Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Институт повышения квалификации и переподготовки

руководящих работников и специалистов

по информационным технологиям и радиоэлектронике

Кафедра «Микропроцессорные системы и сети»

**К защите допустить**

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Полубок

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2015 г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К ДИПЛОМНОМУ ПРОЕКТУ**

**НА ТЕМУ:**

«Программное средство автоматизации и учета персонала предприятия»

Дипломник Г.Г. Алексеюк

(дата, подпись)

Руководитель Н.Л. Боброва

(дата, подпись)

Консультанты:

по ЕСПД и ЕСКД В.Н. Мухаметов

(дата, подпись)

по специальности Н.Л. Боброва

(дата, подпись)

Рецензент /\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись) (И.О. Фамилия)

МИНСК 2015

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»**

**Институт повышения квалификации и переподготовки руководящих работников и специалистов**

**по информационным технологиям и радиоэлектронике**

**Кафедра Микропроцессорные системы и сети**

Специальность I-40 01 73 – Программное обеспечение информационных систем

*У Т В Е Р Ж Д А Ю*

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_ Полубок В.А.

(Подпись, Ф.И.О.)

" \_\_\_\_ " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2015 г.

**ЗАДАНИЕ**

**по дипломному проектированию**

Слушателю *Алексеюку Геннадию Геннадьевичу*

1.Тема проекта *«Программное средство автоматизации и учета персонала предприятия»*

утверждена приказом по Институту № *120-C* от *02.07.2015* г.

2. Сроки сдачи слушателем законченного проекта 26.09.2015

3. Исходные данные к проекту *Рабочие места находятся в пределах одного здания ОАО «Кузлитмаш»;*

*каждому сотруднику соответствуют определенные для его должности права доступа.*

*Реализовать возможность автоматизации и учета персонала предприятия. Обеспечить вывод на печать*

*отчетов и ведомостей сотрудников в требуемом для предприятия формате.*

*В качестве языка программирования использовать C#. В качестве базы данных MS SQL Server 2008R2*

4. Перечень подлежащих разработке вопросов:

*1. Анализ существующих решений по теме проекта и постановка задачи;*

*2. Модели и методы, положенные в основу проекта;*

*3. Разработка проекта программного обеспечения(формирование функциональных требований, разработка*

*структуры, разработка схемы данных реляционной модели, разработка основных алгоритмов);*

*4. Разработка и реализация программного кода;*

*5. Тестирование разработанного программного обеспечения.*

5. Перечень графического материала (с точным указанием обязательных чертежей и графиков):

*1. Схема ресурсов системы, ф. А3, чертеж;*

*2. Схема работы системы, ф.А3, чертеж;*

*3. Создание штатного расписания, ф. А3, чертеж;*

*4. Добавление нового сотрудника, ф. А4, чертеж;*

*5. Заполнение личной карточки сотрудника, ф. А3, чертеж*

*6. Формирование структуры предприятия, ф. А3, чертеж*

*7. Перевод сотрудника, ф. А3, чертеж*

*8. Увольнение сотрудника, ф. А3, чертеж*

6. Консультанты по проекту (с указанием относящихся к ним разделов проекта)

*по ЕСПД, ЕСКД Мухаметов В.Н.*

*по специальности Боброва Н.Л.*

7. Дата выдачи задания 02.07.2015 г.

8. Календарный график работы над проектом на весь период проектирования

*13.07.2015 г. – 25 %*

*30.07.2015 г. – 50 %*

*22.08.2015 г. – 75 %*

*12.09.2015 г. – 100 %*

Руководитель (подпись, Ф.И.О., должность) \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Боброва Н.Л. старший преподаватель МПСС ИИТ БГУИР*

Задание принял к исполнению (дата и подпись слушателя)

Содержание

[Введение 4](#_Toc436064364)

[1. Аналитический обзор и постановка задачи 6](#_Toc436064365)

[1.1. Обзор программных аналогов, их достоинств и недостатков 6](#_Toc436064366)

[1.2. Постановка задачи 16](#_Toc436064367)

[2 Методы и модели положенные в основу проекта 18](#_Toc436064368)

[3. Проектирование системы учета персонала предприятия 30](#_Toc436064369)

[3.1. Функциональный анализ предметной области 30](#_Toc436064370)

[3.2. Разработка структуры и схемы работы объекта 34](#_Toc436064371)

[3.3. Разработка базы данных 35](#_Toc436064372)

[3.4. Разработка интерфейса системы 43](#_Toc436064373)

[4. Разработка основных алгоритмов 45](#_Toc436064374)

[4.1. Общая схема работы автоматизации учета персонала 45](#_Toc436064375)

[4.2. Алгоритм добавления нового сотрудника 45](#_Toc436064376)

[4.3. Алгоритм заполнения личной карточки сотрудника 47](#_Toc436064377)

[4.4. Алгоритм формирования структуры должности/подразделения 49](#_Toc436064378)

[4.5. Алгоритм создания штатного расписания 51](#_Toc436064379)

[4.6. Алгоритм перевода сотрудника 52](#_Toc436064380)

[4.7. Алгоритм увольнения сотрудника 54](#_Toc436064381)

[5. Тестирование приложения. 56](#_Toc436064382)

[5.1. Общие положения 56](#_Toc436064383)

[5.2. Разработка тестов и результаты тестирования приложения 56](#_Toc436064384)

[Заключение 66](#_Toc436064385)

[Список использованных источников 68](#_Toc436064386)

[Приложение А Программный код добавления данных приказа о приеме на работу 69](#_Toc436064387)

[Приложение Б Программный код заполнения личной карточки сотрудника 71](#_Toc436064388)

[Приложение В Программный код формирования структуры должности/подразделения 72](#_Toc436064389)

[Приложение Г Программный код создания штатного расписания 73](#_Toc436064390)

[Приложение Д Программный код приказа о переводе сотрудника 75](#_Toc436064391)

[Приложение Е Программный код приказа об увольнении сотрудника 77](#_Toc436064392)

Введение

В настоящее время, в мире непрерывного потока информации, все большую актуальность приобретают процессы автоматизации работ рутинного характера и работ, трудно поддающихся формализации. В связи с этим ежегодно разрабатываются программы, призванные решить возникающие проблемы.

Создание информационной системы предполагает, что основные операции по накоплению, хранению и переработке информации возлагается на вычислительную технику, специалист же выполняет только определенную часть ручных операций и операций, требующих творческого подхода при подготовке управленческих решений. Вычислительная техника при этом работает в тесном взаимодействии с пользователем, который контролирует ее действия, меняя значения отдельных параметров в ходе решения задачи, а также вводит исходные данные для решения задач и функций управления.

Успех работы любого предприятия зависит от того, насколько эффективной будет работа отдела кадров. Задача работников отдела кадров найти и удержать нужных специалистов. Укомплектовать предприятие рабочими с соответствующими профилю специальностями и квалифицированными специалистами. Отдел кадров должен правильно распределить трудовые ресурсы, обеспечить карьерный рост персонала, подготовить резерв для выдвижения работников, обеспечить повышение квалификации персонала. Работа отдела кадров крупной организации связана с накоплением большого количества информации о личных данных сотрудников. Традиционно информация хранится на бумажных носителях. При этом трудно осуществить быстрый отбор нужных данных при приеме на работу, уходе в отпуск, увольнении, переходе на другую должность.

Автоматизированная система управления персоналом является источником, из которого можно получить информацию различного характера для всех заинтересованных лиц, в том числе руководства, специалистов отделов, контролирующих органов.

Сегодня на рынке программного обеспечение присутствует много различных систем по управлению кадрами. Даже опытному специалисту трудно не растеряться и не запутаться в этом разнообразии. Поэтому, немаловажным является изучение аналогичных программных средств. Следует отметить, что при приобретении готового программного продукта в любом случае потребуется серьезная настройка программы под конкретные требования, а также обязательное дальнейшее программное сопровождение, и обучение персонала, что выразится в ощутимых финансовых и временных затратах.

Целью дипломного проекта является автоматизация системы учета персонала предприятия, что приведет к повышению эффективности работы сотрудника отдела кадров.

Объектом исследования выступает ОАО «Кузлитмаш», а предметом исследования – кадровые процессы, методы и системы контроля и учета рабочего времени.

Для достижения поставленной цели были сформулированы следующие задачи:

* исследовать существующие бизнес-процессы кадрового учета, методы и системы учета и контроля рабочего времени;
* провести проектирование системы учета персонала предприятия;
* сделать функциональный анализ предметной области;
* разработать структуру и схему работы объекта;
* разработать базу данных;
* разработать интерфейс системы;
* разработать основные алгоритмы;
* выполнить проектирование и реализацию системы кадрового учета.

Таким образом, тема данной дипломного проекта является актуальной с практической точки зрения.

1. Аналитический обзор и постановка задачи
   1. Обзор программных аналогов, их достоинств и недостатков

Программный продукт по автоматизации учета персонала позволяет оперативно получать аналитическую информацию и принимать обоснованные управленческие решения. Автоматизированное хранение и обработка полной кадровой информации также позволяет эффективно осуществлять подбор и перемещение сотрудников, создавая тем самым основу для анализа и планирования затрат по персоналу. Кроме того, используется для автоматизированного расчета заработной платы с учетом информации об изменении штатного расписания, отпусках, больничных, командировках, льготах и взысканиях, что дает возможность работникам бухгалтерии точно и оперативно формировать бухгалтерские отчеты, и своевременно относить затраты на себестоимость. Обеспечивает ведение учета в соответствие с законодательством, чем минимизирует риск санкций со стороны контролирующих органов.

Раньше использование автоматизированных систем управления считалось привилегией крупных предприятий. Поэтому масштаб и цены этих систем были соответствующими, а малым и средним компаниям большинство продуктов были не по карману. И хотя ныне практически все поставщики предлагают системы малого и среднего класса, рынок все еще далек от насыщения. Программы кадрового учета нужны, прежде всего, специалистам по кадровому делопроизводству. Такие системы служат для решения двух задач: автоматизации оформления документов и расчета заработной платы. Кроме того, многие системы также позволяют сохранять информацию о семейном положении сотрудника и воинском учете. Информация о сотруднике будет неполной, если не хранить данные об истории его работы, о переводах на новые должности, поощрениях и взысканиях. Обычно возможность хранения подобных сведений называют возможностью отслеживания движения сотрудников, хранения организационной структуры и штатного расписания.

Учет рабочего времени сотрудников позволяет регистрировать действия сотрудника, автоматически выставляет нарушения, считает фактически отработанное время, составляет личные табели, отражает оперативную информацию. Также разработчики часто добавляют возможность регистрации в системе командировок, больничных и отпусков. Ну и, конечно, намного полезнее будет система, которая умеет выводить всю хранящуюся в ней информацию в виде стандартных форм, сформированных по правилам современного делопроизводства. Поэтому, немаловажным будет изучение аналогичных программных средств.

С появлением первых средств автоматизации были разработаны и первые программы учета и управления персоналом. Предприятие или организация, имевшие собственный отдел АСУП пользовались программами учета персонала собственной разработки. Эти программы опирались на различные аппаратные платформы (начиная от мэйнфреймов и заканчивая ПК) и инструментальные средства (начиная от PL-1 и заканчивая Clipper и FoxPro). С появлением новых, более совершенных, аппаратных и инструментальных средств, данные программы модифицировались и расширяли свою функциональность. При этом если на начальном этапе развития HR-систем можно было говорить лишь о локальных программах управления персоналом, то уже с появлением первых ERP-систем модули управления персоналом были органично интегрированы в них, а некоторые, наиболее удачные разработки, продолжили свое распространение и автономно, не потеряв при этом возможности интегрироваться с ERP-системами. Существующие в настоящее время на рынке автоматизированные системы управления персоналом, по их функциональной направленности можно разделить на следующие основные группы [1]:

* многофункциональные экспертные системы, позволяющие проводить профориентацию, отбор, аттестацию сотрудников предприятия;
* экспертные системы для группового анализа персонала, выявления тенденций развития подразделений и организации в целом;
* программы расчета зарплаты;
* комплексные системы управления персоналом, позволяющие формировать и вести штатное расписание, хранить полную информацию о сотрудниках, отражать движение кадров внутри фирмы, рассчитывать зарплату.

Комплексные системы управления персоналом используются для автоматизации кадровой работы на любом предприятии. В первую очередь, такие системы необходимы руководству для получения оперативной информации по любому вопросу, связанному со структурой предприятия, штатным расписанием, вакансиями и информацией о сотрудниках. Оперативно принять правильное решение сможет лишь руководитель, способный быстро оценить сложившуюся ситуацию на основе анализа актуальной информации о состоянии дел на предприятии. Поэтому немаловажным фактором условий использования HR-систем является также возможность интеграции системы кадрового учета с системами бухгалтерского учета и управления предприятием.

В настоящее время на рынке наблюдается многообразие предложений по разработке и поставке автоматизированных систем управления персоналом, как отечественных, так и зарубежных. К достоинствам отечественных пакетов можно отнести их адаптацию к системе учета и делопроизводства, а также более низкую цену по сравнению с наиболее известными пакетами зарубежных фирм. К преимуществу зарубежных пакетов относится в некоторых случаях значительно более полная функциональность.

Подавляющее большинство комплексных информационных систем (КИС) ) построены по модульному принципу и имеют в своем составе модуль управления персоналом, реализующий автоматизированное управление кадрами Часто управление кадрами объединено с расчетом зарплаты. Можно назвать такие известные в мире системы, имеющие в своем составе Human Resources (HR) модули, как SAP R/3, Baan, Oracle Applications и др. Cуществуют и автономные программные пакеты управления персоналом, одним из примеров которых является ПО Renaissance CS Human Resources.

Принято считать, что ощутимый эффект от внедрения HR-систем получают предприятия, с численностью персонала болеет 1000 человек. Западные разработчики КИС утверждают, что внедрение HR-модулей позволит предприятию получить организационные, экономические и социальные эффекты. Следует отметить, что эти эффекты присущи всем современным системам управления персоналом, в том числе и наиболее продвинутым отечественным HR-системам. Организационные эффекты заключаются в следующем [1]:

* сокращении времени принятия решений на всех уровнях управления предприятием;
* повышении качества кадровых решений;
* оперативности подготовки отчетности для органов государственного управления в соответствии с законодательными и нормативными требованиями.

В свою очередь, влияние экономических эффектов от внедрения HR-модулей позволяет [1]:

* снизить затраты на управление персоналом;
* повысить производительность труда персонала;
* оптимально использовать профессиональные качества конкретного сотрудника предприятия.

Социальный эффект от внедрения HR-модулей состоит в следующем:

* ведении полной индивидуальной трудовой истории персонала предприятия;
* подготовке руководящего резерва и продвижении по службе наиболее перспективных сотрудников предприятия.

