Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерного проектирования

Дисциплина «Технологии разработки программного обеспечения»

|  |
| --- |
|  |
|  |
|  |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

на тему:

**«Разработка программы учета выплат заработанной платы сотрудникам предприятия»**

|  |  |
| --- | --- |
|  | Выполнил: студент группы 114301  Тиунчик Даниил Романович |
|  | Проверил: Кабариха Виктор Александрович |

Минск 2022

**Задание по курсовой работе**

Информация о сотрудниках предприятия содержит: ФИО сотрудника, название отдела, должность, размер заработной платы за месяц.

Индивидуальное задание: вычислить общую сумму выплат за месяц по каждому отделу, а также среднемесячный заработок сотрудников по каждому отделу. Вывести список сотрудников, у которых зарплата ниже введенной с клавиатуры.

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 4](#_Toc103124327)

[ВВЕДЕНИЕ 11](#_Toc103124328)

[1 Анализ и моделирование предметной области программного средства 12](#_Toc103124329)

[1.1 Описание предметной области 12](#_Toc103124330)

[1.2 Разработка функциональной модели предметной области в нотации IDEF0 12](#_Toc103124331)

[1.3 Разработка BPMN-модели основного процесса предметной области 14](#_Toc103124332)

[1.3.1 Теоретическая часть 14](#_Toc103124333)

[1.3.2 Практическая часть 15](#_Toc103124334)

[1.4 Анализ требований к разрабатываемому программному средству. Спецификация функциональных требований 17](#_Toc103124336)

[1.5 UML-модели представления программного средства и их описание 18](#_Toc103124337)

[1.5.1 Теоретическая часть 18](#_Toc103124338)

[1.5.2 Практическая часть 19](#_Toc103124339)

[2 Проектирование и конструирование программного средства 22](#_Toc103124340)

[2.1 Постановка задачи 22](#_Toc103124341)

[2.2 Разработка модульной структуры программы 22](#_Toc103124342)

[2.3 Выбор способа организации данных 23](#_Toc103124344)

[2.4 Разработка перечня пользовательских функций программы 23](#_Toc103124345)

[2.5 Разработка схем алгоритмов работы программы 23](#_Toc103124346)

[3 Тестирование и проверка работоспособности программного средства 27](#_Toc103124349)

[4 Инструкция по развертыванию приложения и сквозной тестовый пример 28](#_Toc103124354)

[4.1 Авторизация 28](#_Toc103124355)

[4.2 Модуль администратора 29](#_Toc103124356)

[4.3 Модуль пользователя 31](#_Toc103124358)

# ВВЕДЕНИЕ

Учёт выплат заработанной платы сотрудникам предприятий является актуальным вопросом, так как, при расширении капитала, количество предприятий и людей работающих на них увеличивается с каждым годом, а рассчитывать бухгалтерские задачи на бумаге нецелесообразно. Поэтому компьютерные технологии способствуют автоматизации этого процесса, а также облегчают и повышают эффективность работы предприятий и компаний, ускоряют процессы обработки данных.

Данная курсовая работа посвящена конструированию программы для расчёта выплат заработанной платы сотрудникам предприятия. Она представляет из себя создание консольного приложения на языке C в среде разработки Microsoft Visual Studio.

При разработке программы использовались принципы разработки алгоритмов, организации работы с файлами, массивами структур и хеширование.

Эта программа может использоваться в любой сфере деятельности и на любых предприятиях. Она поможет рассчитать утечки денежных средств предприятия, ведение базы данных сотрудников, пересчёт информации по отделам и определении слабых звеньев. Данное консольное приложение также вычисляет заработанную плату сотрудника относительно его позиции на предприятии и каждый работник сможет в любое время следить за этим. Основные задачи, которые выполняет данная программа:

* Вычисление общей суммы выплат за месяц.
* Среднемесячный заработок сотрудников.
* Вывод сотрудников, у которых зарплата меньше введённой.
* Ведение базы данных сотрудников.

**1 Анализ и моделирование предметной области программного средства**

# 1.1 Описание предметной области

# ТРЕБОВАНИЯ К ПРОГРАММЕ

# Разработать программу учета выплат заработной платы сотрудникам предприятия.

# Информация о сотрудниках предприятия содержит: Ф.И.О. сотрудника, название отдела, должность, размер заработной платы за месяц.

# Вычислить общую сумму выплат за месяц по каждому отделу, а также среднемесячный заработок сотрудников по каждому отделу. Вывести список сотрудников, у которых зарплата ниже введенной с клавиатуры.

# Реализовать авторизацию для входа в систему, функционал администратора и функционал пользователя, как минимум три вида поиска, как минимум три вида сортировки.

