# Введение

Данный курсовой проект выполняется на тему «Разработка программного средства для учета сотрудников организации».

Автоматизация учета сотрудников организации позволит оперативно вести учет сотрудников, хранить их контактные данные, должности и ставки в организации, проводить начисление заработной платы, формировать отчеты по заработной плате, вести электронный документооборот, устранить несоответствие данных в различных документах.

Целью курсового проекта является разработка программного продукта в среде Visual Studio 2019 на языке C#, позволяющее хранить данные о сотрудниках и их координатах и заработной плате в базе данных. Приложение помогает осуществлять поиск в таблицах базы данных, формировать и выводить отчеты о заработной плате на экран компьютера, на печатающее устройство или сохранять на жесткий диск в различных форматах.

# 1 Постановка задачи

## 1.1 Описание предметной области

Программное средство разработано для облегчения учета сотрудников организации, получения быстрой справочной информации по их адресам, телефонам, должностям, заработной плате, формирования отчета по заработной плате, основании ее начисления, даты начисления. Областью применения разрабатываемого приложения может являться небольшая организация, сотрудники которой получают заработную плату не по условиям сдельно-премиальной оплаты труда, а находящиеся на фиксированных окладах с фиксированной премией. Автоматизация учета сотрудников позволит упростить процесс контроля расхода денежных средств фонда заработной платы.

## 1.2 Актуальность решаемой задачи

Начисление заработной платы сотрудникам организации является типовой бухгалтерской задачей любой организации, кадрам же дополнительно необходимо хранить информацию о месте проживания человека и его телефонах. Существуют программы, облегчающие учет сотрудников и их заработной платы, самой известной из них является программа 1С, Парус, Галактика и т.п., однако для небольших организаций покупать подобного рода программы довольно накладно, а возможности, предоставляемые этими программами, зачастую значительно превосходят потребности небольших организаций. Выполнение любых действий в универсальном приложении влечет за собой повышенную нагрузку на пользователя. Применение программного средства, ориентированного на конкретную узкую задачу, позволяет сэкономить на средствах на лицензионное отчисление за использование приобретенного стандартного программного обеспечения.

## 1.3 Характеристика решаемой задачи

Полное наименование программного средства «Учет сотрудников организации».

Краткое наименование программного средства: «ПС учета сотрудников».

Заказчик: Учреждение образования «Могилевский государственный политехнический колледж», город Могилев. От имени Заказчика – А.В.Карманов.

Разработчик программного обеспечения (в дальнейшем Исполнитель) – Бесценный Сергей Петрович, учащийся по специальности 2-40 01 01, «Программное обеспечение информационных технологий».

Основанием для проведения работ является задание на курсовое проектирование.

Информационная система для учета сотрудников организации будет выполнять следующие функции:

* хранение информации о сотрудниках;
* хранение информации об начисленной им заработной плате;
* начисление заработной платы;
* поиск сотрудника в базе данных;
* формирование отчета о заработной плате.

Основными целями создания программного средства для учета являются:

* замещение ручного учета сотрудников организации, что уменьшит затраты времени на поиск и обработку информации;
* повышение качества выполняемой работы, что уменьшит риск ошибок при редактировании информации.

Система представляет собой базу данных, содержащую информацию о сотрудниках организации, их домашних адресах, телефонах, должностях, тарифных ставках и начисленной заработной плате.

Объектом автоматизации является рабочее место бухгалтера организации.

Видом автоматизируемой деятельности являются задачи учета сотрудников и их заработной платы.

Критерием оценки достижения целей создания программного средства является упрощение ведения учета сотрудников и контроля расходования фонда заработной платы организации.

Программное средство разрабатывается как однопользовательское программное обеспечение.

Наличие штата эксплуатационного и ремонтного персонала, ответственного за обеспечение функционирования информационных систем не требуется.

Структурой хранения данных будет являться собственная база данных. Клиентский интерфейс прикладной программы будет разрабатываться в Visual Studio 2019, поддерживающий интерфейс операционной системы Windows. Взаимодействие с пользователем будет осуществляться через диалоговые окна и главное окно приложения.

Система должна поддерживать целостность и сохранность базы данных, а также корректность вводимых данных и их наличие.