Западные разработчики вложили в создание данных НR-модулей огромные человеческие и материальные ресурсы, а также накопили опыт их внедрения на сотнях и тысячах предприятий. В то же время, существуют определенные ограничения для распространения этих модулей на отечественных предприятиях. Основными из этих ограничений являются:

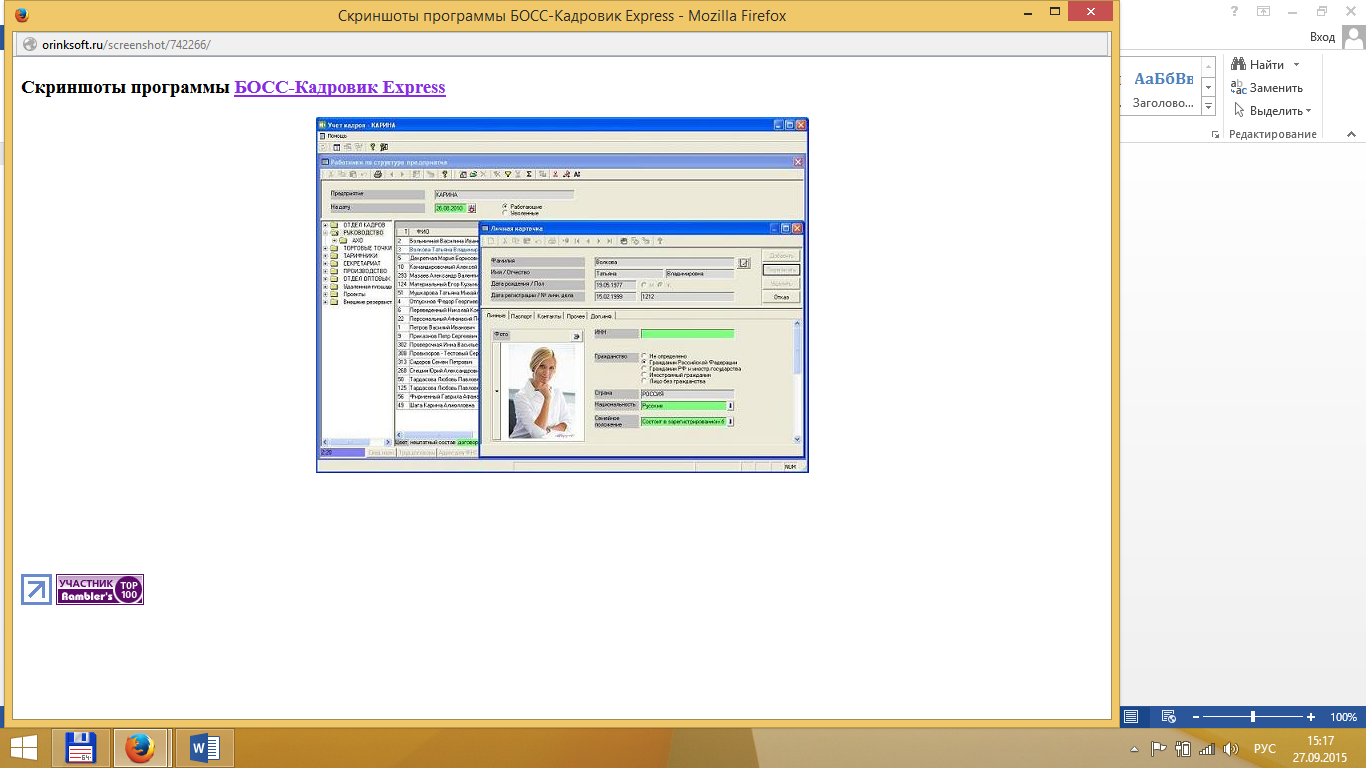
* высокая цена внедрения и поддержки
* функциональная избыточность

Рассмотрим ряд программных средств в области учета и контроля рабочего времени и реализуемые ими методики.

*«БОСС–Кадровик».* Одной из наиболее распространенных автоматизированных систем управления персоналом является БОСС-Кадровик, разработанная и успешно продвигаемая компанией IT.

"БОСС-Кадровик" относится к классу комплексных систем управления персоналом. Он позволяет формировать и вести штатное расписание, хранить полную информацию о сотрудниках, отражать движение кадров внутри фирмы, рассчитывать зарплату. Этот программный продукт может использоваться для автоматизации работы отдела кадров, планово-экономического и расчетного отделов на любом предприятии независимо от рода деятельности.

Дружественный интерфейс облегчает ввод информации и обучение работе. Логика системы понятна любому, даже начинающему пользователю. Экранный вид всех документов приближен к стандартному. Основным действиям в системе соответствует стандартный набор функциональных клавиш. Если вы случайно сделали что-нибудь некорректное, то "БОСС-Кадровик" сам предупредит вас и предложит вариант действий.

Рисунок 1.1 – Главное окно программы «БОСС – Кадровик»

Рутинные операции в системе сведены к минимуму. При заполнении любого документа практически всю необходимую информацию можно найти в справочниках. Использование фильтров позволит выделять из списков самые замысловатые выборки. Многие параметры заполняются по умолчанию с возможностью корректировки и замены.

В состав системы "БОСС-Кадровик" входят модули "Учет кадров", "Штатное расписание" и "Расчет заработной платы"; в зависимости от потребностей конкретного предприятия в систему включается отдельный модуль учета табеля рабочего времени, предназначенный для ведения ежедневного табельного учета непосредственно в подразделениях.

"БОСС-Кадровик" является сетевым продуктом, реализованным в архитектуре клиент-сервер, поэтому система отличается высокой надежностью, способна восстанавливаться при сбоях аппаратуры и обеспечивать надежную защиту от несанкционированного доступа.

Для работы с базой данных используется сервер Microsoft SQL Server. Применение данного SQL-сервера гарантирует максимальное использование сервером всех возможностей операционной системы. MS SQL Server поддерживает широкий спектр средств разработки, позволяет интегрировать программный продукт с офисными приложениями такими, как Microsoft Word, Еxcel, Internet Explorer.

*«АиТ: Управление персоналом».* Данный программный комплекс создан на базе изучения некоторых западных систем, в частности, модулей HR-системы SyteLine (SYMIX) и модуля «Персонал» в Oracle Aplications (Oracle)), а также ряда других разработок (группы компаний БИГ, РОЭЛ-Консалтинг, Психология и Бизнес.

Программный комплекс «АиТ: Управление персоналом» создан по модульному принципу и состоит из следующих модулей [1]:

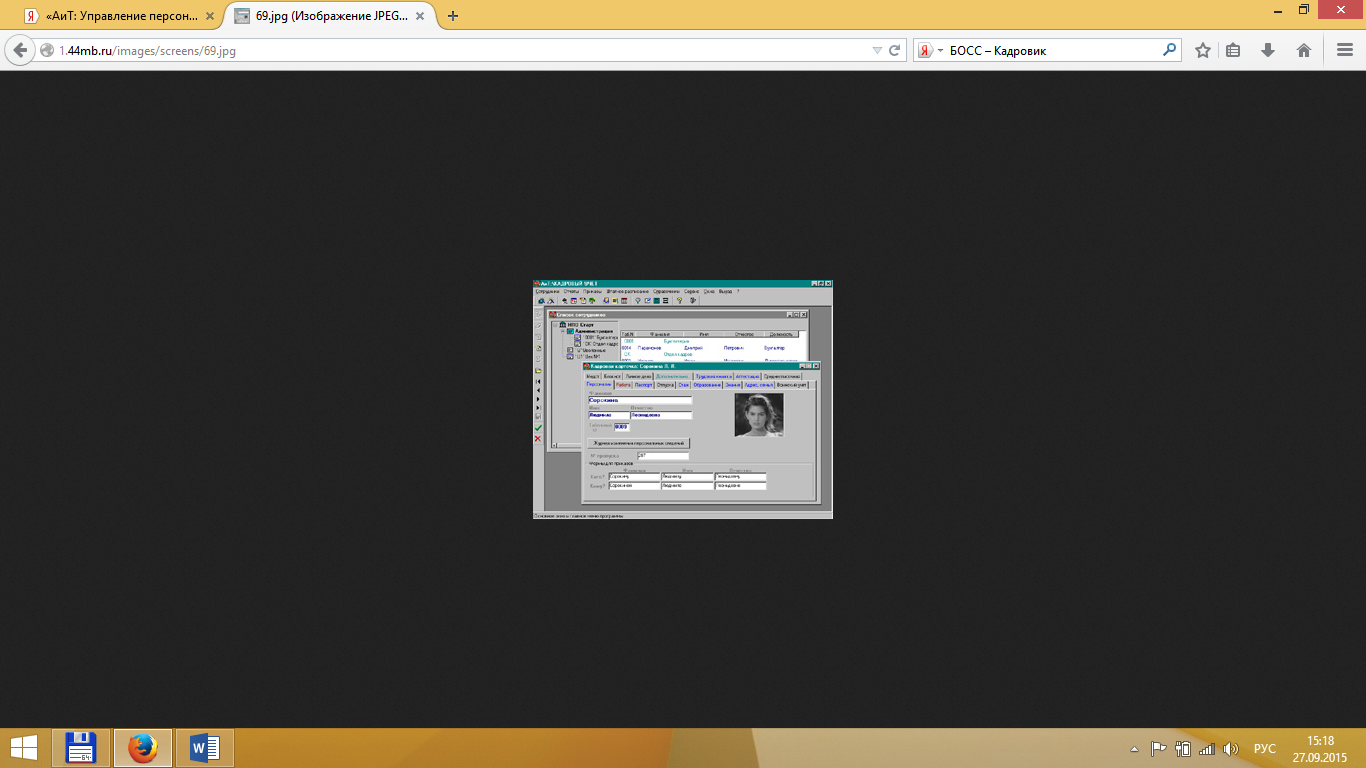
* заработная плата;
* кадровый учет;
* табельный учет;
* персонифицированный пенсионный учет;
* конфигурация системы;
* учет коллективных и бригадных работ.

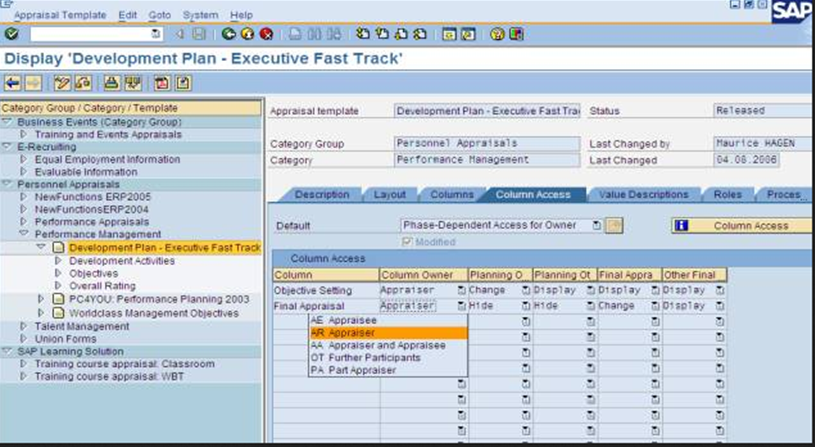
Рисунок 1.2 – Главное окно программы «АиТ: Управление персоналом»

Для решения других специфических задач кадрового менеджмента, например, аттестации и оценки персонала, могут использоваться дополнительные модули или специализированные продукты третьих производителей.

«АиТ: Управление персоналом» функционирует на единой клиент-серверной базе данных как автономно, так и в комплексе с различными финансовыми системами и системами управления предприятием (начиная от Platinum SQL и заканчивая ПО Парус). Существуют 2 версии комплекса: MSDE. Многопользовательская клиент-серверная версия до 5 рабочих мест. Она рекомендуется для организаций с численностью до 500 сотрудников). Версия MS SQL, предназначенная как для малых предприятий, так и для крупных корпораций с разветвленной структурой управления, развитой системой материального стимулирования, большой численностью персонала, возможно, разобщенного территориально.

*SAP Human Resources Management System*. Системы управления персоналом разработки компании SAP достаточно широко распространены в мире. Систему SAP Human Resources в настоящее время используют более 5000 компаний (в более чем 35 странах мира) с общим числом сотрудников более 4,6 млн. В состав КИС SAP R/3 (также разработка компании SAP) входит модуль управления персоналом SAP Human Resources Management System (SAP HR), содержащий следующие основные компоненты [1]:

* администрирование персонала и расчет зарплаты;
* управление данными кандидатов;
* управление основными данными;
* командировочные расходы.
* планирование и профессиональный рост персонала;
* описание рабочего времени и штатного расписания;
* предварительный отбор кандидатов;
* квалификации и квалификационные требования;
* управление повышением квалификации;
* планирование карьеры и замещения должностей преемниками;
* планирование расходов на содержание персонала;
* планирование использования персонала;
* планирование рабочего времени.

Рисунок 1.3 – Главное окно программы SAP Human Resources

Система управления персоналом SAP HR предназначена для применения на средних и крупных предприятиях. Модульная структура системы позволяет внедрять ее поэтапно. В SAP HR реализованы единые система отчетов и дружественный графический интерфейс пользователя. Кроме того, в SAP HR возможна связь с общими организационно-экономическими функциями (например, результаты расчета зарплаты сразу же передаются в бухгалтерию). Достоинством системы SAP HR является возможность ее интеграции с различными приложениями для настольных ПК (MS Word, MS Excel и др.). Помимо этого, в SAP HR реализованы современные технологии.

Существует возможность размещения вакансий в Интернет (как для всего мира, так и только для определенных регионов). После заполнения анкеты претендентами на заполнение вакансии, данные кандидатов переносятся непосредственно в модуль «Управление данными кандидатов». С помощью каталога «Кто есть кто» сотрудники предприятия могут в любое время найти необходимое им контактное лицо, а с помощью приложения «SAP Самообслуживание сотрудника» они могут самостоятельно просматривать свои персональные данные и корректировать их (например, изменять адрес место жительства).

Управление потоками бизнес-операций, поддерживающее целенаправленную поэтапную передачу информации в интегрированной среде без многократного ввода данных, анализ и отслеживание статуса обработки информации, наглядное и соответствующее рабочему месту представление открытых и подлежащих выполнению действий. Объединение отдельных функций в единые процессы позволяет оптимизировать расходы на координацию действий, а также сократить временные затраты на выполнение операций.

Используется архивирование данных по персоналу на оптических носителях. Так же реализована связь с платежными системами расчета по кредитным картам (например, в рамках управления данными по командировкам) и др.

*Oracle Human Resources Analyzer.* Корпорация Oracle для управления персоналом средних и крупных предприятий предлагает систему Oracle Human Resources Analyzer. C ее помощью можно автоматизировать [1]:

* табельный учет на предприятии;
* процедуру найма персонала предприятия;
* учет профессионально-важных свойств и деловых качеств сотрудников предприятия;
* учет потребностей современного обучения персонала;
* планирование развития карьеры сотрудников предприятия;
* разработку компенсационной политики для персонала предприятия.

Система Oracle Human Resources Analyzer также может внедряться помодульно. В ее состав входят следующие основные модули:

* кадры;
* зарплата;
* табельный учет.

В частности, в модуле «Кадры» реализованы следующие задачи:

* персональный учет;
* планирование организации;
* планирование выплат и компенсаций;
* анализ и ведение отчетности.

В системе Oracle HR Analyzer возможно использовать как типовые, так и нестандартные формы отчетности. Нестандартные формы отчетности можно создавать с помощью средства разработки приложений в среде Oracle – Oracle Discoverer. Кроме того, система Oracle HR Analyzer располагает средствами оперативного анализа данных и поддержки принятия решений на базе технологии OLAP (On-line Analytical Processing).

Компания Oracle выпустила ПО Oracle Project Resource Management (PJRM), которое предназначено, для управления персоналом предприятия. Данное ПО представляет собой полную систему управления персоналом, предлагаемую в качестве составной части комплекта программных средств Oracle E-Business Suite. полностью интегрировано с ПО Oracle Projects, Oracle Human Resource Management и другими компонентами E-Business Suite, а также может быть развернуто автономно.

После проведенного анализа приведённых выше программных средств, их функционал можно разделить на несколько групп.

Учет кадров: Учет персонала, оформление кадровых документов, приказов о приеме, увольнении и перемещении; учет поощрений, взысканий, социальных льгот и услуг по страхованию; ведение личных карточек сотрудников, учет аттестации и переподготовки кадров.

Табельный учет: Поддержка работы с сотрудниками одного или нескольких подразделений, ведение табеля учета рабочего времени, оформление больничных листов, отпусков и командировок.

