
Элемент массива [11] = 876543
Элемент массива [12] = 12
Элемент массива [13] = 5
Элемент массива [14] = 976
Элемент массива [15] = -32
Элемент массива [16] = 6

Элемент массива [17] = 88
Элемент массива [18] = 6
Элемент массива [19] = 11
Элемент массива [20] = 12
Элемент массива [21] = 0
Элемент массива [22] = 6
Элемент массива [23] = 6
Элемент массива [24] = 6

Ваш массив:
-76 -65 -32 0 0 2 5 6 6 6 6 6 11 12 12 12 12 23 88 234 876 976 987 876543

Введите значение элемента, который хотите найти в массиве 88
Элемент со значением 88 найден по индексу [19].

    Выберите режим поиска
0. Выход из программы.
1. Линейный.
2. Бинарный.
3. Интерполяционный.
Выбранный пункт-> 1

Выбрано: Линейный поиск.
Как заполнить массив?
1. Вручную, каждый элемент по очереди.
2. Используя случайные числа в неких границах.
Выбранный тип заполнения-> 1

Элемент массива [0] = 1
Элемент массива [1] = 3
Элемент массива [2] = 9
Элемент массива [3] = -54
Элемент массива [4] = -3
Элемент массива [5] = 1
Элемент массива [6] = 0
Элемент массива [7] = 6
Элемент массива [8] = 3
Элемент массива [9] = 1
Элемент массива [10] = -87
Элемент массива [11] = 3
Элемент массива [12] = 2
```

```
Консоль отладки Microsoft V X + v
Элемент массива [13] = 99
Элемент массива [14] = 11
Элемент массива [15] = 11
Элемент массива [16] = 7
Элемент массива [17] = -3
Элемент массива [18] = 0
Элемент массива [19] = 0
Элемент массива [20] = 1
Элемент массива [21] = 2
Элемент массива [22] = 2
Элемент массива [23] = 2
Элемент массива [24] = 2

Ваш массив:
1 3 9 -54 -3 1 0 6 3 1 -87 3 2 99 11 11 7 -3 0 0 1 2 2 2 2

Введите значение элемента, который хотите найти в массиве -3
Элемент со значением -3 найден по индексу [4].

        Выберите режим поиска
0. Выход из программы.
1. Линейный.
2. Бинарный.
3. Интерполяционный.
Выбранный пункт-> 3

Выбрано: Интерполяционный поиск.
Внимание! При любом вводе элементов массива они будут отсортированы.
Как заполнить массив?
1. Вручную, каждый элемент по очереди.
2. Используя случайные числа в неких границах.
Выбранный тип заполнения-> 2

Введите минимальное случайное число -43
Введите максимальное случайное число 50

Ваш массив:
-43 -42 -36 -30 -21 -21 -20 -20 -10 -10 -9 -9 -5 -3 -1 5 9 19 32 34 34 35 43 43 44

Введите значение элемента, который хотите найти в массиве -10
Консоль отладки Microsoft V X + v
Элемент со значением -10 найден по индексу [9].

        Выберите режим поиска
0. Выход из программы.
1. Линейный.
2. Бинарный.
3. Интерполяционный.
Выбранный пункт-> 1

Выбрано: Линейный поиск.
Как заполнить массив?
1. Вручную, каждый элемент по очереди.
2. Используя случайные числа в неких границах.
Выбранный тип заполнения-> 2

Введите минимальное случайное число 8
Введите максимальное случайное число 56

Ваш массив:
19 47 10 23 18 27 12 45 12 49 9 40 14 23 56 33 17 31 17 20 56 31 45 51 37

Введите значение элемента, который хотите найти в массиве 40
Элемент со значением 40 найден по индексу [11].

        Выберите режим поиска
0. Выход из программы.
1. Линейный.
2. Бинарный.
3. Интерполяционный.
Выбранный пункт-> 2

Выбрано: Бинарный поиск.
Внимание! При любом вводе элементов массива они будут отсортированы.
Как заполнить массив?
1. Вручную, каждый элемент по очереди.
2. Используя случайные числа в неких границах.
Выбранный тип заполнения-> 2

Введите минимальное случайное число -98
Введите максимальное случайное число -7
```

```
Консоль отладки Microsoft V x + v

Ваш массив:
-97 -89 -83 -79 -76 -72 -70 -69 -65 -64 -52 -52 -41 -40 -40 -36 -34 -34 -33 -33 -31 -19 -19 -11 -10

Введите значение элемента, который хотите найти в массиве 45
В массиве нет элемента с таким значением.

        Выберите режим поиска
0. Выход из программы.
1. Линейный.
2. Бинарный.
3. Интерполяционный.
Выбранный пункт-> 3

Выбрано: Интерполяционный поиск.
Внимание! При любом вводе элементов массива они будут отсортированы.
Как заполнить массив?
1. Вручную, каждый элемент по очереди.
2. Используя случайные числа в неких границах.
Выбранный тип заполнения-> 65
Выберите существующий пункт меню!
Как заполнить массив?
1. Вручную, каждый элемент по очереди.
2. Используя случайные числа в неких границах.
Выбранный тип заполнения-> 2

Введите минимальное случайное число 7
Введите максимальное случайное число 8

Ваш массив:
7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8

Введите значение элемента, который хотите найти в массиве 8
Элемент со значением 8 найден по индексу [24].

        Выберите режим поиска
0. Выход из программы.
1. Линейный.
2. Бинарный.
3. Интерполяционный.
Выбранный пункт-> 3

Консоль отладки Microsoft V x + v

Выбрано: Интерполяционный поиск.
Внимание! При любом вводе элементов массива они будут отсортированы.
Как заполнить массив?
1. Вручную, каждый элемент по очереди.
2. Используя случайные числа в неких границах.
Выбранный тип заполнения-> 2

Введите минимальное случайное число 0
Введите максимальное случайное число 15

Ваш массив:
0 1 2 3 3 3 4 4 4 5 6 6 6 7 8 8 9 10 11 12 13 13 15 15 15

Введите значение элемента, который хотите найти в массиве 13
Элемент со значением 13 найден по индексу [20].

        Выберите режим поиска
0. Выход из программы.
1. Линейный.
2. Бинарный.
3. Интерполяционный.
Выбранный пункт-> 0

До скорых встреч!
```

ВЫВОД

В итоге этой работы была составлена программа с использованием трёх поисков. Сделан вывод о полезности и различиях этих поисков, сферах их применения. Выявлены плюсы и минусы поисков. В ходе работы были получены навыки работы с оператором множественного выбора и большим количеством функций.

Проведенная лабораторная работа была опубликована в общий доступ по адресу: https://github.com/Fedor0000/TheUltimateFolder/tree/main/Sem_2/Labs/16