Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

ОТЧЕТ

по учебной практике

на тему:

БИРЖА ТРУДА

Студент:

гр. 251001 Лашкин В.Н.

Руководитель:

асс. Фадеева Е.Е.

Минск 2023

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 3](#_Toc132052165)

[1. Задание на лабораторную работу 4](#_Toc132052166)

[1.1. Постановка задачи 4](#_Toc132052167)

[1.2. Эскиз ожидаемого результата 5](#_Toc132052168)

[2. Выполнение 6](#_Toc132052169)

[2.1. Разработка укрупненной схемы программы 6](#_Toc132052170)

[2.2. Разработка алгоритма чтения данных из файла 7](#_Toc132052171)

[2.3. Разработка алгоритма сортировки списка 8](#_Toc132052172)

[2.4. Разработка алгоритма поиска данных 9](#_Toc132052173)

[2.5. Разработка алгоритма добавления элемента в список 11](#_Toc132052174)

[2.6. Разработка алгоритма удаления элемента из списка 12](#_Toc132052175)

[2.7. Разработка алгоритма подбора вакансий для данного кандидата, вывода списка дефицитных вакансий 13](#_Toc132052176)

[2.8. Тестирование и отладка программы 15](#_Toc132052177)

[2.9. Итоговый текст программы 16](#_Toc132052178)

# Задание на лабораторную работу

## Постановка задачи

Реализовать программу, работающую с типизированными файлами. При выборе пункта меню 1 должно производиться чтение данных из файлов, созданных и сохраненных ранее в соответствующие им по типу и назначению динамические списки. Дальнейшие манипуляции над данными выполняются в динамических списках. В момент, когда выбран пункт меню 10, перезаписываются типизированные файлы данными из соответствующих динамических списков.

Программа должна содержать пользовательское меню, состоящее из следующих пунктов:

1. Чтение данных из файла
2. Просмотр всего списка (выпадает подменю с именами списков для просмотра: списки вакансий и кандидатов)
3. Сортировка списка
4. Поиск данных с использованием фильтров
5. Добавление данных в список (выпадает подменю с именами списков для добавления: списки вакансий и кандидатов)
6. Удаление данных из списка (выпадает подменю с именами списков для удаления: списки вакансий и кандидатов)
7. Редактирование данных (выпадает подменю с именами списков для редактирования: списки вакансий и кандидатов)
8. Подбор списка возможных вакансий для данного кандидата
9. Выход из программы без сохранения изменений
10. Выход с сохранением изменений

При реализации пункта 8 меню, полученные данные должны выводиться на экран и записываться в текстовый файл.

Имеется список фирм с вакансиями. Каждая запись списка содержит: название фирмы, наименование специальности, должность, оклад, количество дней отпуска, требования к нанимаемому: наличие высшего образования (да/нет), возрастной диапазон (min/max). Также имеется список кандидатов. Каждая запись списка содержит: ФИО кандидата, дату рождения, специальность, наличие высшего образования (да/нет), желаемую должность, min оклад. Требуется:

1. Для каждого кандидата подобрать список возможных вакансий;
2. Выдать список дефицитных вакансий (количество вакансий на должность ниже 10% процентов от количества кандидатов)

Предусмотреть возможность добавлять, удалять и корректировать записи из списков, а также просматривать списки полностью.