Штатное расписание: Формирование организационной структуры предприятия, составление штатного расписания подразделений.

Расчет заработной платы: Расчет заработной платы по отработанному времени, работа с нарядами, расчет основных налогов; выплата авансов и расчет внешних удержаний.

Формирование отчетности: Формирование списка работников, по которым необходимо предоставить отчетность и персональных данных о работниках предприятия за отчетный период.

С учётом анализа рынка автоматизированных систем управления персоналом и взяв во внимание специфику предприятия, можно сделать следующий вывод: предлагаемые на рынке автоматизированные системы управления персоналом, либо недостаточно гибки, либо обладают излишней мощностью.

При этом в любом случае требуется серьезная настройка программы под конкретные требования, а также обязательное дальнейшее программное сопровождение, что выразится в ощутимых финансовых и временных затратах.

Поэтому, для предприятия имеющего определенные финансовые трудности, целесообразно создать новую АС, способную ускорить и оптимизировать трудовой процесс и быть приемлемой по цене.

* 1. Постановка задачи

На ОАО «Кузлитмаш» работа отдела кадров не была автоматизирована. Все операции выполнялись вручную, что вело к большому количеству ошибок, значительным временным и финансовым затратам. Поэтому была поставлена задача создания автоматизированной системы, которая решала бы эти проблемы. Автоматизация отдела кадров должна избавить от выполнения рутинных операций при работе с кадрами, подготовке и учете приказов. Автоматизированное хранение и обработка полной кадровой информации также позволяет эффективно осуществлять подбор и перемещение сотрудников. Автоматизация учета персонала предприятия позволяет обеспечить увеличение производительности труда, уменьшение времени на обработку информации, качества обработки информации, повышение степени достоверности и защищенности информации.

Автоматизация должна обеспечить прием сотрудника на работу, отслеживание его состояния в течение периода его работы и увольнение сотрудника с работы. Дополнительно, отдел кадров должен формировать много отчетов для бухгалтерии и внешним организациям, например, Пенсионным фондом и статистическими органами. Работа отдела кадров регламентирована законодательством (например, КЗоТ) и коллективным договором. В частности, предопределены формы приказов о приеме и увольнении, личные карточки сотрудников, формы отчетов. Данный программный комплекс может работать как на машинах, подключенных к вычислительной сети так и просто на локальных машинах, если сеть отсутствует.

Поскольку первичное заполнение таблиц и ввод их в машину ведет человек, ошибки в данных являются не исключением, а правилом, и любая информационная система должна иметь средство для диагностики и исправления ошибок.

Автоматизированная система имеет следующие функции:

* заключение трудовых договоров при приеме на работу новых сотрудников;
* формирование, редактирование и печать личной карточки сотрудника;
* формирование и редактирование справочников должностей и подразделений организации;
* ведение штатного расписания;
* кадровое перемещение и увольнение сотрудников;
* учет вакансий;
* расчет стажа и отпусков;
* поиск по журналу сотрудников;
* формирование и печать необходимой отчетности.

1. Методы и модели положенные в основу проекта

Отдел кадров - это структура в организации, которая занимается управлением персоналом в организации, то есть деятельностью людей, выполняющих на предприятии или в организации функции, способствующие наиболее эффективному использованию человеческих ресурсов для достижения первичных целей предприятии (организации). [6].

Цель отдела кадров - это способствование достижению целей предприятия (организации) путем обеспечения ее необходимыми кадрами и эффективного использования их квалификации, опыта, мастерства, работоспособности, творческого потенциала.

Основные задачи отдела кадров ОАО «Кузлитмаш» можно распределить по направлениям.

Учетно-контрольное направление кадровой работы предполагает решение следующих основных задач:

* прием работников на предприятие;
* учет работников;
* увольнение работников;
* работа с временно отсутствующими работниками предприятия (находящихся в отпусках, отсутствующих по болезни, убывших в командировки и т. п.).

Планово-регулятивное направление кадровой работы предполагает решение следующих основных задач:

* подбор (поиск и отбор) работников для предприятия;
* расстановка работников предприятия;
* перемещение работников предприятия;
* становление в должности и адаптация работников.

Отчетно-аналитическое направление кадровой работы предполагает решение следующих основных задач:

* изучение работников;
* оценка работы работников;
* аналитическая работа;
* подготовка отчетов.

Координационно-информационное направление кадровой работы предполагает решение следующих основных задач:

* профессиональная подготовка (обучение и переподготовка) работников;
* организация приема работников предприятия (по служебным и личным вопросам);
* работа с письменными обращениями работников предприятия;
* архивная и справочная работа.

Организационно-методическое направление кадровой работы предполагает решение следующих основных задач:

* документирование деятельности работников предприятия;
* кадровая работа в подразделениях предприятия;
* планирование кадровой работы;
* руководство кадровой работой.

Все задачи отдела кадров определяются спецификой работы организации, при этом учитываются:

* размеры организации;
* направление бизнеса (производство, торговля, сфера оказания услуг населению);
* стратегические цели организации;
* стадия развития организации;
* численность персонала;
* приоритетные задачи работы с персоналом.

Обязанности и структура отдела кадров может быть различной в зависимости от вида деятельности организации.

Классический подход к управлению персоналом получил название «управление кадрами»». Все управление персоналом сосредоточивается в отделе кадров и направлено на то, чтобы обеспечить наличие нужных людей в нужное время в нужных местах и освобождение от ненужных. Руководство людьми представляет собой самостоятельную функцию, осуществляемую непосредственно линейными руководителями независимо от кадровых служб.

К главным задачам системы управления персоналом относят [5]:

* обеспечение организации квалифицированными кадрами;
* создание необходимых условий для эффективного использования знаний, навыков и опыта работников;
* совершенствование системы оплаты труда и мотивации;
* повышение удовлетворенности трудом всех категорий персонала;
* предоставление работникам возможностей для развития, повышения квалификации и профессионального роста; стимулирование творческой активности;
* формирование и сохранение благоприятного морально-психологического климата;
* совершенствование методов оценки персонала; управление внутренними перемещениями и карьерой сотрудников;
* участие в разработке организационной стратегии.

Принципами построения современной системы управления персоналом считаются:

* эффективность подбора и расстановки сотрудников;
* справедливость оплаты труда и мотивации, вознаграждение не только за индивидуальные, но и коллективные достижения;
* продвижение работников в соответствии с результатами труда, квалификацией, способностями, личными интересами, потребностями организации;
* быстрое и эффективное решение личных проблем.

Работа службы персонала ОАО «Кузлитмаш» имеет два направления: тактическое и стратегическое. В рамках тактического направления осуществляется текущая кадровая работа. Это анализ состояния и планирование потребностей в кадрах, разработка штатных расписаний, осуществление набора, оценки и отбора персонала; планирование ближайших кадровых перемещений и увольнений, текущий учет и контроль, подготовка, переподготовка и повышение квалификации, формирование резерва на выдвижение.

Главным направлением деятельности кадровой службы считается формирование трудовых ресурсов: планирование потребности в них и организация практических мероприятий по набору кадров, разрешение конфликтов, проведение социальной политики.

Суть кадровой работы состоит в определении того, что конкретно, кем, как и с помощью чего должно делаться на практике в данный момент в сфере управления персоналом. Решение этих повседневных задач основывается на административных методах.

Стратегическое направление работы службы персонала ориентировано на формирование кадровой политики организации — системы теоретических взглядов, идей, требований, практических мероприятий в области работы с персоналом, ее основных форм и методов.

Разрабатываемое приложение является автоматизированным рабочим местом работника отдела кадров.

Инспектор по кадрам ОАО «Кузлитмаш» занимается непосредственной работой с персоналом предприятия. Он осуществляет оформление документов при приеме на работу новых сотрудников. Проверяет наличие необходимых документов для трудоустройства. В его обязанности входит заполнение и печать приказов о приеме, контрактов и трудовых договоров. Так же кадровик занимается оформлением отпусков и ведёт учет количества дней отпуска по каждому сотруднику. Одной из основных задач инспектора по кадрам является заполнение личной карточки сотрудника, а так же внесение в нее актуальных изменений, таких как паспортные данные, сведения об образовании, семейном положении и составе семьи. Так же в личную карточку заносятся данные о воинском учете. Обязательным является отражение изменений о занимаемой должности сотрудника и перемещении/переводе по структурным подразделениям.

Пользовательский интерфейс представляет собой совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя программы с компьютером.

Пользовательский интерфейс напрямую зависит от решаемых программным обеспечением задач, входных и выходных данных; однако при этом существуют значительная свобода в том, в каком виде все эти данные будут представлены пользователю. От того, насколько пользовательский интерфейс будет функционален, понятен и удобен конечному пользователю, во многом зависит успешность решения задачи поставленной при проектировании программного средства.

Существует ряд критериев, которым должен удовлетворять качественный интерфейс:

* должен обеспечивать минимальное время выполнения задачи;
* должен обеспечивать минимальное число непроизвольных ошибок пользователя;
* объем вводимой пользователем информации должен стремиться к минимуму (одни и те же данные не должны вводиться несколько раз);
* простота и визуальная привлекательность.

Автоматизация отдела кадров должна избавить от выполнения рутинных операций при работе с кадрами, ведении табелей, дальнейшего учета рабочего времени. Автоматизированное хранение и обработка полной кадровой информации также позволяет эффективно осуществлять подбор и перемещение сотрудников.

Приложение должно позволять добавлять, удалять и изменять в базе данных приказы о приеме, увольнении и переводе сотрудников, данные сотрудников, элементы справочников подразделений и должностей, а так же осуществлять формирование структуры предприятия, резервное копирование и восстановление списка приказов и сотрудников предприятия. Программная система должна вести учет работающих и уволенных сотрудников, а так же вести учет вакансий. Позволять редактировать личные карточки выбранных сотрудников и изменять данные о семейном положении и составе семьи, добавляя и редактируя членов семьи с сохранением их в базе данных. Позволять отслеживать текущее структурное подразделение и должность сотрудника и изменять их. Позволять только администратору или начальнику отдела кадров добавлять и удалять структурные подразделения и должности предприятия, изменять информацию о них, просматривать статистику о движении кадров, представленную в виде графиков и диаграмм.

Таким образом, создание автоматизированной системы преследовало решение следующих задач:

– автоматизация работы отдела кадров;

– повышения производительности труда отдела кадров;

– уменьшения затрат на содержание отдела кадров.

Указанные задачи направлены на достижение цели – создать эффективную систему автоматизации процесса контроля и учета сотрудников на предприятии. Данная система призвана помочь пользователям (линейным руководителям, специалистам отдела кадров, труда и заработной платы, менеджерам высшего звена) получать информацию в количественном выражении о расходах рабочего времени сотрудников предприятия.

На основе анализа предприятия можно выделить основные функции отдела кадров, подлежащие автоматизации:

* учет личных данных сотрудников;
* формирование штатного расписания;
* ведение базы данных справочников;
* учет движения кадров на предприятии;
* учет воинского состояния сотрудников.

Предметная область компании была исследована в приложении Rational Rose. Рассмотрим данные процессы с помощью диаграмм деятельности.

Диаграмма деятельности — UML-диаграмма, на которой показано разложение некоторой деятельности на её составные части. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий англ. action, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого. Диаграммы деятельности используются при моделировании бизнес-процессов. [2, 7].

Диаграммы деятельности работы с приказами

Для работы с приказами надо выбрать тип приказа и заполнить его поля. Номер и дата приказа имеются во всех типах приказов и заполняются идентично. Если выбран «Приказ о приеме» то заполняется номер и дата приказа, ФИО и табельный номер сотрудника, выбирается подразделение и должность, на которую он принимается. Для документа «Приказ об увольнении» ФИО и табельный номер сотрудника вводить не нужно. Увольняемый сотрудник выбирается из списка работников. При этом автоматически заполняются его ФИО, табельный номер, подразделение и должность. Так же нужно заполнить дату увольнения сотрудника и статью, по которой производится увольнение. При создании документа «Приказ о переводе» сотрудник так же выбирается из списка работников. Его данные заполняются автоматически. Затем нужно выбрать место перевода, т.е. подразделение и должность, на которую переводится сотрудник. При выборе «Приказа по штату» номер приказа заполняется автоматически по порядку нумерации. Далее заполняется дата приказа и примечание. В поле «примечание» можно выбрать подразделение, для которого будет составляться данный приказ по штатному расписанию. После того, как данные введены, необходимо сохранить приказ. Процесс является линейным.

На рисунке 2.1 представлена диаграмма деятельности «Работа с приказами».

Рисунок 2.1 – Диаграмма деятельности «Работа с приказами»

Диаграммы деятельности заполнения данных о сотрудниках

Для заполнения данных о сотрудниках необходимо открыть личную карточку сотрудника. Затем переключаясь между вкладками заполнить такие данные как пол, дата и место рождения, паспортные данные, адрес. Заполнить сведения о семейном положении и составе семьи, сведения об образовании которые включают наименование учебного заведения, номер и дату диплома, а так же специальность по диплому. Также необходимо вести сведения о трудовой деятельности сотрудников и при необходимости делать отметки о выходе сотрудника на пенсию. Кроме того в личную карточку заносятся данные о воинском учете сотрудника. Они включают в себя дату постановки, группу и категорию учета, состав и воинское звание, №ВУС, годность и номер военного билета. После заполнения всех необходимых полей сотрудник отдела кадров сохраняет изменения в базе данных.

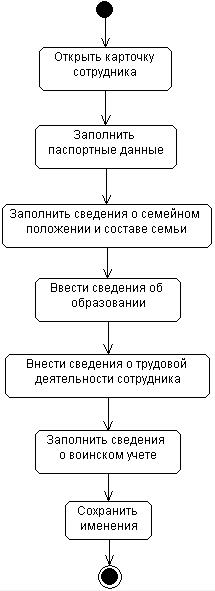
На рисунке 2.2 представлена диаграмма деятельности «Заполнение данных о сотрудниках».

Рисунок 2.2 – Диаграмма деятельности «Заполнение данных о сотрудниках»

Диаграммы деятельности ведения справочников

Процесс ведения справочника начинается с выбора требуемого справочника. Это может быть справочник подразделений или справочник должностей. Далее пользователь выбирает ввести новый элемент справочника или редактировать существующий. После создания нового элемента или редактирования существующего, пользователь сохраняет изменения.

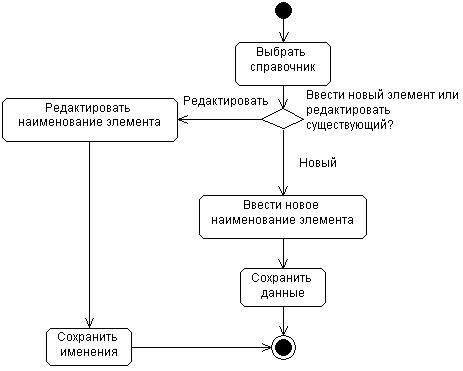
На рисунке 2.3 представлена диаграмма деятельности «Ведение справочников».

Рисунок 2.3 – Диаграмма деятельности «Ведение справочников»

Диаграммы деятельности формирование структуры предприятия

Структура подразделений и должностей является древовидной. Для того чтоб сформировать структуру предприятия, сотрудник открывает справочник подразделений, выбирает из него необходимое и добавляет его в структуру. Затем в каждое добавленное подразделение необходимо добавить должности, соответствующие данному подразделению. Для этого в структуре предприятия необходимо выделить подразделение, в которое нужно добавить должность. После этого нужно открыть справочник должностей, выбрать нужную должность и добавить ее в подразделение в структуре предприятия.