Требования к функциям выполняемой системы. Информационная система должна реализовывать следующие функции:

* добавлять данные о сотрудниках, должностях, тарифных ставках;
* производить редактирование информации о сотрудниках;
* формировать отчет по заработной плате;
* производить поиск сотрудника в базе данных;
* производить начисление заработной платы сотрудникам.

# 2 Проектирование программного продукта

## 2.1 Разработка модели данных

Функциональные требования к программному средству представлены на рисунке 1 в виде диаграммы вариантов использования.

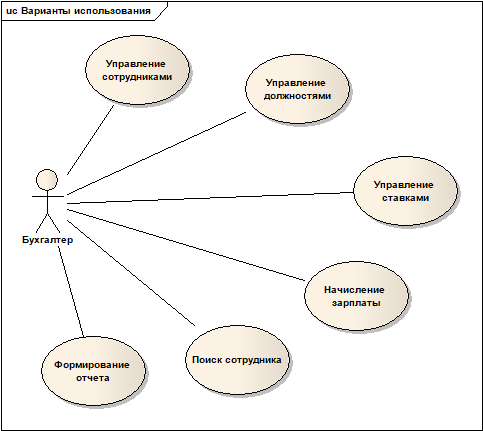


Рисунок 1 – Диаграмма вариантов использования

Описание вариантов использования представлено в таблице 1.

Таблица 1 – Реестр вариантов использования

|  |  |  |
| --- | --- | --- |
| Основной актер | Наименование варианта использования | Краткое описание |
| Бухгалтер | Управление сотрудниками | Ввод данных о сотрудниках |
| Бухгалтер | Управление должностями | Ввод данных о должностях |
| Бухгалтер | Управление ставками | Ввод данных о ставках |
| Бухгалтер | Начисление зарплаты | Начисление зарплаты исходя из должности и ставки |
| Бухгалтер | Поиск сотрудника | Поиск сотрудника по ФИО |
| Бухгалтер | Формирование отчета | Сформировать отчет по заработной плате |

Диаграмма классов форм приложения представлена на рисунке 2.

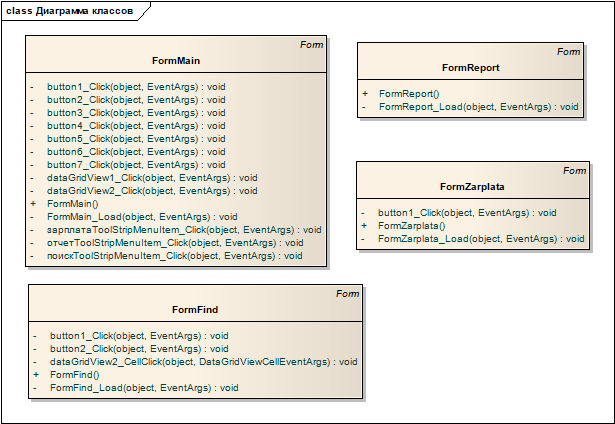


Рисунок 2 Диаграмма классов форм приложения

Диаграмма состояния приложения представлена на рисунке 3.

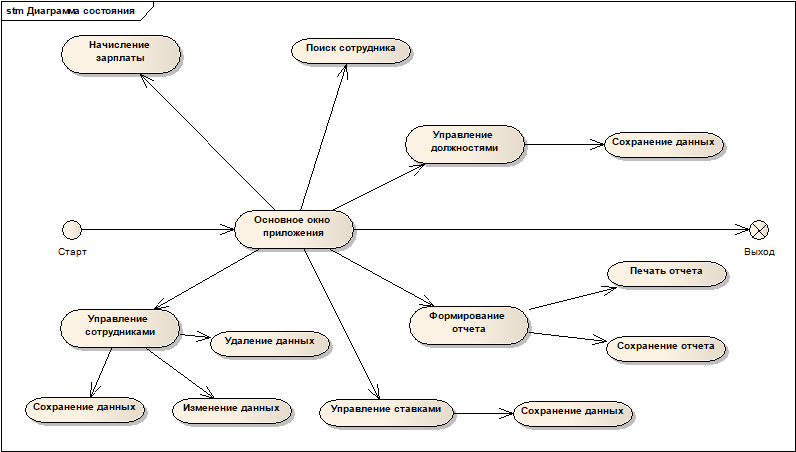


Рисунок 3. Диаграмма состояния приложения.

Диаграмма последовательности «Управление сотрудниками» представлена на рисунке 4.