## Эскиз ожидаемого результата

# Выполнение

## Разработка укрупненной схемы программы

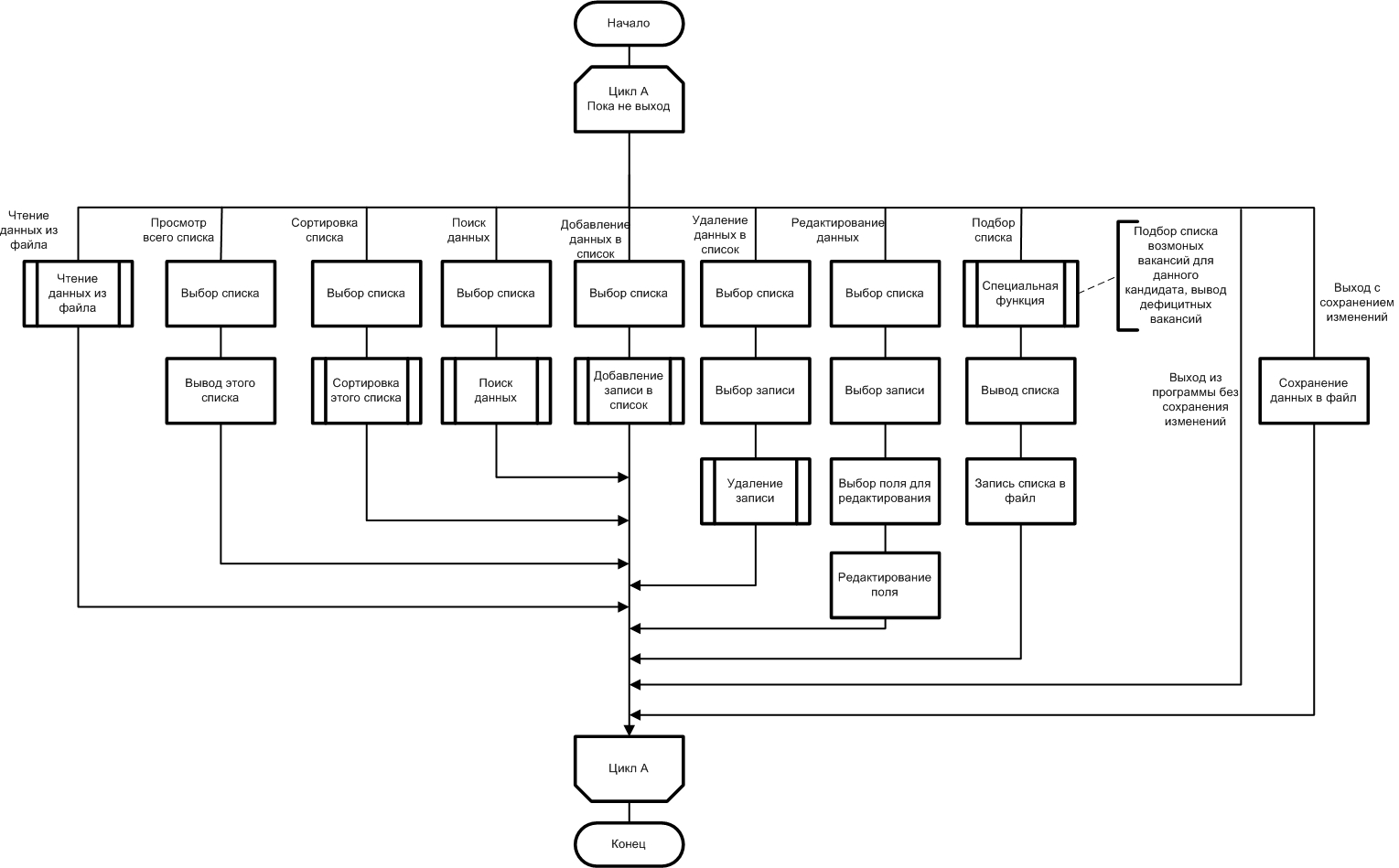


Рисунок 2.1 – Укрупненная схема программы

Эта схема отражает работу программы. После запуска программы пользователь может взаимодействовать с интерфейсом, который включает в себя меню из 10 пунктов. Программа завершается, когда пользователь выбирает «выход из программы».

## Разработка алгоритма чтения данных из файла



Рисунок 2.2 – Схема алгоритма чтения данных из файла

Эта схема отражает работу алгоритма чтения данных из файла. Пользователь вводит путь к файлу, который необходимо прочесть.