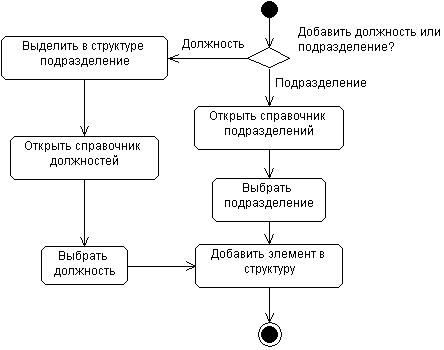
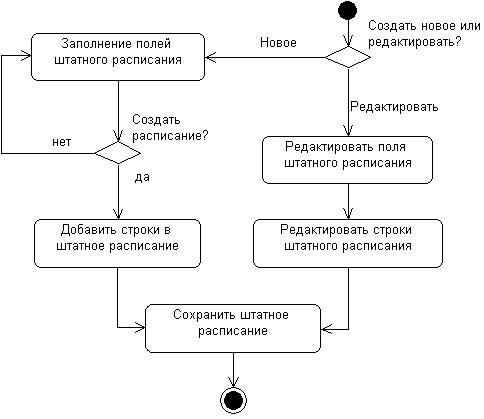
На рисунке 2.4 представлена диаграмма деятельности «Формирование структуры предприятия».

Рисунок 2.4 – Диаграмма деятельности «Формирование структуры предприятия»

Диаграммы деятельности формирования штатного расписания

Формирование штатного расписания начинается с создания нового штатного расписания или редактирования существующего. Номер штатного расписания присваивается автоматически, в соответствии с нумерацией. Далее следует заполнить дату расписания и выбрать приказ, на основании которого составляется данное штатное расписание. Затем вводится дата и тарифная ставка первого разряда, на основании которой будет производиться расчет оклада. После заполнения всех полей, в табличную часть штатного расписания добавляются строки. Таким образом, формируется список элементов штатного расписания. При добавлении строки происходит проверка – создано новое расписание или открыто существующее для редактирования. Если штатное расписание новое, тогда перед добавлением строки будет предложено добавить текущее расписание в базу данных. После этого можно добавлять строки в табличную часть. Каждому элементу (строке) присваивается уникальный номер-идентификатор. Все элементы сохраняются в отдельную таблицу в базе данных, но каждый из них привязан к номеру расписания, к которому он принадлежит. При создании нового элемента сотрудник выбирает должность, для которой создается данный элемент. При этом автоматически заполняется количество штатных единиц и количество занятых вакансий по выбранной должности. Количество штатных единиц можно изменить, если планируется расширение или сокращение штата сотрудников. Далее выбирается тарифный разряд из справочника ЕКДС. При этом автоматически происходит расчет должностного оклада. После заполнения табличной части штатного расписания изменения сохраняются в БД. На рисунке 2.5 представлена диаграмма деятельности «Формирование штатного расписания».

Рисунок 2.5 – Диаграмма деятельности «Формирование штатного расписания»

1. Проектирование системы учета персонала предприятия
   1. Функциональный анализ предметной области

На основе этих данных и с учетом требований к качеству программного обеспечения формируются требования к проектируемой информационной системе с различной степенью детализации. В системе ISO существуют наборы стандартов, которые могут быть полезны на разных этапах формирования требований к проектируемой информационной системе.

Базовые решения модели программного обеспечения: разрабатываемое приложение будет построено по двухуровневой архитектуре «клиент – сервер». В нашем случае программой сервером будет система управления базой данных, а клиентом – разрабатываемое приложение.

В качестве языка разработки была выбрана технология Windows Forms и язык разработки C#. Технология Windows Forms представляет собой набор классов .NET, реализующих функциональность, присущую стандартным и расширенным компонентам Windows-приложений. [8].

В качестве системы управления базами данных была выбрана Microsoft SQL Server 2008 R2.

Она обладает первоклассной масштабируемостью, возможностью создавать хранилища данных, продвинутыми средствами анализа и достаточной безопасностью, что позволяет использовать ее как основу для важных бизнес-приложений. Ее можно объединить в один пакет с такими средствами разработки баз данных и приложений, как Visual Studio, или внедрить в приложение, работающее с локальными базами данных.[9]

Для того, чтобы более точно понять, как должна работать система, все чаще используется описание функциональности системы через варианты использования (UseCase или прецеденты). Диаграмма вариантов использования (сценариев поведения, прецедентов) является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Это описание последовательности действий, которые может осуществлять система в ответ на внешние воздействия пользователей или других программных систем. Варианты использования отражают функциональность системы с точки зрения получения значимого результата для пользователя, поэтому они точнее позволяют ранжировать функции по значимости получаемого результата.

Диаграмма вариантов использования предназначены в первую очередь для определения функциональных требований к системе и управляют всем процессом разработки. Все основные виды деятельности, такие как анализ, проектирование, тестирование выполняются на основе вариантов использования. Во время анализа и проектирования варианты использования позволяют понять как результаты, которые хочет получить пользователь влияют на архитектуру системы и как должны себя вести компоненты системы, для того чтобы реализовать нужную для пользователя функциональность.

Данная диаграмма состоит из актеров, вариантов использования и отношений между ними. При этом актером (действующим лицом, актантом) называется любой объект, субъект или система, взаимодействующая с моделируемой системой извне. Это может быть человек, техническое устройство или другая система, которая может служить источником воздействия на моделируемую систему. В свою очередь вариант использования – это спецификация сервисов (функций), которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемых системой при взаимодействии с актером. При этом в модели никак не отражается то, каким образом будет реализован этот набор действий [3, c. 125].

При построении диаграммы могут использоваться также общие элементы нотации: примечания и механизмы расширения.

Актер графически отображается с помощью фигуры «проволочного человечка», под которым записывается его имя. Для нашего программного обеспечения это будет пользователь инспектор по кадрам. Именно он будет работать в программном комплексе.

Основной функционал программы предназначен для сущности «Инспектор по кадрам», так как эта сущность непосредственно работает с сотрудниками, а значит и определяет весь комплекс предполагаемых необходимых работ. Исходя из обязанностей кадровика по проекту, в диаграмме изображены следующие варианты использования:

* добавление и редактирование элементов справочников подразделений и должностей;
* формирование структуры предприятия;
* заполнение штатного расписания;
* учет приказов по штату;
* составление приказов о приеме сотрудников;
* учет движения сотрудников;
* заполнение личных карточек сотрудников;
* учет выхода на пенсию;
* печать данных по сотрудникам.

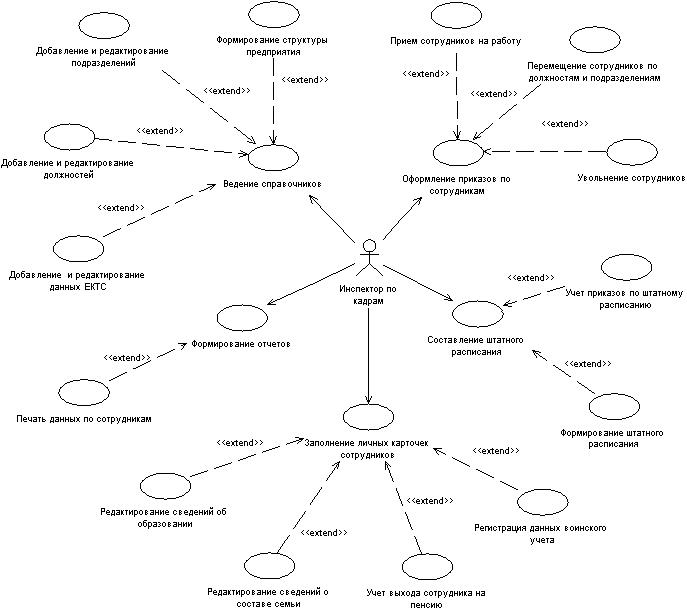
Диаграмма вариантов использования представлена на рисунке 3.1.

Рисунок 3.1 – Бизнес-актёр пользователь «инспектор по кадрам» и его функциональные обязанности

Описание спецификаций для вариантов использования проектируемой системы представим в виде трассировочной таблицы (таблица 3.1).

Таблица 3.1 – Трассировочная таблица для инспектора по кадрам

|  |  |  |  |
| --- | --- | --- | --- |
| **№п/п** | **Выполняемая операция** | **Функция системы** | **Use-Case** |
| 1 | Приказ о приеме на работу служит основанием для заключения трудовых отношений с кандидатом. | Оформление приказов по сотрудникам | Прием сотрудников на работу |
| 2 | В случае перевода сотрудника по подразделениям или должностям, оформляется приказ на перемещение. | Оформление приказов по сотрудникам | Перемещение сотрудников по должностям и подразделениям |
| 3 | При расторжении трудовых отношений оформляется приказ на увольнение. | Оформление приказов по сотрудникам | Увольнение сотрудников |
| 4 | Если в организации создается или переименовывается структурное подразделение, данные по нему должны быть отражены в справочнике. | Ведение справочников | Добавление и редактирование подразделений |
| 5 | В случае добавления или изменения должности, ее данные должны быть внесены в справочник. | Ведение справочников | Добавление и редактирование должностей |
| 6 | Для того чтоб каждому подразделению назначить список его должностей, составляется структура предприятия (должности/подразделения). | Ведение справочников | Формирование структуры предприятия |
| 7 | При изменении в едином квалификационном справочнике должностей служащих, необходимо произвести изменения разрядов и коэффициентов. | Ведение справочников | Добавление и редактирование данных ЕКТС |
| 8 | При предоставлении подтверждающих документов инспектор по кадрам вносит изменения об образовании в личную карточку. | Заполнение личных карточек сотрудников | Редактирование сведений об образовании |
| 9 | А так же производит редактирование состава семьи. | Заполнение личных карточек сотрудников | Редактирование сведений о составе семьи |

Продолжение таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| **№п/п** | **Выполняемая операция** | **Функция системы** | **Use-Case** |
| 10 | В случае выхода сотрудника на пенсию, инспектор по кадрам делает соответствующую отметку в личной карточке. | Заполнение личных карточек сотрудников | Учет выхода сотрудника на пенсию |
| 11 | При трудоустройстве сотрудника, инспектор по кадрам вносит сведения о воинском из военного билета в личную карточку. | Заполнение личных карточек сотрудников | Регистрация данных воинского учета |
| 12 | При необходимости, данные по сотруднику формируются в печатную форму. | Формирование отчетов | Печать данных по сотруднику |
| 13 | При формировании штатного расписания составляется приказ по внесению изменений в должности или заработную плату. | Составление штатного расписания | Учет приказов по штатному расписания |
| 14 | Согласно приказа, сотрудник отдела кадров вносит изменения в штатное расписание. | Составление штатного расписания | Формирование штатного расписания |

* + 1. Разработка структуры и схемы работы объекта

При анализе ресурсов программного обеспечения нужно отметить, что комплекс будет включать в себя человеческие ресурсы, база данных со всеми данными о работе, ресурсы данных о сотрудниках являются инициаторами рабочего процесса, выходными ресурсами будут данные о проведенных изменениях.

Каждое пользовательское требование, связанное с функциональностью, можно представить в виде процесса взаимодействия пользователя (актёра) и системы через соответствующий интерфейс. В случае если пользователь физическое лицо, то взаимодействие будет осуществляться через графический интерфейс (GUI), т. е. посредством элементов экранных форм.

Схемы ресурсов системы отображают конфигурацию блоков данных и обрабатывающих блоков, которые требуется для решения задачи или набора задач.

Приложение по автоматизации отдела кадров представляет собой клиент-серверную систему, где в качестве серверной части выступает MS SQL Server 2008R2, а клиентской частью является интерфейс приложения. Такая структура системы может позволить работать с ней множеству пользователей, получая самую актуальную информацию в реальном времени.

Ядром системы является база данных, аккумулирующая всю необходимую производственную информацию. Обращения кадровика к базе обрабатываются массивом хранимых процедур. [11].

Клиентская часть системы представляет собой Windows Forms, написана на языке C#. Приложение имеет дружественный интуитивно понятный интерфейс. Весь функционал сгруппирован соответственно пользовательских задач. Ввод и редактирование данных осуществляется с помощью диалоговых окон. Вывод информации производится в табличном виде. Также имеется возможность изменения порядка следования колонок таблицы и фильтрации данных.

Подробная схема ресурсов системы содержится в Приложении Ж, на графической части РТДП 3033.057.07.

Схема ресурсов была построена на основании определенных правил обработки данных и конфигурации управляющих этой обработкой блоков.

Информационная система состоит из:

* хранилища данных SQL Server;
* сервера, обрабатывающего запросы пользователей;
* компьютеры пользователей;
* вспомогательных ресурсов (дополнительных библиотек и т.п.).

Доступ сервера к БД осуществляется посредством технологии ADO, настройка источников данных осуществляется с помощью панели настройки сервера.

* 1. Разработка базы данных

Частью процесса создания средства управления проектами является информационное моделирование. Оно упрощает разработку информационной системы, помогает сделать её легко сопровождаемой и управляемой.

В данном дипломном проекте для моделирования информационной системы использовался Erwin. В основе реализации моделирования лежат теория реляционных баз данных и методология IDEF1X.

Основные составляющие части визуального представления Erwin – это сущности, атрибуты и связи. Каждая сущность является множеством сходных объектов, называемых экземплярами. Каждый экземпляр индивидуален и должен отличаться от всех остальных экземпляров. Атрибут выражает некоторое свойство объекта. В базе данных сущности соответствует таблица, экземпляру сущности – строка (запись) в таблице, а атрибуту – колонка (поле) таблицы. Построение информационной модели данных предполагает определение сущностей и атрибутов. [10].

Связь – это функциональная зависимость между двумя сущностями. В Erwin связь может быть идентифицирующей и не идентифицирующей. Связь называется идентифицирующей, если экземпляр дочерней сущности идентифицируются через ее связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в первичный ключ дочерней сущности. Дочерняя сущность при идентифицирующей связи всегда является зависимой. Связь называется не идентифицирующей, если экземпляр дочерней сущности идентифицируется иначе, чем через связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в состав не ключевых атрибутов дочерней сущности. [3].

В данной модели связи между сущностями являются идентифицирующими и не идентифицирующими.

Сущности бывают зависимыми и независимыми. Зависимые сущности отображаются при помощи прямоугольных таблиц с закругленными углами, а независимые – с помощью прямоугольных таблиц.

В Erwin существуют два уровня представления — логический и физический. Логический уровень означает непосредственное отображение фактов из реальной жизни. На нем данные представляются так, как они выглядят в реальном мире. Объекты модели, представляемые на логическом уровне, называются сущностями и атрибутами. Логическая модель данных не связана с конкретной СУБД. [4].

Рассмотрим логический уровень информационной модели проектируемой системы. В ней выделено 13 сущностей:

* Штатное Расписание – сущность хранит сведения о текущем штатном расписании предприятия;
* Личные данные – сущность для хранения личных данных сотрудников;
* Образование – сущность для хранения сведений об образовании сотрудников;
* ПриемНаРаботу – сущность для хранения сведений о приеме на должность;
* Перевод – сущность для хранения сведений о переводах сотрудников по должностям;
* Увольнение – сущность для хранения сведений об увольнении сотрудников по должностям;
* Сотрудники – сущность для хранения сведений о сотрудниках;
* СоставСемьи – сущность для хранения сведений о составе семьи сотрудников.
* Пенсионеры – сущность для хранения сведений о выходе сотрудников на пенсию.
* ВоинскийУчет – сущность для хранения сведений о воинском учете сотрудников.
* ЕТКС – сущность для хранения сведений о единой тарифно-квалификационной сетке сотрудников.
* СоставШтатного – сущность для хранения сведений о текущем составе штатного расписания.
* ПриказШтатное – сущность для хранения сведений о приказах, затрагивающих штатное расписание.

Для более наглядного описания информационной системы между сущностями имеются определенные связи.

