Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №5**

**«Исследование алгоритмов сортировки»**

**ПО ДИСЦИПЛИНЕ «Основы алгоритмизации и программированния»**

Выполнил: студент учебной группы

ИСПк-205-52-00

Сибейкин Иван Алексеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

Цель работы

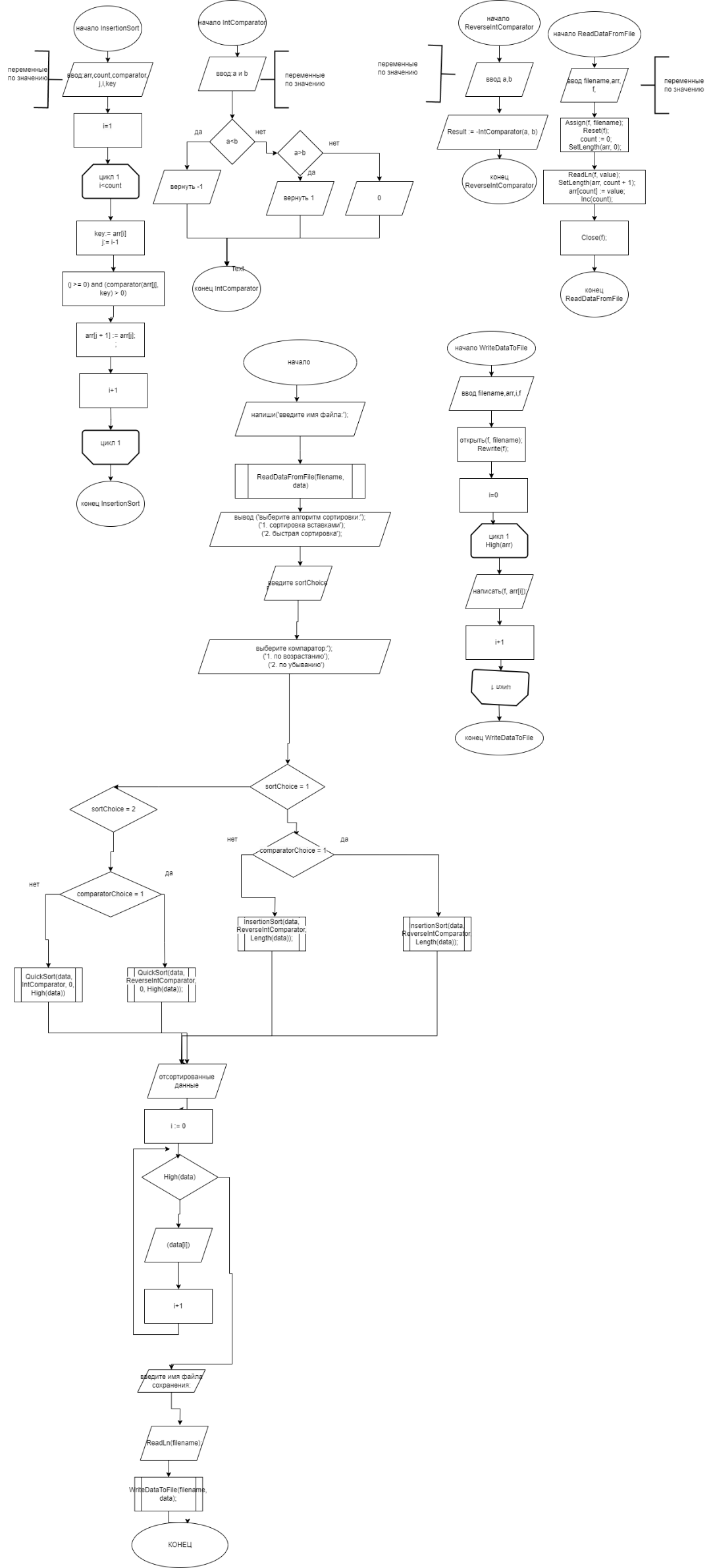
Цель работы: получить базовые сведения о наиболее известных алгоритмах сортировки, изучит принципы работы с текстовыми файлами.

Задание:

Формулировка задания (с вариантом)

Вариант 15

1. ﻿﻿﻿ Реализовать сортировку данных с помощью быстрого алгоритма.
2. Реализовать сортировку данных с помощью вставок.
3. В обоих случаях необходимо предусмотреть возможность изменения компаратора
4. (реализация компаратора в виде передаваемой в подпрограмму функции).
5. Считывание и вывод данных необходимо производить из текстового файла.
6. Для демонстрации работы программных реализаций самостоятельно подготовить варианты
7. входных данных (при этом объем тестовых файлов должен позволять оценить скорость работы



Код для программы в Pascal.