# 

# Исходные требования к курсовой работе

# Язык программирования С++.

# Среда разработки Microsoft Visual Studiо.

# Вид приложения – консольное.

# Парадигма программирования – процедурная.

# Способ организации данных – структуры (struct).

# Способ хранения данных – файлы.

# Каждая логически завершенная подзадача программы должна быть реализована в виде отдельной функции.

# Построение программного кода должно соответствовать соглашению о коде «С++ Code Convention».

# К защите курсовой работы представляются: консольное приложение и пояснительная записка.

# 

# Функциональные требования к курсовой работе

# Первым этапом работы программы является авторизация – предоставление прав доступа.

# В рамках данного этапа необходимо считать данные из файла с учетными записями пользователей следующего вида:

# login;

# password;

# role (данное поле служит для разделения в правах администраторов и пользователей).

# После ввода пользователем своих персональных данных (логина и пароля) и сверки со считанной из файла информацией необходимо предусмотреть возможность входа:

# в качестве главного администратора (в этом случае role = 2),

# в качестве администратора (в этом случае role = 1),

# в качестве пользователя (в этом случае role = 0).

# 

# Если файл с учетными записями пользователей не существует, то

# необходимо программно создать его и записать учетные данные администратора.

# Регистрация новых пользователей осуществляется самим пользователем путем ввода желаемых логина и пароля и ожидания подтверждения администратором новой учетной записи. Для реализации этого способа в структуре учетных записей пользователей необходимо предусмотреть дополнительное поле activation:

# login;

# password;

# role;

# activation (данное поле служит для подтверждения или блокировки администратором учетных записей).

# По умолчанию activation = 0 при попытке зарегистрироваться; далее администратор меняет значение на activation = 1 и тем самым подтверждает новую учетную запись: пользователь может осуществить вход в систему.

# 

# Вторым этапом работы программы является собственно работа с данными, которая становится доступной только после прохождения авторизации. Данные хранятся в отдельном файле.

# Для работы с данными должны быть предусмотрены два функциональных модуля: модуль администратора и модуль пользователя.

# Модуль администратора включает следующие подмодули:

# Управление учетными записями пользователей:

# просмотр всех учетных записей;

# редактирование учетной записи;

# удаление учетной записи.

# Работа с данными:

# просмотр всех данных;

# добавление новой записи;

# редактирование записи;

# удаление записи;

# вывод списка сотрудников, у которых зарплата ниже введенной с клавиатуры;

# cортировка данных (по трем различным параметрам):

# по фамилии;

# по отделу;

# по зарплате;

# поиск данных (по трем различным параметрам):

# по фамилии сотрудника;

# по должности сотрудника;

# сотрудника с наибольшей зарплатой.

# Модуль пользователя включает подмодуль работы с данными со следующими функциональными возможностями:

# просмотр всех данных;

# добавление новой записи;

# редактирование записи;

# удаление записи;

# вывод списка сотрудников, у которых зарплата ниже введенной с клавиатуры;

# cортировка данных (по трем различным параметрам):

# по фамилии;

# по отделу;

# по зарплате;

# поиск данных (по трем различным параметрам):

# по фамилии сотрудника;

# по должности сотрудника;

# сотрудника с наибольшей зарплатой.

# Для реализации перечисленных модулей/подмодулей необходимо создавать меню с соответствующими пунктами.

# Необходимо предусмотреть:

# Обработку исключительных ситуаций:

# имя пользователя или пароль введены неверно;

# запись с указанным идентификатором не найдена;

# пользователь с таким именем уже существует;

# введённые данные не соответствуют формату поля;

# Возможность возврата назад (навигация);

# Обратную связь с пользователем (вывод сообщения об успешности создания/удаления файла либо записи).

# 

# Требования к программной реализации

# Все переменные и константы должны иметь осмысленные имена в рамках тематики варианта к курсовой работе.

# Имена функций должны быть осмысленными и строиться по принципу «глагол + существительное». Если функция выполняет какую-либо проверку и возвращает результат типа bool, то ее название должно начинаться с глагола is (например, isFileExist, isUnicLogin).

# Код не должен содержать неименованных числовых констант (так называемых «магических» чисел), неименованных строковых констант (например, имен файлов и др.). Подобного рода информацию следует выносить в глобальные переменные с атрибутом const. По правилам хорошего стиля программирования тексты всех информационных сообщений, выводимых пользователю в ответ на его действия, также оформляются как константы.

# Код необходимо комментировать (как минимум в части нетривиальной логики).

# Код не должен дублироваться – для этого существуют методы и функции.

# Одна функция решает только одну задачу (например, не допускается в одной функции считывать данные из файла и выводить их на консоль – это две разные функции). При этом внутри функции возможен вызов других функций.

# Следует избегать длинных функций и глубокой вложенности: текст функции должен умещаться на один экран, а вложенность блоков и операторов должна быть не более трёх.

# Следует выносить код логически независимых модулей в отдельные .cpp файлы и подключать их с помощью заголовочных .h файлов.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

Предметной областью данной программы является любое предприятие, где присутствуют работники. Для нормальной работы приложения потребуется техническое обеспечение, квалифицированные работники, а также сервера с данными для хранения информации о сотрудниках организации.

Автоматизированная программа предназначена для быстрой и качественной обработки, учета и контроля информации, задействованной в данной предметной области для оценки заработанной платы сотрудников.

Под учетом в данной программе подразумевается быстрый поиск информации по всем категориям, присутствующим в базе данных. Например, по запросу фамилии сотрудника должна выводиться соответствующая информация.

Система создаётся для следующих групп пользователей:

* Руководство предприятия.
* Начальники отделов.
* Сотрудники отделов.

Функциональные возможности данного приложения:

* Ведение базы данных (запись, чтение).
* Обеспечение защиты данных от несанкционированного или случайного доступа.
* Обеспечение логической непротиворечивости базы данных.

# 1.2 Разработка функциональной модели предметной области в нотации IDEF0

Методология IDEF0 (Integrated DEFinition) – это совокупность методов, правил и процедур, предназначенных для построения функциональной модели предметной области. Функциональная модель IDEF0 отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями; основана на методе SADT Росса.