Логический уровень данной информационной модели представлен на рисунке 3.6, физический уровень на рисунке 3.2.

Рисунок 3.2 – Логический уровень информационной модели

Проведем нормализацию физической модели БД. Процесс нормализации базы данных - это формальный метод анализа отношений на основе их первичных или потенциальных ключей и существующих функциональных зависимостей, являющийся одним из наиболее строгих способов улучшения характеристик БД. Существует пять нормальных форм (НФ) баз данных.

Рисунок 3.3 – Физический уровень информационной модели

Первая нормальная форма

Отношение находится в первой нормальной форме (1НФ), если все его атрибуты имеют простые (атомарные) значения, т.е. не имеют компонентов.

Иными словами, значения в домене каждого атрибута отношения не являются ни списками, ни множествами простых или сложных значений, в отношении не должно быть повторяющихся групп. Каждый простой атрибут должен иметь свою колонку в таблице. В противном случае отношение считается ненормализованным и ему соответствует многоуровневая таблица (иерархия) в отличие от однородной табличной структуры нормализованного отношения. Схема всей базы данных находится в 1НФ, если каждая схема отношения находится в 1НФ. [12].

Для разработанной изначально схемы таблица ЛичныеКарточки включает колонку ФИО, которая не является атомарной и поэтому её необходимо разбить на колонки фамилия, имя, отчество.

Вторая нормальная форма

Отношение находится во второй нормальной форме (2НФ), если она находится в 1НФ, и при этом любой её атрибут, не входящий в состав первичного ключа, функционально полно зависит от первичного ключа.

Вторая нормальная форма применяется к отношениям с составными ключами, т. е. к таким отношениям, первичный ключ которых состоит из двух или более атрибутов. Отношение, у которого первичный ключ включает только один атрибут, всегда находится во 2НФ.

В данной схеме базы данных отношения имеют первичные ключи, состоящие только из одного атрибута, поэтому схема БД находится во второй нормальной форме.

Третья нормальная форма

Отношение находится в третьей нормальной форме (3НФ), если она находится во второй нормальной форме (2НФ) и при этом любой её неключевой атрибут нетранзитивно зависит от первичного ключа, т.е. неключевые атрибуты не должны зависеть друг от друга.

Например, таблица Сотрудник изначально включала колонки должность и подразделение, которых для одного и того же сотрудника может быть много. Для того, чтобы избежать дублирования информации и ошибок, создаётся отдельная таблица Подразделения и Должности.

При решении практических задач в большинстве случаев третья нормальная форма является достаточной. Процесс проектирования реляционной базы данных, как правило, заканчивается приведением к 3НФ.

В результате нормализации была получена физическая модели базы данных, графически представленная на рисунке 3.4.

Рисунок 3.4 – Физический уровень информационной модели после нормализации

В результате проектирования и нормализации были получены следующие таблицы:

* Штатное Расписание – таблица хранит сведения о текущем штатном расписании предприятия;
* Подразделения – таблица для описания всех подразделений предприятия;
* Должности – таблица для описания всех должностей предприятия;
* Личные данные – таблица для хранения личных данных сотрудников;
* Образование – таблица для хранения сведений об образовании сотрудников;
* ПриемНаРаботу – таблица для хранения сведений о приеме на должность;
* Перевод – таблица для хранения сведений о переводах сотрудников по должностям;
* Увольнение – таблица для хранения сведений об увольнении сотрудников по должностям;
* Сотрудники – таблица для хранения сведений о сотрудниках;
* СоставСемьи – таблица для хранения сведений о составе семьи сотрудников.
* Пенсионеры – таблица для хранения сведений о выходе сотрудников на пенсию.
* ВоинскийУчет – таблица для хранения сведений о воинском учете сотрудников.
* ЕТКС – таблица для хранения сведений о единой тарифно-квалификационной сетке сотрудников.
* СоставШтатного – таблица для хранения сведений о текущем составе штатного расписания.
* ПриказШтатное – таблица для хранения сведений о приказах, затрагивающих штатное расписание.

Диаграмма классов – статическая структурная диаграмма, описывающая структуру системы, она демонстрирует классы системы, их атрибуты, методы и зависимости между классами.

Диаграммы классов являются центральным звеном методологии объектно-ориентированных анализа и проектирования.

Диаграмма классов показывает классы и их отношения, тем самым представляя логический аспект проекта. На стадии анализа диаграммы классов используются, чтобы выделить общие роли и обязанности сущностей, обеспечивающих требуемое поведение системы. На стадии проектирования диаграммы классов используются, чтобы передать структуру классов, формирующих архитектуру системы.

Важно отметить, что как представление, так и контроллер зависят от модели. Однако модель не зависит ни от представления, ни от контроллера. Тем самым достигается назначение такого разделения: оно позволяет строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели.

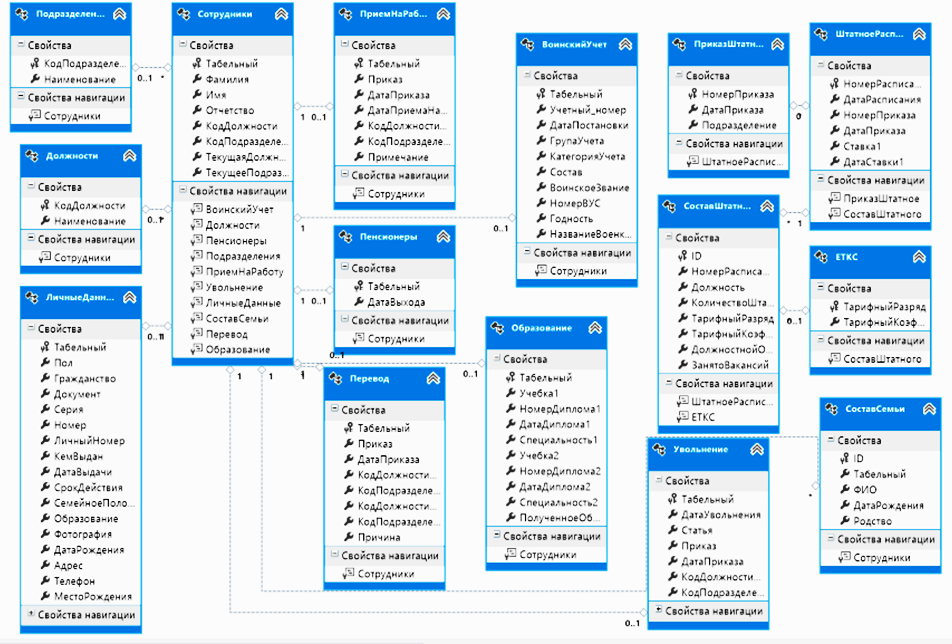
Диаграмма классов представлена на рисунке 3.5.

Рисунок 3.5 – Диаграмма классов

В процессе проектирования были разработаны следующие классы:

* GetAllSotrud\_Result – класс, который позволяет получить сведения обо всех сотрудниках;
* GetOffices\_Result – класс, который позволяет получить сведения обо всех подразделениях;
* GetPosts\_Result – класс, который позволяет получить сведения обо всех должностях;
* OKDBEntities – класс хранения сведений о единой тарифно-квалификационной сетке сотрудников
* ШтатноеРасписание – класс хранит сведения о текущем штатном расписании предприятия;
* Подразделения – класс для описания всех подразделений предприятия;
* Должности – класс для описания всех должностей предприятия;
* ЛичныеДанные – класс для хранения личных данных сотрудников;
* Образование – класс для хранения сведений об образовании сотрудников;
* Перевод – класс для хранения сведений о переводах сотрудников по должностям;
* Увольнение – класс для хранения сведений об увольнении сотрудников по должностям;
* Сотрудники – класс для хранения сведений о сотрудниках;
* СоставСемьи – класс для хранения сведений о составе семьи сотрудников;
* Пенсионеры – класс для хранения сведений о выходе сотрудников на пенсию.
* ВоинскийУчет – класс для хранения сведений о воинском учете сотрудников;
* СоставШтатного – класс для хранения сведений о текущем составе штатного расписания;
* ПриказШтатное – класс для хранения сведений о приказах, затрагивающих штатное расписание.

# Разработка интерфейса системы

Интерфейс пользователя представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными, чаще всего сложными, машинами, устройствами и аппаратурой.

Для разрабатываемой информационной системе был выбран оконный интерфейс, в котором доступ к основным функциям осуществляется с помощью системы меню.

Главной страницей для пользователя будет страница авторизации. Структура окон приложения приведена на рисунке 3.6.

Рисунок 3.6 – Схема размещения элементов интерфейса



Иерархия меню пользователя представлена на рисунке 3.7.



Рисунок 3.7 - Иерархия меню

Диалог пользователя и системы реализуется через формы и сообщения системы. При работе со справочниками пользователя необходимо заполнить таблицу данных справочников и нажать кнопку «Сохранить изменения», после чего ему будет выдано сообщение об успешном или не успешном завершении операции.

1. Разработка основных алгоритмов
   1. Общая схема работы автоматизации учета персонала

Схема программы в соответствии с ее реализацией представлена в Приложении И, на графической части РТДП 3033.057.08. Рассмотрим набор действий, которые описывает данная схема программы.

На начальном этапе происходит переход на основную форму программы. На главной форме формируется меню программы.

В зависимости от выбора пользователя, выражающемся во вводе необходимой операции, происходит переход к одному из возможных действий:

* работа с данными сотрудников;
* работа со структурой предприятия;
* работа со штатным расписанием;
* работа с воинским учетом.

В случае перехода к функции «Работа со справочниками» пользователю предлагается внести или изменить информацию в выбранном справочнике, после этого изменения сохраняются в базу данных.

В случае перехода к функции «Работа со структурой предприятия» необходимо внести подразделения или должностями. После этого изменения сохраняются в базу данных.

В случае перехода к функции "Работа со штатным расписанием" необходимо выбрать штатное расписание и редактировать его или ввести новое штатное расписание. После этого изменения сохраняются в базу данных.

В случае перехода к функции "Работа с воинским учетом " пользователю предлагается внести или изменить информацию о воинском учете сотрудников.

После выполнения одной из функции пользователь может переходит к выбору другой, нужной ему функции.

* 1. Алгоритм добавления нового сотрудника

После загрузки программы пользователь выбирает пункт меню Сотрудники/Приказ о приеме на работу.

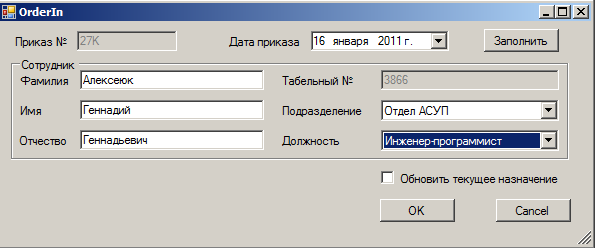
Появляется форма добавления данных о приказе на прием нового сотрудника.

Рисунок 4.1 – Окно для ввода информации в базу данных

Пользователь заполняет форму. Алгоритм начинает свое исполнение сразу же после нажатия кнопок «ОК».

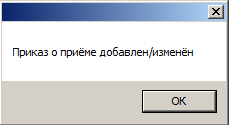
В случае успешного заполнения полей формы, система добавляет данные в БД и выводит окно с сообщением (рисунок 4.2)

Рисунок 4.2 – Информационное окно о добавлении записи

В случае неуспешного заполнения полей формы, система останавливает выполнение программного кода и выдает одно из сообщений об ошибке (рисунок 4.3)

Рисунок 4.3 – Информационные окна об ошибке

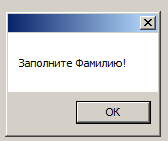
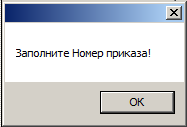
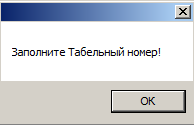


Схема алгоритма представлена на чертеже РТДП 3033.057.01, код представлен в Приложении А.

* 1. Алгоритм заполнения личной карточки сотрудника

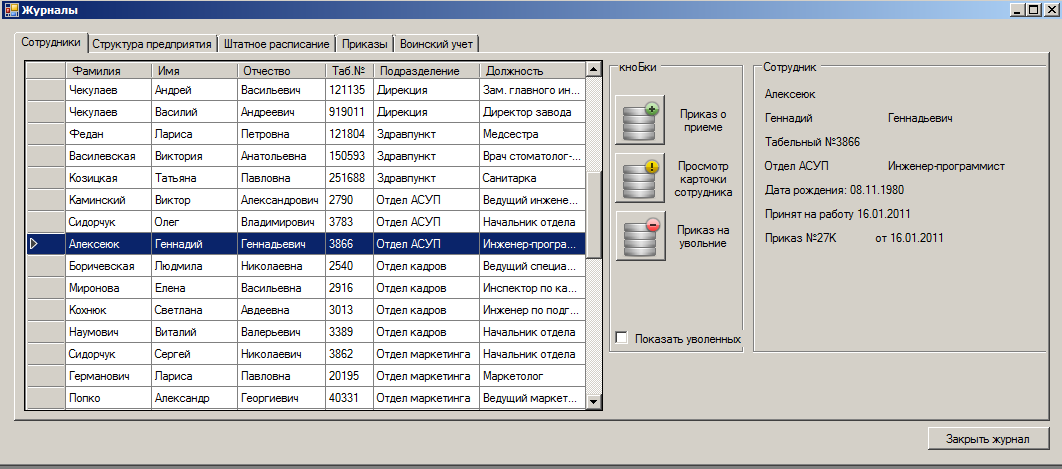
Пользователь выбирает пункт меню Сотрудники/Журнал сотрудников. Появляется форма просмотра журнала сотрудников.

Рисунок 4.4 – Журнал сотрудников

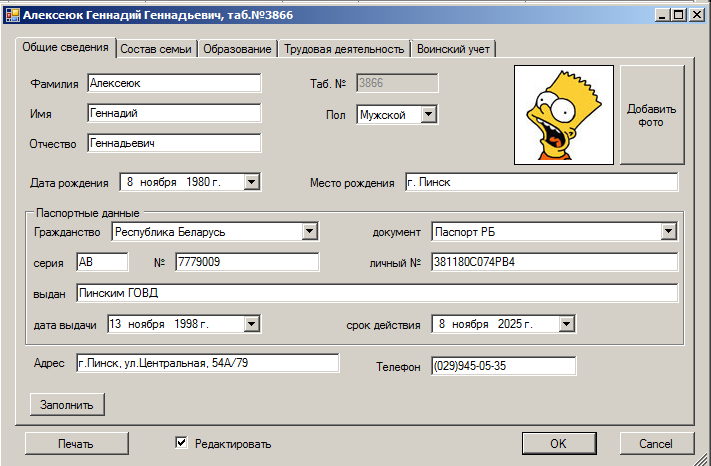
Пользователь выбирает сотрудника и нажимает кнопку «Просмотр карточки сотрудника». Система открывает форму карточки сотрудника (рисунок 4.5).

Рисунок 4.5 – Личная карточка сотрудника

Алгоритм начинает свое исполнение сразу же после нажатия кнопок «ОК».

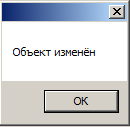
В случае успешного заполнения полей формы, система добавляет данные в БД и выводит окно с сообщением (рисунок 4.6)

Рисунок 4.6 – Информационное окно

В противном случае, система выводит окно с сообщением об ошибке (рисунок 4.7)

Рисунок 4.7 – Информационное окно с сообщением об ошибке

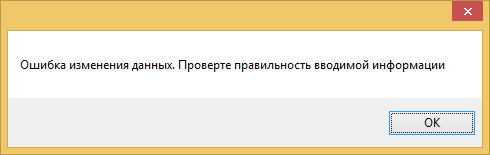
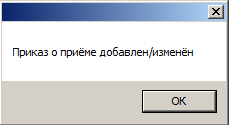


Схема алгоритма представлена на чертеже РТДП 3033.057.02, код представлен в приложении Б.