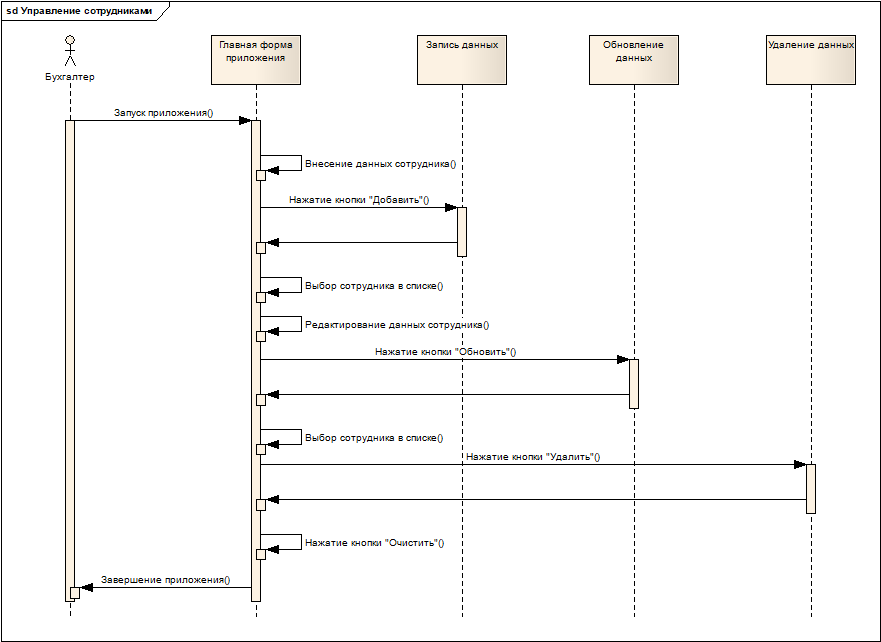


Рисунок 4. Диаграмма последовательности «Управление сотрудниками».

Таблица 2 - Описание процесса выполнения варианта использования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пользователь | Система | Экранная форма | Условие:  последующий шаг |
| 1 | Запуск программы | Запускается программа | Главная форма приложения | Ожидание действия |
| 2 | Внесение данных о сотруднике | Ожидание действия | Форма приложения «Автомобили» | Ожидание действия |
| 3 | Нажатие кнопки «Добавить» | Осуществляет сохранение данных в базе данных | Главная форма приложения | Ожидание действия |
| 4 | Нажатие кнопки «Обновить» | Осуществляет сохранение данных в базе данных | Главная форма приложения | Ожидание действия |
| 5 | Нажатие кнопки «Удалить» | Удаляет выделенную в табличной форме запись | Главная форма приложения | Ожидание действия |
| 6 | Нажатие кнопки закрытия главной формы программы или выбор пункта «Закрыть» главного меню | Завершает работу приложения |  |  |

Диаграмма последовательности «Управление ставками» представлена на рисунке 5.

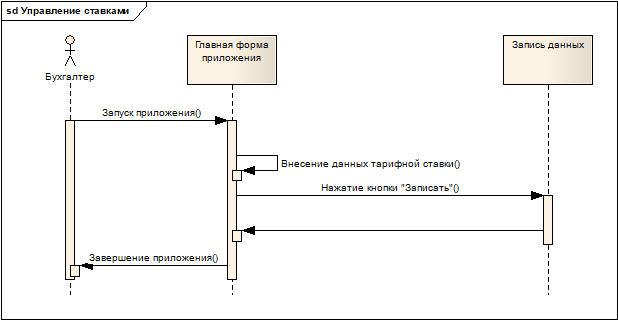


Рисунок 5. Диаграмма последовательности «Управление ставками».

Таблица 3 - Описание процесса выполнения варианта использования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пользователь | Система | Экранная форма | Условие:  последующий шаг |
| 1 | Запуск программы | Запускается программа | Главная форма приложения | Ожидание действия |
| 2 | Внесение данных о ставке | Ожидание действия | Главная форма приложения | Ожидание действия |
| 3 | Нажатие кнопки «Записать» | Осуществляет сохранение данных в базе данных | Главная форма приложения | Ожидание действия |
| 4 | Нажатие кнопки закрытия главной формы программы или выбор пункта «Закрыть» главного меню | Завершает работу приложения |  |  |

Диаграмма последовательности «Управление должностями» представлена на рисунке 6.