Методология IDEF0 основана на следующих концептуальных положениях:

* модель – искусственный объект, представляющий собой отображение (образ) системы и ее компонентов;
* блочное моделирование и его графическое представление;
* лаконичность и точность;
* передача информации.

Концепция предполагает собою совокупность связанных и взаимодействующих элементов, исполняющих определенную полезную работу. Компонентами концепции могут являться всевозможные комбинации различных сущностей.

Все, что происходит в системе и ее элементах, называется функциями.

Для построения контекстной диаграммы процесса выплаты заработанной платы сотрудникам нужно проанализировать, что преобразуются или расходуются функцией, чтобы создать то, что появится на ее выходе, механизмы, участвующие в процессе, условия, необходимые функции, чтобы произвести правильный выход и сами данные либо материальные объекты, полученные в результате работы функции.

Для того, чтобы выплатить заработанную плату сотрудникам, функция использует документы студента, информацию о нем и его заявление на общежитие.

При расчёте заработанной платы используются такие документы как: законодательство, трудовой кодекс, нормативы.

В процессе участвуют бухгалтерия и администрация. Подключается база данных о сотрудниках. На выходе функции можно увидеть расчётную ведомость и количество денег, которое получил сотрудник предприятия(рисунок 1.1).

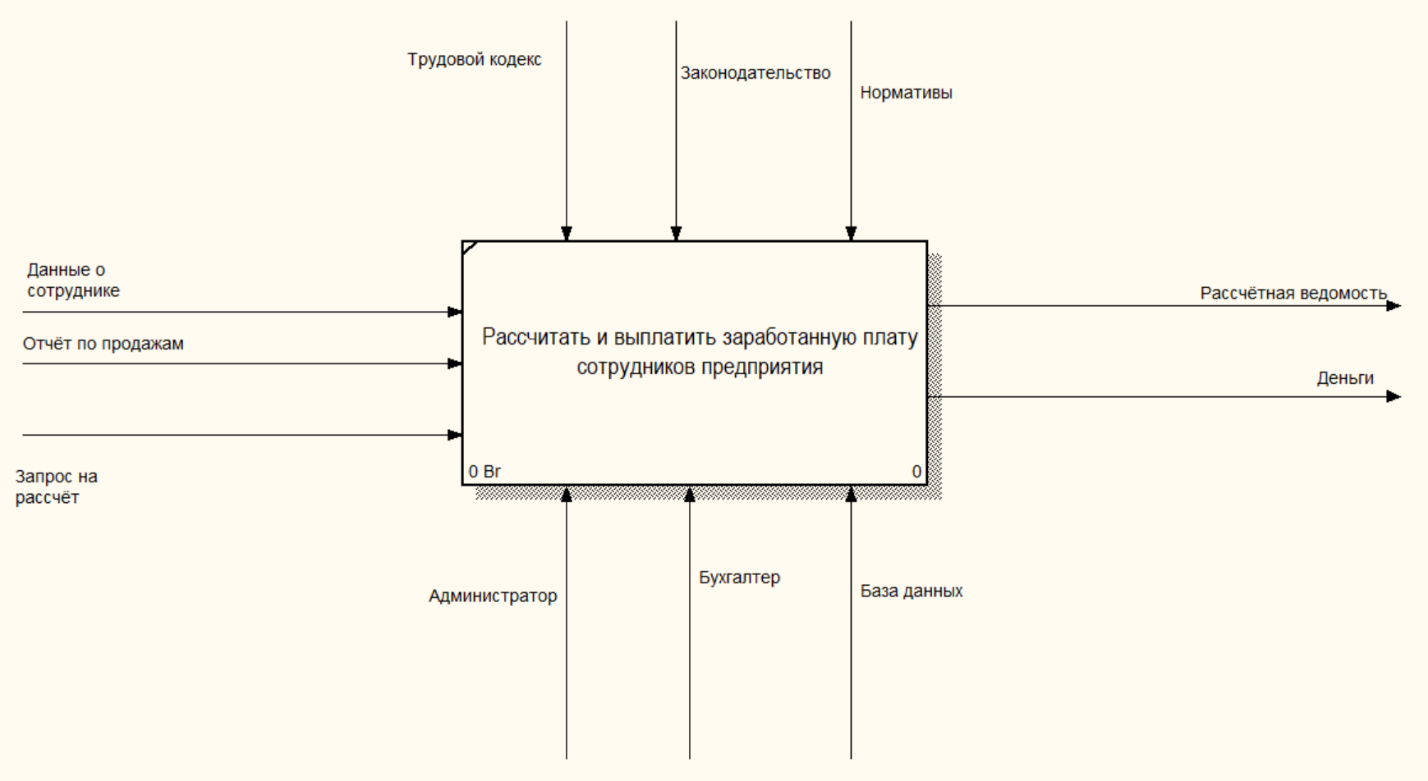


Рисунок 1.1 ­­­­­— Контекстная диаграмма процесса «Рассчитать и выплатить заработанную плату сотрудников предприятия»

Декомпозиция процесса состоит из 4 компонентов (рисунок 1.2):

* рассмотрение запроса;
* проверки наличия данных о сотруднике;
* расчета заработанной платы;
* выдачи зарплаты.