* 1. Алгоритм формирования структуры должности/подразделения

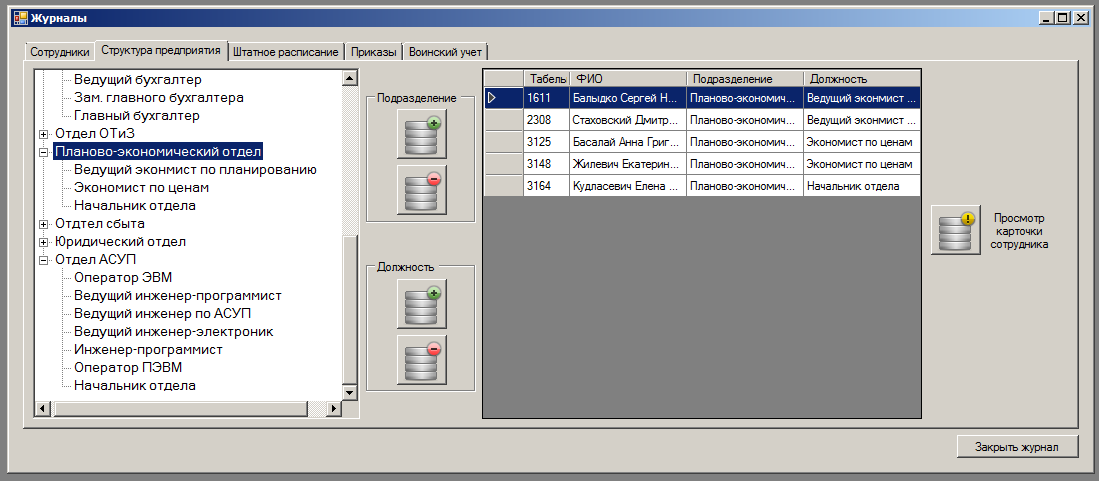
Пользователь выбирает пункт меню Структура предприятия/ Должности - Подразделения. Появляется форма просмотра журнала справочников.

Рисунок 4.8 – Структура предприятия

Алгоритм начинает свое исполнение сразу же после нажатия кнопок «Добавить» для того или другого справочника.

Появляется форма добавления данных (рисунок 4.9).

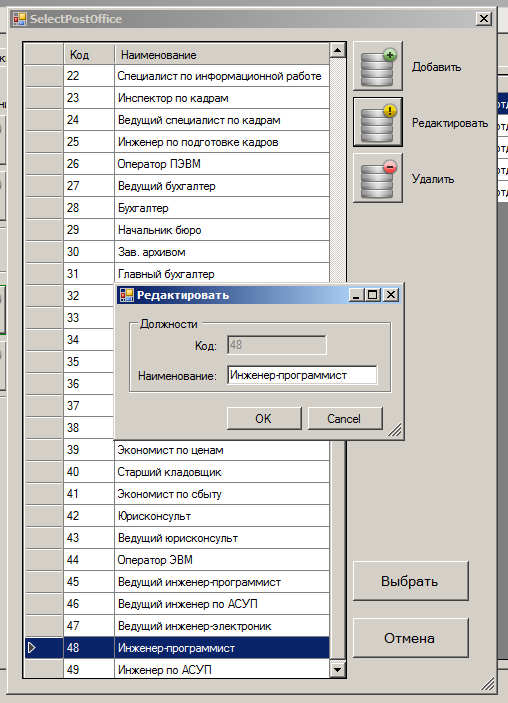
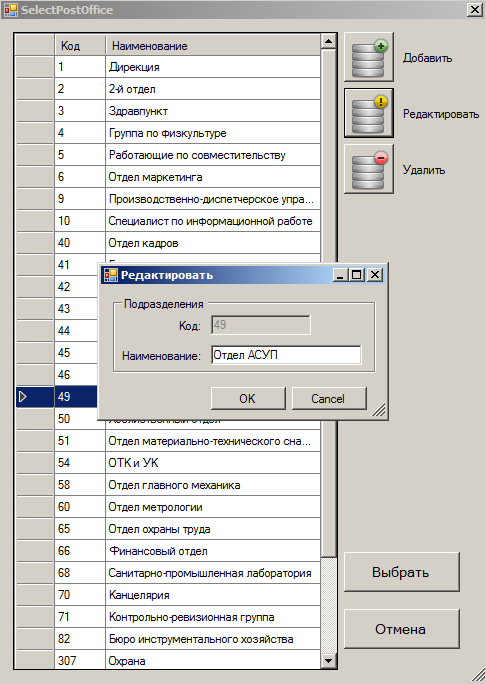


Рисунок 4.9 – Форма добавления данных в справочник

В случае успешного заполнения полей формы, система добавляет данные в БД и выводит окно с сообщением (рисунок 4.10)

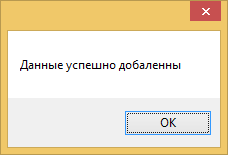
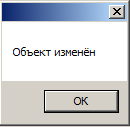


Рисунок 4.10 – Информационное окно

В противном случае, система выводит окно с сообщением об ошибке (рисунок 4.11)

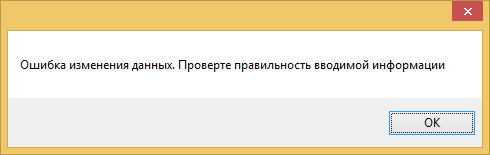
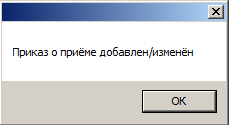


Рисунок 4.11 – Информационное окно с сообщением об ошибке

Схема алгоритма представлена на чертеже РТДП 3033.057.03, код представлен в приложении В.

* 1. Алгоритм создания штатного расписания

Модуль предназначен для добавления в базу данных новой записи штатного расписания, коррекции или удаления имеющейся записи.

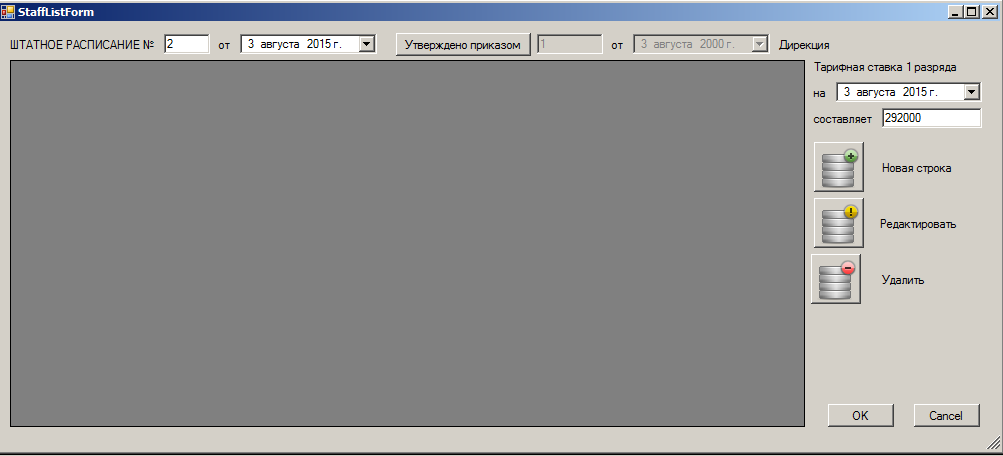
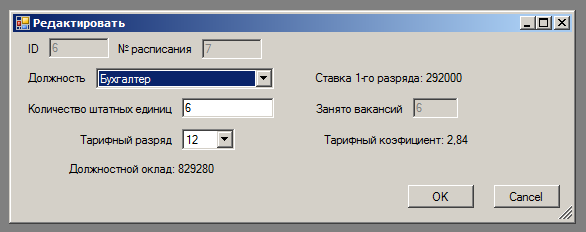
Пользователь выбирает пункт меню Штатное расписание / Создать новое. Алгоритм начинает свое исполнение. Появляется форма штатного расписания.

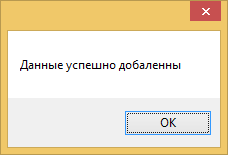
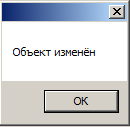
Рисунок 4.12 – Форма нового штатного расписания

Для добавления строк в новое штатное расписание нужно нажать соответствующую кнопку. При этом появится форма элемента штатного расписания.

Рисунок 4.13 – Форма элемента штатного расписания

В случае успешного заполнения полей формы, система добавляет данные в БД и выводит окно с сообщением (рисунок 4.14)

Рисунок 4.14 – Информационное окно



В противном случае, система выводит окно с сообщением об ошибке (рисунок 4.15)

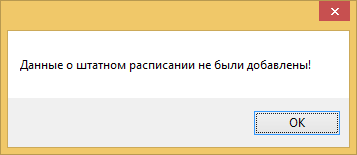
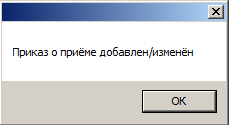


Рисунок 4.15 – Информационное окно с сообщением об ошибке

Схема алгоритма представлена на чертеже РТДП 3033.057.04, код представлен в приложении Г.

* 1. Алгоритм перевода сотрудника

После загрузки программы пользователь выбирает пункт меню Приказы /Приказ о переводе.

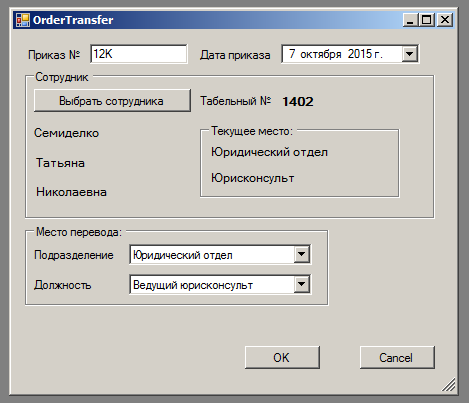
Появляется форма добавления данных о приказе на перевод сотрудника.

Рисунок 4.16 – Окно для ввода информации о приказе на перевод

Пользователь заполняет форму. Алгоритм начинает свое исполнение сразу же после нажатия кнопок «ОК».

В случае успешного заполнения полей формы, система добавляет данные в БД и выводит окно с сообщением (рисунок 4.17)

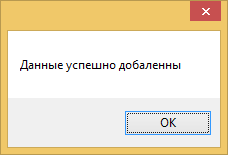
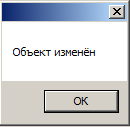


Рисунок 4.17 – Информационное окно о добавлении записи

В случае неуспешного заполнения полей формы, система останавливает выполнение программного кода и выдает сообщение об ошибке (рисунок 4.18)

Рисунок 4.18 – Информационное окно об ошибке

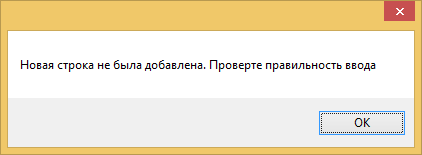
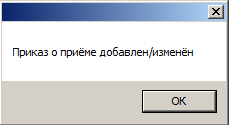


Схема алгоритма представлена на чертеже РТДП 3033.057.05, код представлен в приложении Д.

* 1. Алгоритм увольнения сотрудника

После загрузки программы пользователь выбирает пункт меню Приказы /Приказ об увольнении.

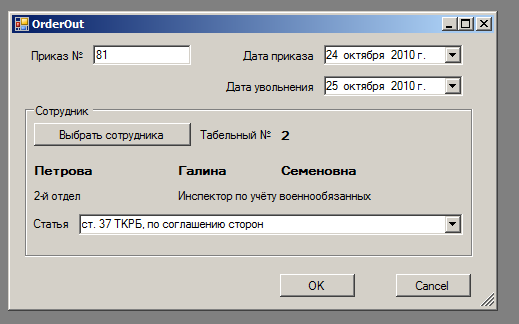
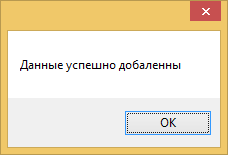
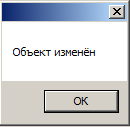
Появляется форма добавления данных о приказе на увольнение сотрудника.

Рисунок 4.19 – Окно для ввода информации о приказе на увольнение

Пользователь заполняет форму. Алгоритм начинает свое исполнение сразу же после нажатия кнопок «ОК».

В случае успешного заполнения полей формы, система добавляет данные в БД и выводит окно с сообщением (рисунок 4.20)

Рисунок 4.20 – Информационное окно о добавлении записи



В случае неуспешного заполнения полей формы, система останавливает выполнение программного кода и выдает сообщение об ошибке (рисунок 4.21).

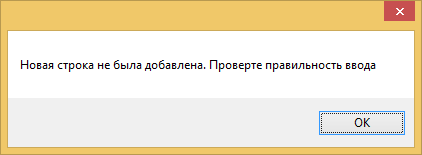
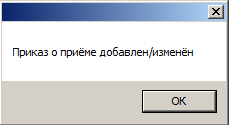


Рисунок 4.21 – Информационное окно об ошибке

Схема алгоритма представлена на чертеже РТДП 3033.057.06, код представлен в приложении Е.

1. Тестирование приложения.
   1. Общие положения

Для оценки работоспособности системы было проведено ручное тестирование.

Ручное тестирование – это процесс поиска дефектов в работе программы, когда тестировщик проверяет работоспособность всех компонентов программы, как если бы он был пользователем.

Системный подход к тестированию включал в себя несколько этапов.

Выбор методологии тестирования. Тестирование системы проводилось по нисходящему принципу. Это означает, что последовательно выполнялась проверка корректности работы сайта, начиная от наиболее общих функций (выбрать пункт меню, создать запись, отредактировать запись и т.п.), до узкоспециализированных (в случае возникновения ошибки на более высоком уровне).

Составление тестов с описанием выполнения и ожидаемым результатом. Был разработан набор функциональных тест-кейсов, сгруппированных в сценарии в соответствии с ролями пользователей системы. Для тестирования отдельных функций системы была применена стратегия «белого ящика». Это тестирование логики управления программы. Стратегия основана на анализе внутренней структуры программы. Стратегия «белого ящика» включает проверку операторов, решений, условий, покрытие решений и условий, комбинированное покрытие.

* 1. Разработка тестов и результаты тестирования приложения

Объектом испытания является созданное программное обеспечение «Программное средство автоматизации и учета персонала предприятия». Цель испытания: устранение возможных неполадок в работе программного обеспечения

Программу испытаний построим по следующей стратегии тестирования.

Разработаем набор функциональных тест-кейсов, сгруппированных в сценарии. Для тестирования отдельных функций системы была применена стратегия «белого ящика». Это тестирование логики управления программы. Стратегия основана на анализе внутренней структуры программы. Стратегия «белого ящика» включает проверку (покрытие) операторов, решений, условий, покрытие решений и условий, комбинированное покрытие.

Тестирование системы проводилось по нисходящему принципу. Это означает, что последовательно выполнялась проверка корректности работы программы, начиная от наиболее общих функций, до узкоспециализированных.

Для проверки на кроссплатформенность использовались следующие версии ОС, наиболее распространенные в настоящее время: Windows XP, Windows 7. Карта тестирования приведена в таблице 5.1.