## Разработка алгоритма сортировки списка

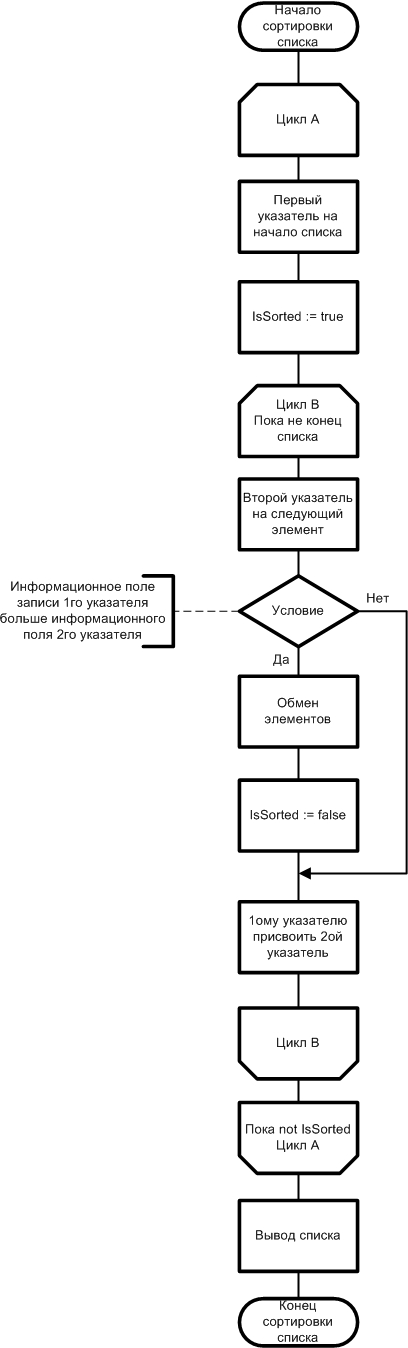


Рисунок 2.3 – Схема алгоритма сортировки списка

## Разработка алгоритма поиска данных

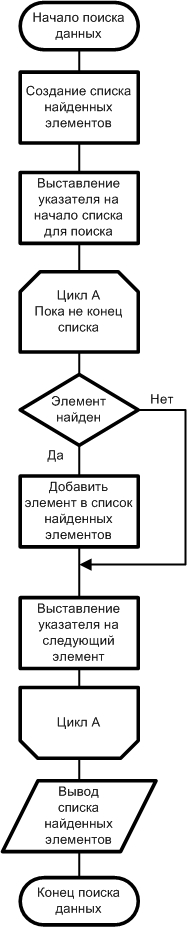


Рисунок 2.4 – Схема алгоритма поиска данных

Так как для некоторых полей найденных записей может быть несколько, то я создал однонаправленный список найденных записей. Устанавливается указатель на начало списка для поиска. Если элемент списка равен ключу, который ввел пользователь, то этот элемент заносится в список найденных элементов, выставляется указатель на следующий элемент. В конце цикла выставляется указатель на nil. Полученный список выводится на экран.

## Разработка алгоритма добавления элемента в список

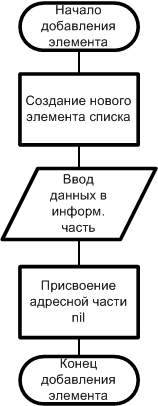


Рисунок 2.5 – Схема алгоритма добавления элемента в список

После того как пользователь выбрал список, в который он хочет добавить элемент, он вводит информационную часть записи добавляемого элемента.

## Разработка алгоритма удаления элемента из списка

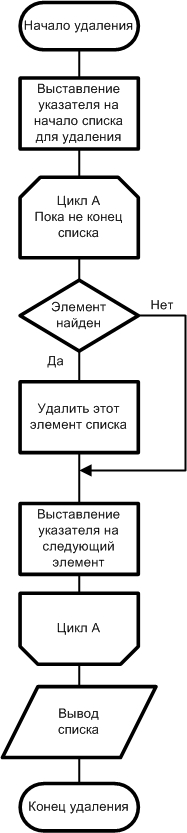


Рисунок 2.6 – Схема алгоритма удаления записи

Так как удаляемых элементов может быть несколько, то цикл А проходится по всему списку и удаляются только те записи, у которых информационная часть совпадает с введенным пользователем ключом.

## Разработка алгоритма подбора вакансий для данного кандидата, вывода списка дефицитных вакансий

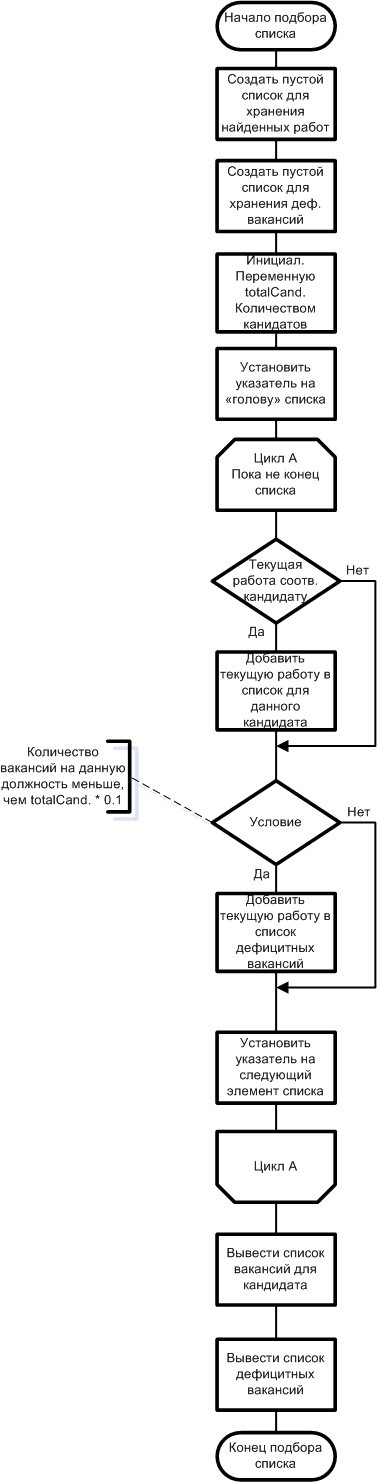


Рисунок 2.7 – Схема алгоритма подбора вакансий для данного кандидата, вывода списка дефицитных вакансий

Пользователь вводит должность, специальность, оклад. Создается список вакансий для данного кандидата, соответствующие введенным данным. Если число вакансий на данную должность меньше, чем 10% от общего количества кандидатов, то эта должность вносится в список дефицитных вакансий. В конце алгоритма эти два списка выводятся на экран.

## Тестирование и отладка программы

Таблица 2.1. Прохождение тестов программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специфика тестирования | Номер теста | Вводимые данные | Ожидаемый результат | Полученный результат |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Итоговый текст программы