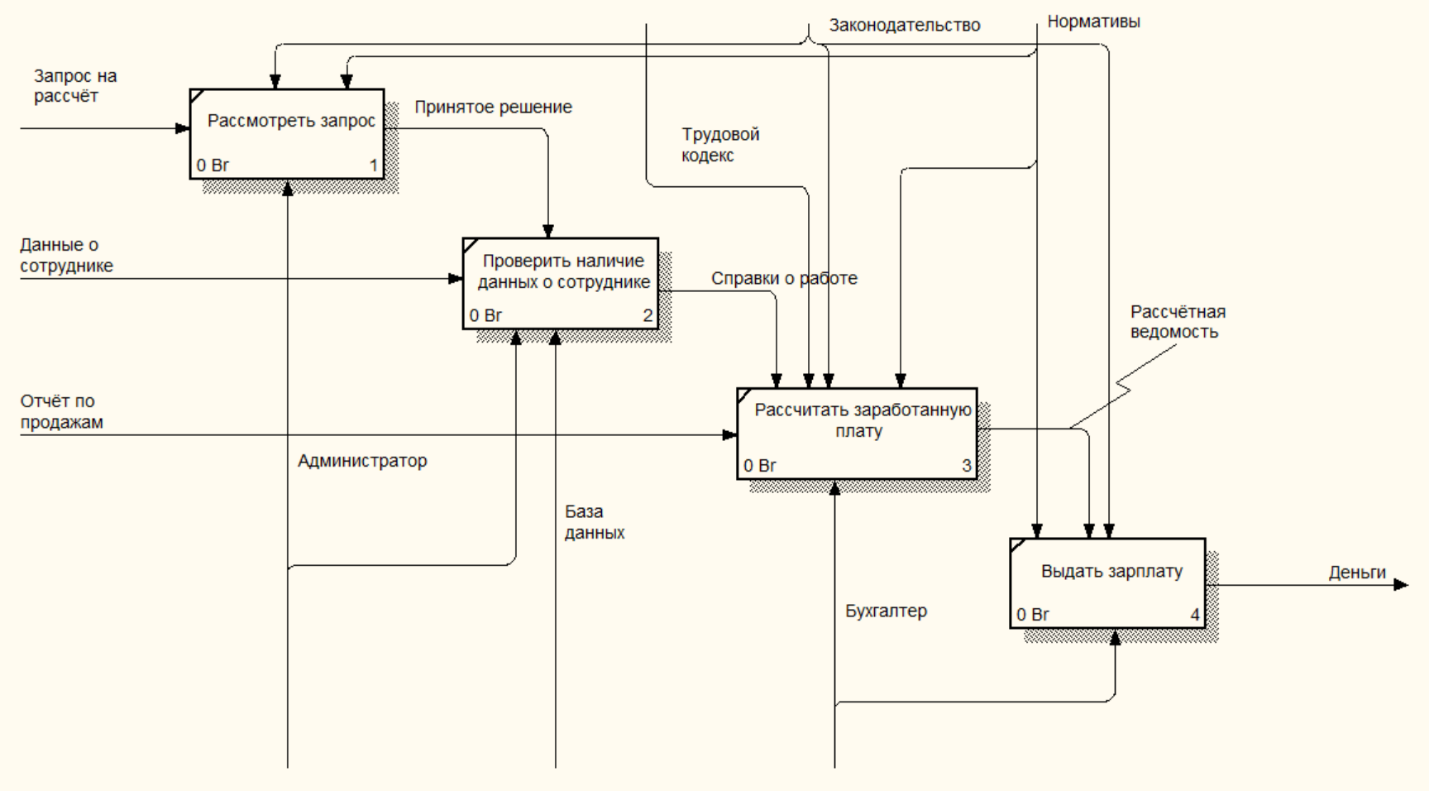


Рисунок 1.2 — Диаграмма декомпозиции процесса «Рассчитать и выплатить заработанную плату сотрудников»

Таким образом, благодаря процессу декомпозиции, объект описан на уровне детализации, необходимом для создания точного списка сотрудников и выдачи заработанной платы.

# 1.3 Разработка BPMN-модели основного процесса предметной области

# 1.3.1 Теоретическая часть

Bizagi Process Modeler *–* это очень эффективная программа, специализирующаяся на разработке диаграмм или цепей бизнес-процессов.

Программный продукт Bizagiпозволяет строить модели деятельности, выполнять и совершенствовать бизнес-процессы, используя графическую среду, без необходимости программирования. Любой процесс, описанный в нотации BPMN, представляет собой последовательное или параллельное выполнение различных действий (операций) с указанием определённых бизнес-правил.

Моделирование в BPMNосуществляется посредством диаграмм с небольшим числом графических элементов. Это помогает пользователям быстро понимать логику процесса. Выделяют четыре основные категории элементов:

* объекты потока управления: события, действия и логические операторы;
* соединяющие объекты: поток управления, поток сообщений и ассоциации;
* роли: пулы и дорожки;
* артефакты: данные, группы и текстовые аннотации.

Элементы этих четырёх категорий позволяют строить простейшие диаграммы бизнес-процессов. Для повышения выразительности модели спецификация разрешает создавать новые типы объектов потока управления и артефактов.

Нотация моделирования бизнес-процессов (BPMN) — это метод составления блок-схем, отображающий этапы выполнения бизнес-процесса от начала до конца.

Основная цель BPMN — создание стандартного набора условных обозначений, понятных всем бизнес-пользователям.