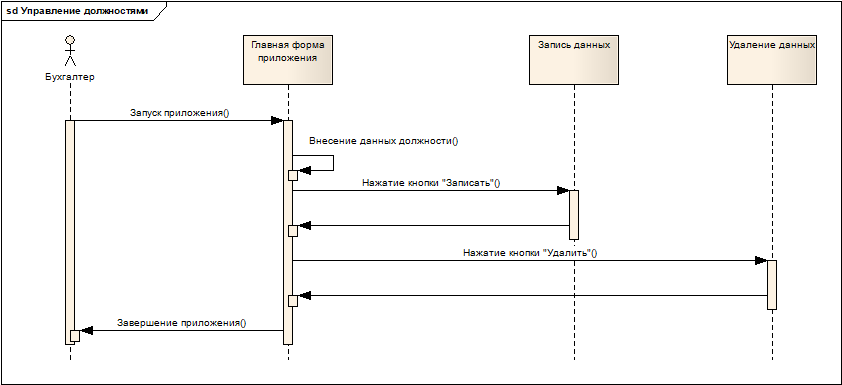


Рисунок 6. Диаграмма последовательности «Управление должностями».

Таблица 4 - Описание процесса выполнения варианта использования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пользователь | Система | Экранная форма | Условие:  последующий шаг |
| 1 | Запуск программы | Запускается программа | Главная форма приложения | Ожидание действия |
| 2 | Внесение данных о должности | Ожидание действия | Главная форма приложения | Ожидание действия |
| 3 | Нажатие кнопки «Записать» | Осуществляет сохранение данных в базе данных | Главная форма приложения | Ожидание действия |
| 4 | Нажатие кнопки «Удалить» | Удаляет выделенную в табличной форме запись | Главная форма приложения | Ожидание действия |
| 5 | Нажатие кнопки закрытия главной формы программы или выбор пункта «Закрыть» главного меню | Завершает работу приложения |  |  |

Диаграмма последовательности «Начисление зарплаты» представлена на рисунке 7.

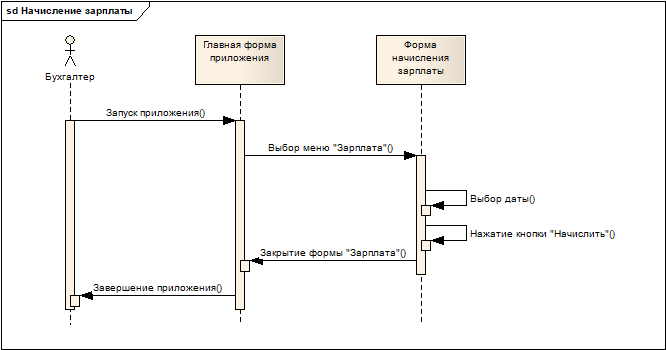


Рисунок 7. Диаграмма последовательности «Начисление зарплаты».

Таблица 5 - Описание процесса выполнения варианта использования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пользователь | Система | Экранная форма | Условие:  последующий шаг |
| 1 | Запуск программы | Запускается программа | Главная форма приложения | Ожидание действия |
| 2 | Выбор пункта меню «Зарплата» | Открывается форма «Зарплата» | Форма приложения «Зарплата» | Ожидание действия |
| 3 | Внесение данных о дате | Ожидание действия | Форма приложения «Зарплата» | Ожидание действия |
| 4 | Нажатие кнопки «Начислить» | Осуществляет сохранение данных в базе данных | Форма приложения «Зарплата» | Ожидание действия |
| 5 | Нажатие кнопки закрытия формы | Закрывает форму «Зарплата» | Главная форма приложения | Ожидание действия |
| 6 | Нажатие кнопки закрытия главной формы программы или выбор пункта «Закрыть» главного меню | Завершает работу приложения |  |  |

Диаграмма последовательности «Поиск сотрудника» представлена на рисунке 8.