Таблица 5.1 – Карта тестирования на кроссплатформенность совместимость

|  |  |  |  |
| --- | --- | --- | --- |
| Тестовый вариант | Ожидаемый результат | Windows XP | Windows 7 |
| 1 | 2 | 3 | 4 |
| Запустить приложение, | Появилась главная форма | + | + |
| Проверить доступность всех ссылок главного меню | Открытие соответствующих форм | + | + |
| Проверить единообразие формы отображения  информации. | Полученные результаты должны совпадать | + | + |
| Проверить содержание отображаемой информации | Полностью совпадают | + | + |

Разработаем Smoke Test для проверки правильности ввода оценок. Smoke Test в [тестировании программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) означает минимальный набор тестов на явные [ошибки](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3). Он обычно выполняется самим программистом; не проходящую этот тест программу не имеет смысла отдавать на более глубокое тестирование.

В области программного обеспечения, дымовое тестирование рассматривается как короткий цикл тестов, выполняемый для подтверждения того, что после сборки кода (нового или исправленного) устанавливаемое приложение, стартует и выполняет основные функции.

Испытаниям были подвергнуты модули «Справочники», «Работа с сотрудниками», «Штатное расписание» и «Приказы». Выбор именно этих модулей был обусловлен их центральной ролью в приложении, а также первостепенной важностью выполняемых ими функций.

Модуль «Справочники» отражает все справочные данные, необходимые для заполнения БД. Этот модуль предоставляет возможность пользователю добавлять и корректировать данные справочников.

Модуль «Штатное расписание» предназначен для создания нового штатного расписания.

Модуль «Работа с сотрудниками» предполагает ведение личных карточек сотрудников.

Модуль «Приказы» предоставляет возможность пользователю добавлять и редактировать данные о приеме, переводе и увольнении сотрудников.

Использование методики smoke-тестирования обусловлено ее способностью быстрого охвата значительной части функциональности приложения и способности выявить грубые ошибки и блокировки в процессе работы.

Для проведения испытаний указанных выше модулей, были разработаны тест-кейсы на проверку работоспособности основных функций. Их детальное рассмотрение и результаты работы приведены в таблице 5.2

В целом по результатам тестирования и разработки можно сделать выводы:

* в результате проведенного тестирования информационной системы не обнаружено фатальных ошибок, приводящих к отказу ее работы;
* структурные и визуальные характеристики форм не изменяются существенно в зависимости от ОС;
* навигационные характеристики также не зависят от ОС;
* все обязательные требования, предъявленные в спецификации, выполнены.

Таблица 5.2 – Тест-кейсы для проведения smoke-тестирования приложения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Модуль** | **Подмодуль/ экран** | **Описание теста** | **Ожидаемый результат** | **Статус** |
| **1** | **2** | **3** | **4** | **5** | **6** |
| **1-ST-1-1** | Справочники | Добавление данных в справочник подразделения | **Проверить работоспособность кнопки «Добавить»** 1. Открыть справочник подразделений 2. Нажать кнопку «Добавить» 3. Нажать кнопку «ОК» | 1. Откроется форма справочника 2. Откроется форма элемента с списком полей. 3. Происходит добавление данных. | Выполнено успешно |
| **2-ST-1-2** | Справочники | Редактирование данных в справочнике подразделения | **Проверить работоспособность кнопки «Редактировать»**  1. Открыть справочник подразделений 2. Выбрать элемент справочника 3. Нажать кнопку «Редактировать» 4. Нажать кнопку «ОК» | 1. Откроется форма справочника 2. Выделится вся строка элемента справочника 3.Откроется форма элемента с списком полей. 4. Происходит изменение данных. | Выполнено успешно |
| **3-ST-1-3** | Справочники | Добавление данных в справочник должности | **Проверить работоспособность кнопки «Добавить»** 1. Открыть справочник должностей 2. Нажать кнопку «Добавить» 3. Нажать кнопку «ОК» | 1. Откроется форма справочника 2. Откроется форма элемента с списком полей. 3. Происходит добавление данных. | Выполнено успешно |
| **4-ST-1-4** | Справочники | Редактирование данных в справочнике должности | **Проверить работоспособность кнопки «Редактировать»**  1. Открыть справочник должностей 2. Выбрать элемент справочника 3. Нажать кнопку «Редактировать» 4. Нажать кнопку «ОК» | 1. Откроется форма справочника 2. Выделится вся строка элемента справочника 3.Откроется форма элемента с списком полей. 4. Происходит изменение данных. | Выполнено успешно |
| **5-ST-2-1** | Структура предприятия | Редактирование структуры предприятия | **Проверить работоспособность кнопки «Выбрать»** 1. Открыть справочник  2. Выбрать элемент справочника 3. Нажать кнопку «Выбрать» | 1. Откроется форма справочника 2. Выделится вся строка элемента справочника. 3. Происходит добавление данных в структуру предприятия. | Выполнено успешно |

Продолжение таблицы 5.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** |
| **6-ST-2-2** | Структура предприятия | Редактирование структуры предприятия | **Проверить работоспособность кнопки «Удалить»** 1. Выбрать элемент в структуре предприятия 2. Нажать кнопку «Удалить» | 1. Выделится элемент структуры 2. Происходит удаление элемента из структуры предприятия. | Выполнено успешно |
| **7-ST-3-1** | Приказы | Отбор приказов | **Проверить работоспособность переключателей приказов**  1. Открыть журнал «Приказы» 2. Последовательно выбирать переключатели: «Приказы о приеме», «Приказы об увольнении», «Приказы о переводе», «Приказы по штату» | 1. Открывается журнал «Приказы» 2. В журнал последовательно загружаются «Приказы о приеме», «Приказы об увольнении», «Приказы о переводе», «Приказы по штату», в соответствии с выбранным переключателем. | Выполнено успешно |
| **8-ST-3-2** | Приказы | Добавление новой записи в приказы о приеме на работу | **Проверить работоспособность функции добавления приказа о приеме.** 1. Открыть журнал «Приказы» 2. Выбрать переключатель «Приказы о приеме» 3. Нажать кнопку «Новый приказ» 4. Заполнить поля формы 5. Нажать кнопку «ОК» | 1. Открывается журнал «Приказы». 2. В журнал загружаются «Приказы о приеме». 3. Открывается форма «Приказ о приеме» 4. Форма заполняется данными. 5. Происходит добавление приказа в базу данных. Форма приказа закрывается | Выполнено успешно |
| **9-ST-3-3** | Приказы | Добавление новой записи в приказы об увольнении | **Проверить работоспособность функции добавления приказа об увольнении.** 1. Открыть журнал «Приказы» 2. Выбрать переключатель «Приказы об увольнении» 3. Нажать кнопку «Новый приказ» 4. Заполнить поля формы 5. Нажать кнопку «ОК» | 1. Открывается журнал «Приказы». 2. В журнал загружаются «Приказы об увольнении». 3. Открывается форма «Приказы об увольнении» 4. Форма заполняется данными. 5. Происходит добавление приказа в базу данных. Форма приказа закрывается | Выполнено успешно |

Продолжение таблицы 5.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** |
| **10-ST-3-4** | Приказы | Добавление новой записи в приказы о переводе | **Проверить работоспособность функции добавления приказа об увольнении.** 1. Открыть журнал «Приказы» 2. Выбрать переключатель «Приказы о переводе» 3. Нажать кнопку «Новый приказ» 4. Заполнить поля формы 5. Нажать кнопку «ОК» | 1. Открывается журнал «Приказы». 2. В журнал загружаются «Приказы о переводе». 3. Открывается форма «Приказы о переводе» 4. Форма заполняется данными. 5. Происходит добавление приказа в базу данных. Форма приказа закрывается | Выполнено успешно |
| **11-ST-3-5** | Приказы | Добавление новой записи в приказы по штату | **Проверить работоспособность функции добавления приказа об увольнении.** 1. Открыть журнал «Приказы» 2. Выбрать переключатель «Приказы по штату» 3. Нажать кнопку «Новый приказ» 4. Заполнить поля формы 5. Нажать кнопку «ОК» | 1. Открывается журнал «Приказы». 2. В журнал загружаются «Приказы по штату». 3. Открывается форма «Приказы по штату» 4. Форма заполняется данными. 5. Происходит добавление приказа в базу данных. Форма приказа закрывается | Выполнено успешно |
| **12-ST-3-6** | Приказы | Редактирование приказа о приеме на работу | **Проверить работоспособность функции добавления приказа о приеме.** 1. Выбрать переключатель «Приказы о приеме» 2. Выбрать в журнале необходимый приказ 3. Нажать кнопку «Редактировать» 4. Отредактировать поля формы 5. Нажать кнопку «ОК» | 1. В журнал загружаются «Приказы о приеме». 2. Строка приказа выделяется 3. Открывается заполненная форма «Приказ о приеме» 4. Форма заполняется данными. 5. Происходит добавление измененного приказа в базу данных. Форма приказа закрывается. | Выполнено успешно |

Продолжение таблицы 5.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** |
| **13-ST-3-7** | Приказы | Редактирование приказа об увольнении | **Проверить работоспособность функции добавления приказа о приеме.** 1. Выбрать переключатель «Приказы об увольнении» 2. Выбрать в журнале необходимый приказ 3. Нажать кнопку «Редактировать» 4. Отредактировать поля формы 5. Нажать кнопку «ОК» | 1. В журнал загружаются «Приказы об увольнении». 2. Строка приказа выделяется 3. Открывается заполненная форма «Приказ об увольнении» 4. Форма заполняется данными. 5. Происходит добавление измененного приказа в базу данных. Форма приказа закрывается. | Выполнено успешно |
| **14-ST-3-8** | Приказы | Редактирование приказа о переводе | **Проверить работоспособность функции добавления приказа о приеме.** 1. Выбрать переключатель «Приказы о переводе» 2. Выбрать в журнале необходимый приказ 3. Нажать кнопку «Редактировать» 4. Отредактировать поля формы 5. Нажать кнопку «ОК» | 1. В журнал загружаются «Приказы о переводе». 2. Строка приказа выделяется 3. Открывается заполненная форма «Приказ о переводе» 4. Форма заполняется данными. 5. Происходит добавление измененного приказа в базу данных. Форма приказа закрывается. | Выполнено успешно |
| **15-ST-3-9** | Приказы | Редактирование приказа по штату | **Проверить работоспособность функции добавления приказа о приеме.** 1. Выбрать переключатель «Приказы по штату» 2. Выбрать в журнале необходимый приказ 3. Нажать кнопку «Редактировать» 4. Отредактировать поля формы 5. Нажать кнопку «ОК» | 1. В журнал загружаются «Приказы по штату». 2. Строка приказа выделяется 3. Открывается заполненная форма «Приказ по штату» 4. Форма заполняется данными. 5. Происходит добавление измененного приказа в базу данных. Форма приказа закрывается. | Выполнено успешно |

Продолжение таблицы 5.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** |
| **16-ST-4-1** | Работа с личными карточками | Редактирование данных в личной карточке | **Проверить работоспособность кнопки «Просмотр карточки сотрудника»** 1. Открыть журнал «Сотрудники» 2. Выбрать в журнале необходимого сотрудника 3. Нажать кнопку «Просмотр карточки сотрудника» 4. Установить переключатель «Редактировать» 5. Отредактировать необходимые поля в карточке 6. Нажать кнопку «ОК» | 1. Открывается журнал «Сотрудники». 2. Строка выбранного сотрудника выделяется 3. Открывается форма «Личная карточка сотрудника». 4. Поля формы становятся доступными для редактирования 5. Форма заполняется новыми данными 6. Происходит обновление базы данных. Форма личной карточки закрывается. | Выполнено успешно |
| **17-ST-4-2** | Работа с личными карточками | Вывод на печать личной карточки сотрудника | **Проверить работоспособность кнопки «Печать» в карточке сотрудника** 1. Открыть карточку сотрудника 2. Нажать кнопку «Печать» в нижнем левом углу формы 3. Закрыть карточку сотрудника | 1. Открывается форма «Личная карточка сотрудника». 2. Происходит открытие шаблона в Excel и заполнение его данными из карточки сотрудника 3. Форма личной карточки закрывается. | Выполнено успешно |
| **18-ST-4-3** | Работа с личными карточками | Добавление данных о воинском учете | **Проверить работоспособность функции добавления данных о воинском учете** 1. Открыть карточку сотрудника 2. Установить переключатель «Редактировать» 3. Перейти на вкладку «Воинский учет» 4. Установить переключатель «Военнообязанный». 5. Заполнить поля формы 6. Нажать кнопку «ОК» | 1. Открывается форма «Личная карточка сотрудника». 2. Поля формы становятся доступными для редактирования. 3. Открывается вкладка «Воинский учет» 4. Отображаются поля воинского учета. 5. Форма заполняется данными. 6. Происходит обновление базы данных. Форма личной карточки закрывается. | Выполнено успешно |

Продолжение таблицы 5.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** |
| **19-ST-4-4** | Работа с личными карточками | Редактирование состава семьи сотрудника | **Проверить работоспособность функции редактирования данных о составе семьи** 1. Открыть карточку сотрудника 2. Установить переключатель «Редактировать». 3. Перейти на вкладку «Состав семьи» 4. Нажать на кнопку «Добавить». 5. Заполнить поля формы. 6. Нажать кнопку «ОК». | 1. Открывается форма «Личная карточка сотрудника». 2. Поля формы становятся доступными для редактирования. 3. Открывается вкладка «Состав семьи». 4. Открывается форма добавления члена семьи. 5. Форма заполняется данными. 6. Член семьи добавляется в базу данных и в табличную часть вкладки «Состав семьи». | Выполнено успешно |
| **20-ST-4-5** | Работа с личными карточками | Добавление данных об образовании | **Проверить работоспособность функции добавления данных об образовании** 1. Открыть карточку сотрудника 2. Установить переключатель «Редактировать» 3. Перейти на вкладку «Образование» 4. Заполнить поля формы 5. Нажать кнопку «ОК» | 1. Открывается форма «Личная карточка сотрудника». 2. Поля формы становятся доступными для редактирования. 3. Открывается вкладка «Образование» 4. Форма заполняется данными. 5. Происходит обновление базы данных. Форма личной карточки закрывается. | Выполнено успешно |
| **21-ST-4-6** | Работа с личными карточками | Редактирование данных о трудовой деятельности | **Проверить работоспособность функции редактирования данных о трудовой деятельности** 1. Открыть карточку сотрудника 2. Установить переключатель «Редактировать» 3. Перейти на вкладку «Трудовая деятельность» 4. Отредактировать поля формы 5. Нажать кнопку «ОК» | 1. Открывается форма «Личная карточка сотрудника». 2. Поля формы становятся доступными для редактирования. 3. Открывается вкладка «Трудовая деятельность» 4. Форма заполняется данными. 5. Происходит обновление базы данных. Форма личной карточки закрывается. | Выполнено успешно |

Продолжение таблицы 5.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** |
| **22-ST-4-7** | Работа с личными карточками | Добавление данных о выходе на пенсию | **Проверить работоспособность функции добавления данных о выходе на пенсию** 1. Открыть карточку сотрудника 2. Установить переключатель «Редактировать» 3. Перейти на вкладку «Трудовая деятельность» 4. Установить переключатель «Пенсионер» 5. Заполнить поля выхода на пенсию. 5. Нажать кнопку «ОК» | 1. Открывается форма «Личная карточка сотрудника». 2. Поля формы становятся доступными для редактирования. 3. Открывается вкладка «Трудовая деятельность» 4. Отображаются поля выхода на пенсию. 5. Форма заполняется данными. 5. Происходит обновление базы данных. Форма личной карточки закрывается. | Выполнено успешно |
| **23-ST-5-1** | Штатное расписание | Создание штатного расписания | **Проверить работоспособность функции добавления штатного расписания.**  1. Открыть журнал «Штатное расписание». 2. Нажать на кнопку «Новое штатное расписание». 3. Открыть для выбора приказы по штатному расписанию, нажав на кнопку «Утверждено приказом». 4. Выбрать необходимый приказ и нажать кнопку «Выбрать». 5. Заполнить дату и ставку первого разряда. 6. Нажать кнопку «Новая строка».  7. Нажать «Да» 8. Заполнить поля формы элемента. 9. Нажать кнопку «ОК». | 1. Открывается журнал «Штатное расписание». 2. Открывается форма штатного расписания.  3. Открывается форма со списком приказов по штатному расписанию. 4. Заполняются номер и дата приказа. 5. Форма заполняется данными. 6. Появляется сообщение «Создать штатное расписание? Да/Нет» 7. Происходит добавление штатного расписания в базу данных. Открывается форма элемента штатного расписания. 9. Происходит добавление элемента в базу данных. Форма элемента закрывается. | Выполнено успешно |

Заключение

Автоматизация отдела кадров должна избавить от выполнения рутинных операций при работе с кадрами, ведении табелей, дальнейшего учета рабочего времени. Автоматизированное хранение и обработка полной кадровой информации также позволяет эффективно осуществлять подбор и перемещение сотрудников.