Нотация BPMN описывает условные обозначения для отображения бизнес-процессов в виде диаграмм бизнес-процессов. BPMN ориентирована как на технических специалистов (разработчиков, ответственных за реализацию процессов), так и на бизнес-пользователей (бизнес-аналитиков, создающих и улучшающих процессы) и менеджеров, следящих за процессами и управляющих ими. Следовательно, BPMN призвана служить связующим звеном между фазой дизайна бизнес-процесса и фазой его реализации. Для этого язык использует базовый набор интуитивно понятных элементов, которые позволяют определять сложные семантические конструкции.

# 1.3.2 Практическая часть

Процесс выдачи заработанной платы сотрудникам начинается с подачи запроса и нахождения данных о сотруднике. Эта задача выполняется администрацией предприятия. Далее проверяются справки о работе. При их отсутствии пользователю отсылается письмо с отказом, в ином случае происходит проверка наличия нужных документов. Если у сотрудника присутствуют не все документы – ему отправляется сообщение о недостаточном количестве справок, документов. Если все в порядке, то при расчёте заработанной платы бухгалтерия учитывает отчёт по продажам. Процесс завершается выдачей заработанной платы(рисунок 2).

# 

Рисунок 2 – диаграмма BPMN бизнес-процесса «Процесс подачи заявки на общежитие»

В результате, BPMN-схемы описывают бизнес-процессы единым стандартизированным языком, который понятен всем участникам независимо от уровня их технических познаний.

# 1.4 Анализ требований к разрабатываемому программному средству. Спецификация функциональных требований

Для выплаты заработанной платы формируется список сотрудников, который включает: ФИО сотрудника, дату рождения, название отдела, должность, дату начала работы.

Индивидуальное задание: вычислить общую сумму выплат за месяц по каждому отделу, а также среднемесячный заработок сотрудников по каждому отделу. Вывести список сотрудников, у которых зарплата ниже введенной с клавиатуры.

Общее для всех вариантов задание: реализовать авторизацию для входа в систему, функционал администратора и функционал пользователя.

Исходные данные:

* Тема курсовой работы выбирается из списка, приведенного в приложении А;
* язык программирования С/С++;
* среда разработки Microsoft Visual Studio;
* вид приложения – консольное;
* парадигма программирования – процедурная;
* способ организации данных – структуры;
* способ хранения данных – файлы;
* каждая логически завершенная подзадача программы должна быть реализована в виде отдельной функции;
* построение программного кода должно соответствовать соглашению о коде «С++ Code Convention»;
* к защите курсовой работы представляются: консольное приложение и пояснительная записка.

Первым этапом работы программы является авторизация (либо создание аккаунта и последующая авторизация). Далее требуется считать данные из файла с учетными записями пользователей. Предусмотреть возможность входа как администратора, так и пользователя. Регистрация новых пользователей осуществляется администратором в режиме работы с учетными записями пользователей либо пользователями при входе в программу. Для работы с данными должны быть предусмотрены два функциональных модуля: модуль администратора и модуль пользователя.

Модуль администратора включает следующие подмодули (с указанием функциональных возможностей):

* Управление учетными записями пользователей;
* Работа с данными.

Модуль пользователя включает подмодуль работы с данными со следующими функциональными возможностями:

* просмотр всех данных;
* выполнение индивидуального задания;
* поиск данных (как минимум по трем различным параметрам);
* сортировка (как минимум по трем различным параметрам).

В курсовой работе необходимо предусмотреть:

* ввод пользователем данных, несоответствующих формату поля;
* ввод пользователем нелогичных данных;
* отсутствие файла с данными для чтения;
* отсутствие данных по результатам поиска;
* выход за пределы массива/вектора номера удаляемой записи;
* возможность возврата назад (навигация);
* запрос на выполнение необратимых действий, а именно подтверждение удаления вида «Вы действительно хотите удалить файл (запись)?»;
* обратную связь с пользователем, например, вывод сообщения об успешности удаления/редактирования записи и т. д.

Все переменные и константы должны иметь [осмысленные](http://ci-plus-plus-snachala.ru/?page_id=228) имена в рамках тематики варианта курсовой работы. Переменным рекомендуется присваивать имена, состоящие из букв нижнего регистра. Имена функций должны быть осмысленными, начинаться с буквы нижнего регистра, строиться по принципу глагол плюс существительное. Не допускается использование оператора прерывания goto. Код не должен содержать неименованных числовых констант, неименованных строковых констант. Подобного рода информацию следует представлять, как глобальные константы. Код необходимо комментировать. Код не должен дублироваться. Одна функция решает только одну задачу. При этом внутри функции возможен вызов других функций. Выполнение операций чтения/записи в файл должно быть сведено к минимуму. Следует избегать глубокой вложенности условных и циклических конструкций. Следует избегать длинных функций. Следует выносить код логически независимых модулей в отдельные .cpp файлы и подключать их с помощью заголовочных .h файлов.

# 2 Проектирование и конструирование программного средства

# 2.1 Постановка задачи

Задача заключается в написании и отладке программы для формирования списка сотрудников предприятия. Список должен содержать: ФИО сотрудника, дату рождения, название отдела, должность, дату начала работы.

# 2.2 Разработка модульной структуры программы

# 

Рисунок 4 – Модульная структура программы

# 2.3 Выбор способа организации данных

В качестве выбора способа описания входных данных приводится описание следующих типов struct:

* + для учетных записей пользователей;
  + для данных;
  + для данных о сотрудниках;
  + для данных о отделах.

Структура учетных данных пользователей содержит такие поля, как логин, пароль, уровень доступа и роль пользователя (пользователь, администратор или главный администратор).