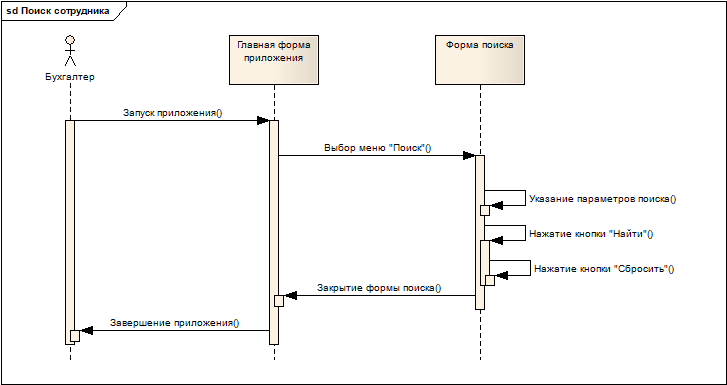


Рисунок 8. Диаграмма последовательности «Поиск».

Таблица 6 - Описание процесса выполнения варианта использования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пользователь | Система | Экранная форма | Условие:  последующий шаг |
| 1 | Запуск программы | Запускается программа | Главная форма приложения | Ожидание действия |
| 2 | Выбор пункта меню «Поиск» | Открывается форма «Поиск» | Форма приложения «Поиск» | Ожидание действия |
| 3 | Внесение данных о параметрах поиска | Ожидание действия | Форма приложения «Поиск» | Ожидание действия |
| 4 | Нажатие кнопки «Искать» | Осуществляет поиск в базе данных и отображает результаты в таблице | Форма приложения «Поиск» | Ожидание действия |
| 5 | Нажатие кнопки «Стереть» | Очищает поля ввода экранной формы | Форма приложения «Поиск» | Ожидание действия |
| 6 | Нажатие кнопки закрытия формы | Закрывает форму «Поиск» | Главная форма приложения | Ожидание действия |
| 7 | Нажатие кнопки закрытия главной формы программы или выбор пункта «Закрыть» главного меню | Завершает работу приложения |  |  |

Диаграмма последовательности «Формирование отчета» представлена на рисунке 9.

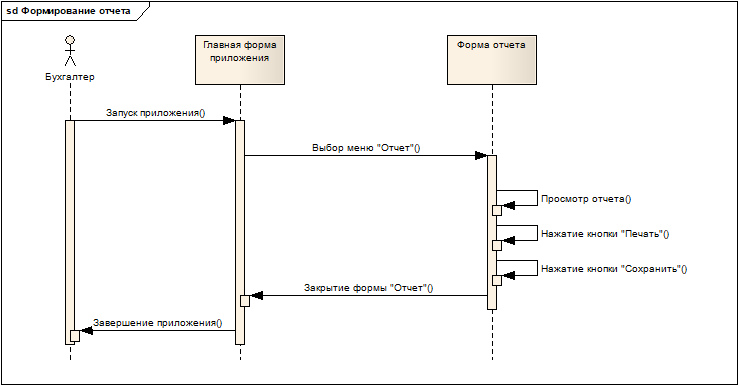


Рисунок 9. Диаграмма последовательности «Отчет».

Таблица 7 - Описание процесса выполнения варианта использования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пользователь | Система | Экранная форма | Условие:  последующий шаг |
| 1 | Запуск программы | Запускается программа | Главная форма приложения | Ожидание действия |
| 2 | Выбор пункта меню «Отчет» | Открывается форма «Отчет» | Форма приложения «Отчет» | Ожидание действия |
| 3 | Нажатие кнопки «Печать» | Осуществляет печать отчета на принтере | Форма приложения «Отчет» | Ожидание действия |
| 4 | Нажатие кнопки «Экспорт» | Осуществляет сохранение отчета на диск | Форма приложения «Отчет» | Ожидание действия |
| 5 | Нажатие кнопки закрытия формы | Закрывает форму «Отчет» | Главная форма приложения | Ожидание действия |
| 6 | Нажатие кнопки закрытия главной формы программы или выбор пункта «Закрыть» главного меню | Завершает работу приложения |  |  |

## 2.2 Выбор программного обеспечения

C# (произносится “Си-шарп”) — это язык программирования, предназначенный для разработки самых разнообразных приложений, предназначенных для выполнения в среде .NET Framework. Язык C# прост, типобезопасен и объектно-ориентирован. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую языкам C.

Visual C# — это реализация языка C# корпорацией Майкрософт. Поддержка Visual C# в Visual Studio обеспечивается с помощью полнофункционального редактора кода, компилятора, шаблонов проектов, конструкторов, мастеров кода, мощного и удобного отладчика и многих других средств. Библиотека классов .NET Framework предоставляет доступ ко многим службам операционной системы и другим полезным, правильным классам, что существенно ускоряет цикл разработки.