**Задание 1**

**program** SortAlgorithms;

**type**

TComparator = **function**(**const** a, b: Integer): Integer;

**procedure** InsertionSort(**var** arr: **array of** Integer; comparator: TComparator; count: Integer);

**var**

i, j: Integer;

key: Integer;

**begin**

**for** i := 1 **to** count - 1 **do**

**begin**

key := arr[i];

j := i - 1;

**while** (j >= 0) **and** (comparator(arr[j], key) > 0) **do**

**begin**

arr[j + 1] := arr[j];

j := j - 1;

**end**;

arr[j + 1] := key;

**end**;

**end**;

**procedure** QuickSort(**var** arr: **array of** Integer; comparator: TComparator; left, right: Integer);

**var**

i, j, pivot: Integer;

temp: Integer;

**begin**

**if** left < right **then**

**begin**

pivot := arr[(left + right) **div** 2];

i := left;

j := right;

**repeat**

**while** comparator(arr[i], pivot) < 0 **do** Inc(i);

**while** comparator(arr[j], pivot) > 0 **do** Dec(j);

**if** i <= j **then**

**begin**

// Обмен значениями с использованием временной переменной

temp := arr[i];

arr[i] := arr[j];

arr[j] := temp;

Inc(i);

Dec(j);

**end**;

**until** i > j;

QuickSort(arr, comparator, left, j);

QuickSort(arr, comparator, i, right);

**end**;

**end**;

**function** IntComparator(**const** a, b: Integer): Integer;

**begin**

**if** a < b **then** Result := -1

**else if** a > b **then** Result := 1

**else** Result := 0;

**end**;

**function** ReverseIntComparator(**const** a, b: Integer): Integer;

**begin**

Result := -IntComparator(a, b);

**end**;

**procedure** ReadDataFromFile(**const** filename: string; **var** arr: **array of** Integer);

**var**

f: TextFile;

value: Integer;

count: Integer;

**begin**

Assign(f, filename);

Reset(f);

count := 0;

SetLength(arr, 0); // Инициализируем пустым массивом

**while not** EOF(f) **do**

**begin**

ReadLn(f, value);

SetLength(arr, count + 1); // Увеличиваем размер массива

arr[count] := value; // Записываем значение

Inc(count); // Увеличиваем счетчик

**end**;

Close(f);

**end**;

**procedure** WriteDataToFile(**const** filename: string; **const** arr: **array of** Integer);

**var**

f: TextFile;

i: Integer;

**begin**

Assign(f, filename);

Rewrite(f);

**for** i := 0 **to** High(arr) **do**

WriteLn(f, arr[i]);

Close(f);

**end**;

**var**

data: **array of** Integer; // Используем стандартный массив

filename: string;

sortChoice, comparatorChoice: Integer;

**begin**

WriteLn('Введите имя файла:');

ReadLn(filename);

ReadDataFromFile(filename, data);

WriteLn('Выберите алгоритм сортировки:');

WriteLn('1. Сортировка вставками');

WriteLn('2. Быстрая сортировка');

ReadLn(sortChoice);

WriteLn('Выберите компаратор:');

WriteLn('1. По возрастанию');

WriteLn('2. По убыванию');

ReadLn(comparatorChoice);

**if** sortChoice = 1 **then**

**begin**

**if** comparatorChoice = 1 **then**

InsertionSort(data, IntComparator, Length(data))

**else**

InsertionSort(data, ReverseIntComparator, Length(data));

**end**

**else if** sortChoice = 2 **then**

**begin**

**if** comparatorChoice = 1 **then**

QuickSort(data, IntComparator, 0, High(data))

**else**

QuickSort(data, ReverseIntComparator, 0, High(data));

**end**;

WriteLn('Отсортированные данные:');

**for var** i := 0 **to** High(data) **do** WriteLn(data[i]);

WriteLn('Введите имя файла для сохранения отсортированных данных:');

ReadLn(filename);

WriteDataToFile(filename, data);

ReadLn;

**end**.

Вывод:

Изучение языка программирования Pascal позволило глубже понять принципы построения программ и эффективного применения основных языковых конструкций для решения различных задач. В ходе освоения Pascal были усвоены не только особенности синтаксиса, но и ключевые принципы программирования, что важно для развития логического мышления.

Также в процессе работы была изучена методология создания алгоритмических схем, что способствует упорядоченной организации решений и помогает лучше понять взаимодействие компонентов программы. Владение алгоритмическими методами стало основой для будущих программных разработок и создания более сложных решений.

Таким образом, поставленные цели были успешно выполнены, и приобретенные навыки будут активно применяться в последующих проектах. Базовое понимание Pascal укрепило уверенность в собственных силах и повысило мотивацию для дальнейшего изучения информатики и программирования.