Структура данных содержит полное ФИО сотрудника, название отдела, должность, размер заработной платы за месяц.

В качестве способа объединения входных данных были выбраны массивы структур, которые объявлены локально. Глобальные переменные не используются.

# 2.4 Разработка перечня пользовательских функций программы

Программа содержит прототипы функций для просмотра, добавления, редактирования, удаления, сортировки, поиска списка сотрудников или пользователей, а также чтение и запись в файл и выполнение индивидуального задания.

Прототипы функций для работы с пользователями:

void programmStart();

void getNumberOfAccountsInFile(int&);

void readFileAccounts(Accounts\*);

void writeAccountsInFile(Accounts\*, int);

void showAccounts(Accounts\*, int);

Accounts\* resizeOfArray(Accounts\*, int&, int);

void addAccount(Accounts\*, int&);

void newAccount(Accounts\*, int&);

void delAccounts(Accounts\*, int&);

Accounts\* menuLoginOrRegistration(Accounts\*, int&);

void checkLoginPassword(Accounts\*, int&);

void checkRole(Accounts\*, int&, int);

void adminMenu(Accounts\*, int&);

void adminMenuOfAdminAccounts(Accounts\*, int&);

void userMenu();

Прототипы функций для работы с сотрудниками:

void userMenu();

void getNumberOfEmployeesInFile(int&);

void readFileEmployees(Employees\*, int&);

void writeFileEmployees(Employees\*, int);

Employees\* menuEmployees(Employees\*, int&);

void showEmployees(Employees\*, int);

Employees\* reallocationForEmployees(Employees\*, int&, int);

void addEmployees(Employees\*, int&);

void editEmployees(Employees\*, int&);

void delEmployees(Employees\*, int&);

void menuIndTask(Employees\*, int);

void indTaskLessSalary(Employees\*, int);

void indTaskDepartments(Employees\*, int);

Departments\* indTask\_Resize(Departments\*, int&, int);

Employees\* menuSortEmployees(Employees\*, int);

Employees\* sortSurname(Employees\*, int);

Employees\* sortDepartment(Employees\*, int);

Employees\* sortSalary(Employees\*, int);

void menuFindEmployees(Employees\*, int);

void findSurname(Employees\*, int);

void findPosition(Employees\*, int);

void findMaxSalary(Employees\*, int);

void showFind(Employees\*, int, int);

# 2.5 Разработка схем алгоритмов работы программы

Функция main является точкой входа в программу, вызывает основные функции инициализации, а также функцию с авторизацией и дальнейшей обработкой данных. На рисунке 5.1 представлена блок-схема алгоритма функции main.

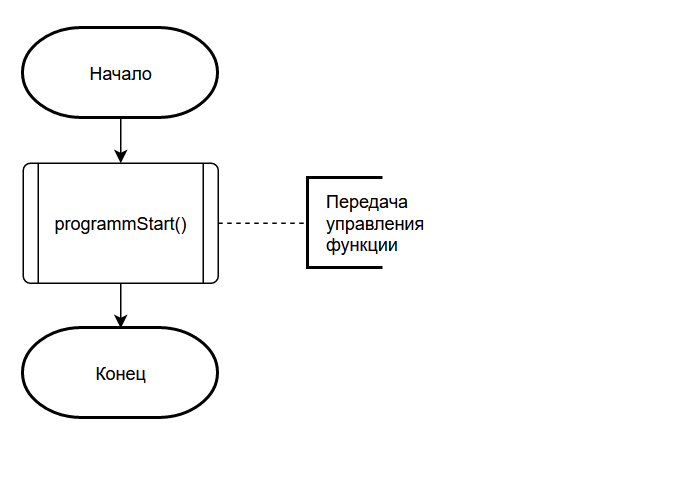


Рисунок 5.1 - Алгоритм функции main

Схему работы функции для вывода списка пользователей можно увидеть на рисунке 5.2.

# 

Рисунок 5.2 - Алгоритм функции showAccounts

Схема работы функции по добавлению студента в список показана на рисунке 5.3.

# 

Рисунок 5.3 - Алгоритм функции userMenu

# 3. РАЗРАБОТКА АЛГОРИТМОВ ПРОГРАММЫ

Учтены проверки на некорректный ввод в некотором диапазоне, диапазон выбирается в зависимости от места вызова функции в программе.

# 

Таким образом, были исключены всевозможные ошибки ввод чисел, будь то ошибка потокового ввода или просто некорректное згачение.

# 4 Инструкция по развертыванию приложения и сквозной тестовый пример

# 4.1 Авторизация

Авторизация состоит из 3 пунктов: «Вход в аккаунт», «Регистрация» и «Выход» (рисунок 7.1). Создать аккаунт можно только с уникальным логином, регистрация не пройдёт, если он уже существует. Пароль должен содержать 4 и более элементов, а на его ввод при авторизации даётся 3 попытки, если пользователь их исчерпал - его возврат в главное меню. При несовпадении логина и пароля либо при отсутствии аккаунта на экран выводится сообщение (рисунок 7.2).

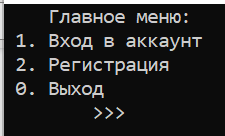


Рисунок 7.1 – Пример создания аккаунта администратора

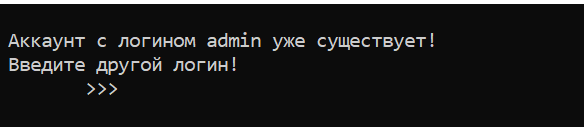


Рисунок 7.2 – Сообщение об ошибке

# 4.2 Модуль администратора

Меню администратора содержит такие пункты, как «Работа с аккаунтами», «Работа с сотрудниками» и «Выход из учётной записи» (рисунок 8.1).