## 2.3 Определение требований к техническим средствам

Для разрабатываемой программы рекомендуется использовать аппаратно-программные средства со следующими характеристиками:

1) тип центрального процессора Intel Celeron (Pentium) или AMD Athlon:

- число ядер не менее 2, желательно от 4х и более;

2) системная плата с параметрами:

- форм-фактор - ATX;

- минимальный объем оперативной памяти - 4 Гб, желательно от 6 и более;

4) типы внешней памяти компьютера:

- минимальный объем жесткого диска: 120 ГБ;

6) параметры монитора:

- тип монитора - ЖК;

- ширина диагонали экрана – более 17 дюймов;

- разрешение по горизонтали и вертикали – не менее 1450x900;

Диаграмма компонентов представлена на рисунке 10.

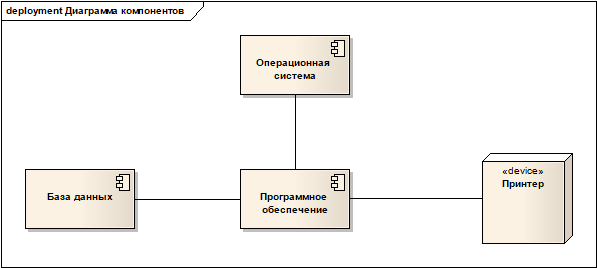


Рисунок 10 – Диаграмма компонентов

# 3 Реализация программного модуля

## 3.1 Определение формы представления входных и выходных данных

При разработке программного модуля была выбрана модель, при которой в главном окне приложения отображается общая информация по сотрудникам, а выполнение дополнительных действий, таких как просмотр и печать отчетов, поиск в базе данных и начисление заработной платы производятся в отдельных окнах приложения. На вспомогательных формах присутствуют поля ввода данных, поле отображения уже внесенной в таблицу информации, кнопки выполнения поиска, печати и сохранения отчета, начисления заработной платы.

В качестве выходных форм, помимо сводной главной формы приложения выступают формы поиска и отчета. Форма поиска позволяет осуществить быстрый поиск в базе данных записей сотрудников по части их ФИО, а форма отчета – сформировать отчет и вывести его, при необходимости, на печать. Вид форм и отчетов размещен в приложении Б.

## 3.2 Тестирование программного средства

Результаты тестирования приложения в различных режимах работы представлены в виде таблицы (таблица 8).

Таблица 8 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Примечание |
| Главная форма приложения | | | |
| Запуск приложения | Отображение главной формы приложения с заполненными таблицами из базы данных | Выполнено |
| Выбор сотрудника в таблице сотрудников | Отображение в связанных таблицах соответствующей информации о сотруднике | Выполнено |
| Выбор пункта меню для начисления заработной платы | Открытие соответствующей действию экранной формы начисления заработной платы сотрудникам | Выполнено |
| Выбор пункта меню «Отчет» | Открытие формы отчета, содержащей сформированный отчет по путевкам | Выполнено |
| Выбор пункта меню «Поиск» | Открытие формы поиска | Выполнено |
| Выбор пункта меню «Справка» | Открытие справочной системы приложения | Выполнено |
| Форма работы с заработной платой | | | |
| Нажатие кнопки «Начислить» | Расчет и внесение записей в базу данных | Выполнено |
| Форма поиска | | | |
| Нажатие кнопки «Поиск» | Вывод результатов поиска в табличной части формы, в соответствии с указанными критериями | Выполнено |
| Нажатие кнопки «Стереть» | Очистка полей ввода поисковой строки | Выполнено |
| Нажатие кнопки закрытия формы | Закрытие текущей формы и возврат в главную форму приложения | Выполнено |
| Форма отчета | | | |
| Нажатие кнопки «Печать» | Вывод отчета на притер | Выполнено |
| Нажатие кнопки «Экспорт» | Сохранение отчета в указанном формате | Выполнено |
| Нажатие кнопки закрытия формы | Закрытие текущей формы и возврат в главную форму приложения | Выполнено |

## 3.3 Защита информации

Защита программы от несанкционированного доступа необходима для сохранения личных данных сотрудников, таких как их адреса, телефоны и размер заработной платы. Для осуществления подобного рода защиты существуют различные способы. Одним из них является ограничение свободного доступа к рабочему месту бухгалтера, вторым – защита самого файла базы данных паролем известным только нужным пользователям системы.

Защита программы заключается в защите доступа к базе данных и осуществляется при необходимости соответствующей настройкой самого файла базы данных. С целью ограничения несанкционированного доступа при запуске программы производится запрос имени пользователя и его пароля. При некорректном вводе логина и пароля, система выдает сообщения об ошибке.

## 3.4 Разработка справочной системы

В проекте используется справочная система в формате CHM. Из её достоинств можно назвать ее надежность и меньшие, чем у HTML-help, требования к ресурсам. Существуют различные программы для создания справочных систем названных типов. Для создания справочной системы использовалось программное обеспечение HTML Help WorkShop фирмы Microsoft.

Справочная система кратко описывает пользователю основные принципы работы с программной, предоставляет ее описание и особенности. Справочная система приложения вызывается нажатием клавиши F1 на клавиатуре, либо выбором в главном меню приложения. При вызове справочной системы открывается справочное окно, состоящее из 2х частей – в левой можно выбрать интересующий раздел описания программы, в правой отображается информация об этом разделе, либо приветственное окно при первом запуске.

# 4Энерго- и ресурсосбережение

Стандарт управления энергопотреблением компьютеров, описывает три различных режима работы компьютера: ждущий, спящий, основной. Эти режимы отличаются потребляемой мощностью электроэнергии.

Ждущий режим позволяет экономить энергию, за счет отключения периферийных устройств. Спящий режим позволяет экономить электроэнергию на 100%, так как отключаются все устройства кроме ЦПУ. При работе в основном режиме экономии нет. Потребляемая мощность около 400 Ватт.

Для нахождения количества рабочих дней, в течение которых разрабатывался программный продукт, используем формулу

(1)

где Тпк – время работы компьютера, ч;

Тпк =260,1 ч;

tрп – суммарное время регламентированных перерывов, в течение

рабочего дня, ч;

tрп =50 мин=5/6 часа.

Для нахождения суммарной продолжительности регламентированных перерывов в течение всего времени разработки программного модуля

(2)

где Трп – суммарная продолжительность регламентированных перерывов в течение всего времени разработки программного модуля, ч.

Экономия электроэнергии рассчитывается по формуле

(3)

где Wпк – потребляемая мощность ПК, кВт;

Wсп – потребляемая мощность компьютера в ждущем режиме, кВт;

Wпк = 0,4 кВт;

Wсп = 0,16 кВт;

Сэ – стоимость 1 кВт электроэнергии, руб.

Таким образом сумма сэкономленной электроэнергии за время разработки программного обеспечения составила 2,24 рубля.

# Заключение

Результатом выполнения данного курсового проекта является разработанное программное средство для автоматизации учета сотрудников, их координат и заработной платы, позволяющее хранить данные в базе данных и осуществлять управление, поиск информации и печать отчетов.

Программа выполнена в соответствии с техническим заданием.

Достоинствами данной программы являются:

* удобный и простой интерфейс;
* возможность редактирования, добавления, удаления, изменения данных.

Разработанная программа нетребовательна к ресурсам компьютера, обладает интуитивно-понятным интерфейсом, проста и удобна в использовании.

# Список используемых источников

1. ГОСТ ИСО/ МЭК 2382-99. Информационные технологии. Словарь. Ч. Основные термины.
2. ГОСТ 34.602 – 89 Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы
3. СТУ СМК 4.04-2011. Стандарт учреждения. Общие требования к оформлению текстовых документов».
4. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон; Пер. с англ. Мухин Н. – 2-е изд. – М.: ДМК Пресс, 2007. – 496 с.
5. Рамбо, Дж. UML 2.0. Объектно-ориентированное моделирование и разработка / Рамбо Дж., М. Блаха. – 2-е изд. – СПб.: Питер, 2007. – 554 с. (Библиотека программиста)
6. Фролов, А. В. Язык С#. Самоучитель. / Г. В. Фролов, А. В. Фролов - М.: ДИАЛОГ-МИФИ, 2003. - 560 с.
7. Шилдт Г. С# 4.0: полное руководство / Г. Шилдт. – М.: ООО «И.Д. Вильямс», 2011. -1056 с.: ил.