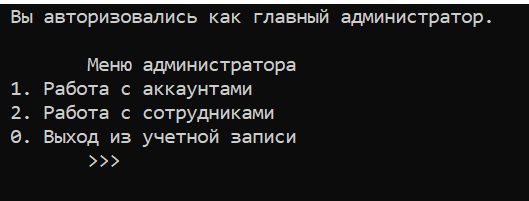


Рисунок 8.1 – Меню администратора

Пункт «Работа с аккаунтами» содержит следующие подпункты (рисунок 8.2):

# 

Рисунок 8.2 – Управление учетными записями пользователей

Пункт «Работа с сотрудниками» содержит следующие подпункты (рисунок 8.3):

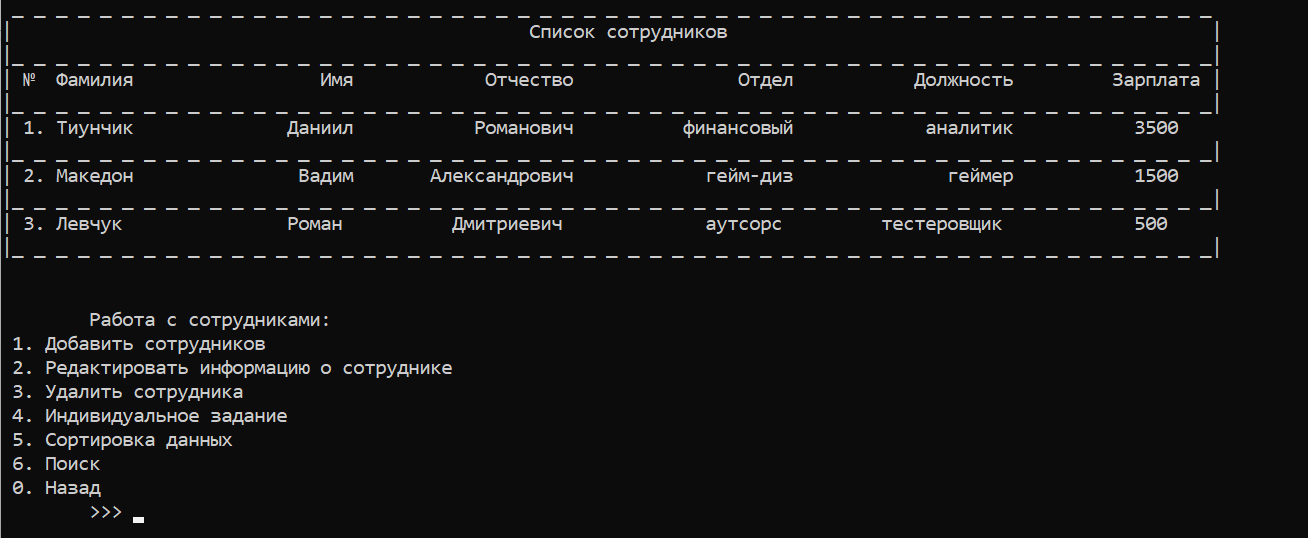


Рисунок 8.3 – Работа с данными

Также администратор может изменять данные пользователей, а главный администратор данные пользователей и администраторов (рисунок 8.4).

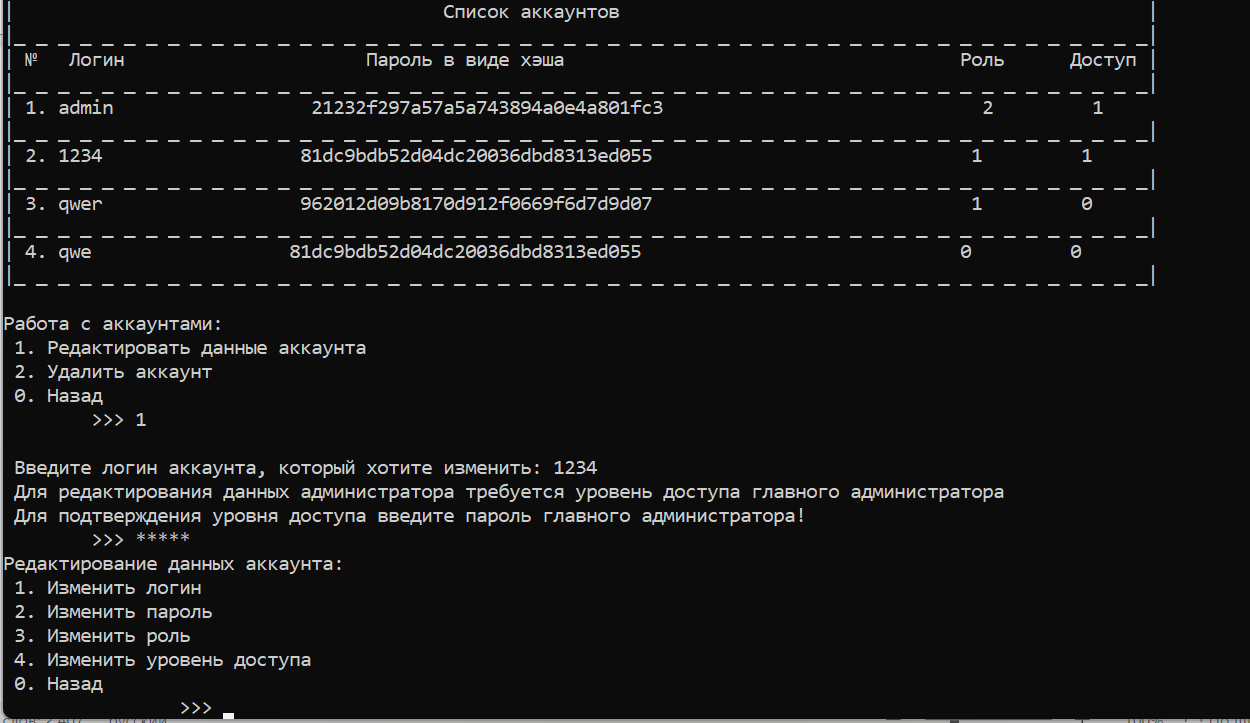


Рисунок 8.4 – Изменение данных пользователя

На всех этапах учтена навигация.

# 4.3 Модуль пользователя

Меню пользователя состоит из таких пунктов, как «Добавить сотрудников», «Редактировать информацию о сотруднике», «Удалить сотрудника», «Индивидуальное задание», а также сортировки и поиск по разным полям (рисунок 9.1). Также учтена навигация.

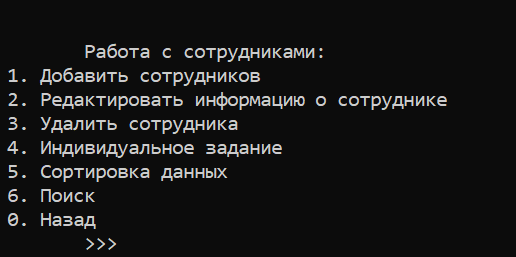


Рисунок 9.1 – Меню пользователя

Пример просмотра данных сотрудников можно увидеть на рисунке 9.2. Создается таблица, содержащая нижеизложенные колонки.

# 

Рисунок 9.2 – Просмотр данных студентов

После сортировки данных, программа предложит сохранить полученный списко в файл

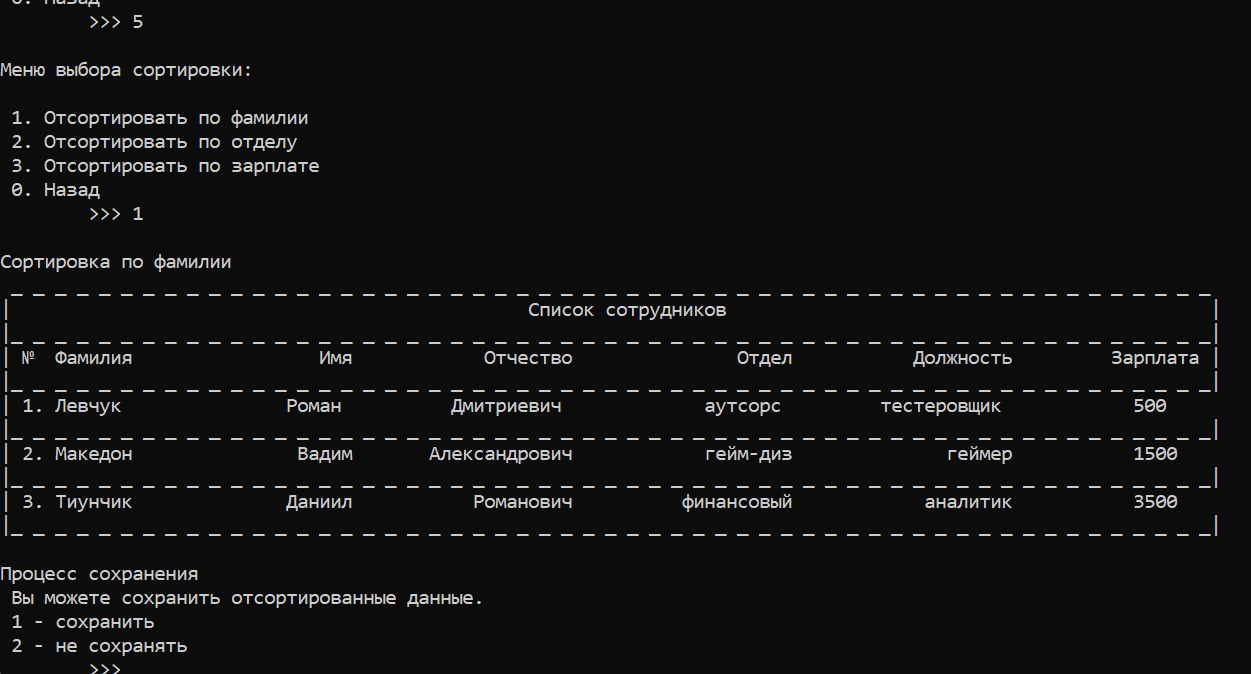


Рисунок 9.3 – Сортировка данных студентов по фамилии

Учтены все требования, описанные в пункте 1.4.

**Заключение**

Исследованы технологии нисходящего проектирования и принципов методологии структурного программирования, графические блок-схемы алгоритмов на языке С/С++.

Получены теоретические знания и практические навыки по конструированию и реализации программ на языке С/С++ в рамках процедурной парадигмы.

Написана и отлажена программа учёта выплат заработанной платы сотрудникам предприятия.

Листинг кода