С учётом проведённого выше анализа системы управления персоналом можно сделать следующий вывод: предлагаемые на рынке автоматизированные системы управления персоналом, либо недостаточно гибки, либо обладают излишней мощностью.

При этом в любом случае требуется серьезная настройка программы под конкретные требования, а также обязательное дальнейшее программное сопровождение, что выразится в ощутимых финансовых и временных затратах. Поэтому для ОАО «Кузлитмаш» целесообразно было создать новую АС, с учетом его специфики предприятия, способную решить поставленные задачи и быть приемлемой по цене.

Для успешного ведения кадрового учета, программный продукт, призванный автоматизировать эту работу, должен:

* облегчить заключение трудовых договоров при приеме на работу новых сотрудников;
* формирование, редактирование и печать личной карточки сотрудника;
* формирование и редактирование справочников должностей и подразделений организации;
* ведение штатного расписания;
* кадровое перемещение и увольнение сотрудников;
* учет вакансий;
* расчет стажа и отпусков;
* поиск по журналу сотрудников;
* формирование и печать необходимой отчетности.

Программный продукт АРМ «Отдел кадров» соответствует всем указанным выше критериям и реализует весь необходимый функционал, в рамках поставленного задания.

Автоматизированная система управления персоналом является источником, из которого можно получить информацию различного характера для всех заинтересованных лиц, в том числе руководства, специалистов отделов, контролирующих органов.

Таким образом, создание автоматизированной системы, позволит:

* повысить производительность труда;
* уменьшить затраты на содержание отдела кадров;
* повысить качество и достоверность кадрового учета;
* перейти от учета к анализу кадров предприятия;
* привести кадровый учет предприятия в порядок и в соответствие с законодательными документами.

Некоторые задачи, особенно отчетность, можно будет решать намного быстрее. Снизится количество ошибок пользователей в связи с тем, что функции контроля возлагаются на программу. Повысится обоснованность принимаемых руководством решений на основе отчетов отдела кадров, так как повысится достоверность и полнота кадрового учета.

Данная система позволит перейти от простого учета кадров на более высокий уровень. Будет возможно производить анализ движения кадров по разным аспектам, на основе исходных данных легко получать нужную управленческую информацию. Всё вышеизложенное позволяет сделать вывод о целесообразности разработки.

Так, например, для системы целесообразно будет добавление бухгалтерского модуля с возможностями проведения учета рабочего времени, учета и расчета отпусков, расчета заработной платы. Это позволит существенно упростить работу не только кадровика, но и бухгалтерии предприятия. А так же позволит автоматизировать предоставление отчетности в фонд соцзащиты и госстрах.

В результате разработки данного программного продукта, снизилась трудоемкость выполнения задач, существенно повысилось качество и производительность работы отдела кадров ОАО «Кузлитмаш», что и будет являться основным результатом внедрения программного продукта.

Список использованных источников

1. Агашкова, А. А. Формирование кадровой политики организации: практические рекомендации / А. А. Агашкова, Е. П. Пархимчик. – Мн.: Регистр, 2010. — 288 с.
2. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Дж. Рамбо, А. Джекобсон. – Санкт-Петербург: Питер, 2007. – 432 с.
3. Дейт, К. Введение в системы управления базами данных / К. Дейт. – Москва: Бином, 2007. – 565 с.
4. Дубейковский, В. И. Эффективное моделирование с AllFusion Process Modeler 4.1.4 и AllFusion PM / В. И. Дубейковский. – Москва: Нолидж, 2007. – 284 с.
5. Ендовицкий, Д. А. Комплексный экономический анализ деятельности управленческого персонала / Д. А. Ендовицкий. – СПб.: [КноРус](http://mq2.ru/catalogue/category/vse-knigi?mkey=producer&mval=%D0%9A%D0%BD%D0%BE%D0%A0%D1%83%D1%81), 2011. – 192 с.
6. Каймакова, М. В. Анализ использования человеческих ресурсов / М. В. Каймакова. – Ульяновск : УлГТУ, 2010. – 80 с.
7. Киммел, П. UML. Основы визуального анализа и проектирования / П. Киммел. – Москва: Нолидж, 2008. – 237 с.
8. Кобайло, А. С. Rational Rose для моделирования информационных систем: методическое пособие / А. С. Кобайло, Н. А. Жиляк. – Минск: БГТУ, 2007. – 316 с.
9. Магура, М. И. Оценка работы персонала: подготовка и проведение аттестации / М. И. Магура, М. Б. Курбатова. – СПб.: Бизнес-школа Интел-синтез, 2012. – 144 с.
10. Маклаков, С. В. BPwin и ERwin. CASE-средства разработки / С. В. Маклаков. – Москва: Диалог МИФИ, 2002. – 224 с.
11. Молинаро Э. SQL Сборник рецептов; пер. с англ. – Спб.: «Символ-Плюс», 2009. – 672 с.
12. Мартин, Дж. Организация баз данных в вычислительных системах / Дж. Мартин. – Москва: Мир, 2005. – 450 с.

Приложение А  
(обязательное)  
Программный код добавления данных приказа о приеме на работу

try

{

if (JournalOfEmployees.conditionStringOrder == "новый")

{

// 1. Создание ObjectContext

OKDBEntities context = new OKDBEntities();

// 2. Создание экземпляра сущностного объекта

ПриемНаРаботу orderIN = new ПриемНаРаботу

{

Табельный = Convert.ToInt32(tbTabN.Text),

Приказ = tbOrderN.Text,

ДатаПриказа = dtPickerOrderDate.Value,

ДатаПриемаНаРаботу = dtPickerDateIn.Value,

КодПодразделенияКуда = callsMethods.GetOfficeKodFromOfficeName(comboBoxOffice.Text),

КодДолжностиКуда = callsMethods.GetPostKodFromPostName(comboBoxPost.Text),

Примечание = textBox10.Text

};

Сотрудники employees = new Сотрудники

{

Табельный = Convert.ToInt32(tbTabN.Text),

Фамилия = tbSurname.Text,

Имя = tbName.Text,

Отчетство = tbPatronymic.Text,

КодПодразделения = callsMethods.GetOfficeKodFromOfficeName(comboBoxOffice.Text),

КодДолжности = callsMethods.GetPostKodFromPostName(comboBoxPost.Text),

ТекущееПодразделение = callsMethods.GetOfficeKodFromOfficeName(comboBoxOffice.Text),

ТекущаяДолжность = callsMethods.GetPostKodFromPostName(comboBoxPost.Text),

};

ЛичныеДанные personalData = new ЛичныеДанные

{

Табельный = Convert.ToInt32(tbTabN.Text)

};

Образование education = new Образование

{

Табельный = Convert.ToInt32(tbTabN.Text)

};

// 3. Добавление сущностного объекта в таблицу

context.ПриемНаРаботу.Add(orderIN);

context.Сотрудники.Add(employees);

context.ЛичныеДанные.Add(personalData);

context.Образование.Add(education);

// 4. Вызов метода SaveChanges

context.SaveChanges();

int tbTabN\_Text = Convert.ToInt32(tbTabN.Text);

// Проверяем наличие вставленной записи

ПриемНаРаботу getrowIn = (from c in context.ПриемНаРаботу

where c.Табельный == tbTabN\_Text

select c).SingleOrDefault<ПриемНаРаботу>();

if (getrowIn == null) MessageBox.Show("getrowIn не");

else Console.WriteLine("getrowIn ok");

Сотрудники getrowEmp = (from c in context.Сотрудники

where c.Табельный == tbTabN\_Text

select c).SingleOrDefault<Сотрудники>();

if (getrowEmp == null) MessageBox.Show("getrowEmp не");

else Console.WriteLine("getrowEmp ok");

ЛичныеДанные getrowPD = (from c in context.ЛичныеДанные

where c.Табельный == tbTabN\_Text

select c).SingleOrDefault<ЛичныеДанные>();

if (getrowPD == null) MessageBox.Show("getrowPD не");

else Console.WriteLine("getrowPD ok");

Образование getrowEd = (from c in context.Образование

where c.Табельный == tbTabN\_Text

select c).SingleOrDefault<Образование>();

if (getrowEd == null) MessageBox.Show("getrowEd не");

else Console.WriteLine("getrowEd ok");

}

else//если не новый, а просмотр, тогда обновить

{

MessageBox.Show(JournalOfEmployees.conditionStringOrder);

//обновляем таблицу ПриемНаРаботу

IQueryable<ПриемНаРаботу> Table = from cellvalue in MainForm.ОтделКадров.ПриемНаРаботу

select cellvalue;

foreach (ПриемНаРаботу rowCell in Table)

{

if (rowCell.Приказ == tbOrderN.Text)//тогда обновляем

{

rowCell.ДатаПриказа = dtPickerOrderDate.Value;

rowCell.ДатаПриемаНаРаботу = dtPickerDateIn.Value;

rowCell.КодПодразделенияКуда = callsMethods.GetOfficeKodFromOfficeName(comboBoxOffice.Text);

rowCell.КодДолжностиКуда = callsMethods.GetPostKodFromPostName(comboBoxPost.Text);

rowCell.Примечание = textBox10.Text;

MainForm.ОтделКадров.SaveChanges();

}

}

//обновляем таблицу Сотрудники

IQueryable<Сотрудники> TableEmp = from cellvalue in MainForm.ОтделКадров.Сотрудники

select cellvalue;

foreach (Сотрудники rowCell in TableEmp)

{

if (rowCell.Табельный == Convert.ToInt32(tbTabN.Text))//тогда обновляем

{

rowCell.Фамилия = tbSurname.Text;

rowCell.Имя = tbName.Text;

rowCell.Отчетство = tbPatronymic.Text;

rowCell.КодПодразделения = callsMethods.GetOfficeKodFromOfficeName(comboBoxOffice.Text);

rowCell.КодДолжности = callsMethods.GetPostKodFromPostName(comboBoxPost.Text);

if (checkBoxUpdateDest.Checked == true)

{

rowCell.ТекущееПодразделение = callsMethods.GetOfficeKodFromOfficeName(comboBoxOffice.Text);

rowCell.ТекущаяДолжность = callsMethods.GetPostKodFromPostName(comboBoxPost.Text);

}

MainForm.ОтделКадров.SaveChanges();

}

}

}

MainForm.ОтделКадров.SaveChanges();

MessageBox.Show("Приказ о приёме добавлен/изменён");

//чтоб оставаться на позиции

int currentRow = MainForm.journalOfEmployees.dataGridViewOrders.CurrentRow.Index;

MainForm.journalOfEmployees.JournalOfEmployees\_Load(sender, e);//обновить датаГрид

MainForm.journalOfEmployees.dataGridViewOrders.CurrentCell = MainForm.journalOfEmployees.dataGridViewOrders.Rows[currentRow].Cells[2];

Close();

}

catch (DbUpdateException ex)

{

MessageBox.Show(ex.ToString());

}

}

Приложение Б  
(обязательное)  
Программный код заполнения личной карточки сотрудника

private void btnOK\_Click(object sender, EventArgs e)

{

try

{

UpdateGeneralInformationEmployeeTab();

UpdateFamilyTab();

UpdateEducationTab();

UpdateWorkActivityTab();

UpdateMilitaryRegistrationTab();

MainForm.ОтделКадров.SaveChanges();

MessageBox.Show("Объект изменён");

//чтоб оставаться на позиции

int currentRow = MainForm.journalOfEmployees.dataGridViewJournalOfEmp.CurrentRow.Index;

MainForm.journalOfEmployees.JournalOfEmployees\_Load(sender, e);//обновить датаГрид

MainForm.journalOfEmployees.dataGridViewJournalOfEmp.CurrentCell = MainForm.journalOfEmployees.dataGridViewJournalOfEmp.Rows[currentRow].Cells[2];

MainForm.journalOfEmployees.FillLabels();//обновить лэйблы по выделенному сотруднику

//MessageBox.Show("Объект изменён");

Close();

}

catch (DbUpdateException ex)

{

MessageBox.Show(ex.InnerException.ToString());

//Close();

}

//MainForm.ОтделКадров.SaveChanges();

//MessageBox.Show("Объект изменён");

//Close();

}

Приложение В  
(обязательное)  
Программный код формирования структуры должности/подразделения

internal void GetOffices()

{

SelectPostOffice selectPostOffice = new SelectPostOffice();

DataTable dataTable = new DataTable();

dataTable.Columns.Add("Код", typeof(int));

dataTable.Columns.Add("Наименование", typeof(string));

IQueryable<Подразделения> offices = from cellvalue in MainForm.ОтделКадров.Подразделения

select cellvalue;

foreach (Подразделения rowCell in offices)

{

dataTable.Rows.Add(new object[] {

rowCell.КодПодразделения,

rowCell.Наименование

});

selectPostOffice.dataGridView1.DataSource = dataTable; }

SelectPostOffice.rowCount = selectPostOffice.dataGridView1.RowCount;

selectPostOffice.dataGridView1.Columns[0].Width = 50;

selectPostOffice.dataGridView1.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

selectPostOffice.ShowDialog();

}

internal void GetPosts()

{

SelectPostOffice selectPostOffice = new SelectPostOffice();

DataTable dataTable = new DataTable();

dataTable.Columns.Add("Код", typeof(int));

dataTable.Columns.Add("Наименование", typeof(string));

IQueryable<Должности> posts = from cellvalue in MainForm.ОтделКадров.Должности

select cellvalue;

foreach (Должности rowCell in posts)

{

dataTable.Rows.Add(new object[] {

rowCell.КодДолжности,

rowCell.Наименование

});

selectPostOffice.dataGridView1.DataSource = dataTable; }

SelectPostOffice.rowCount = selectPostOffice.dataGridView1.RowCount;

selectPostOffice.dataGridView1.Columns[0].Width = 50;

selectPostOffice.dataGridView1.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

selectPostOffice.ShowDialog();

}

internal void AddPostOfficeInTree(string PostOffice)

{

switch (param)

{

case "Подразделения":

treeViewPostOffice.BeginUpdate();

treeViewPostOffice.Nodes.Add(PostOffice.Trim());

treeViewPostOffice.EndUpdate(); break;

case "Должности":

if (treeViewPostOffice.SelectedNode != null)

{

treeViewPostOffice.BeginUpdate();

treeViewPostOffice.SelectedNode.Nodes.Add(PostOffice.Trim());

treeViewPostOffice.ExpandAll(); } break;

}

TreePostOfficeSave(); }

Приложение Г  
(обязательное)  
Программный код создания штатного расписания

private void CreateStaffList()

{

try

{

// 1. Создание ObjectContext

OKDBEntities context = new OKDBEntities();

// 2. Создание экземпляра сущностного объекта

ШтатноеРасписание staffList = new ШтатноеРасписание

{

НомерРасписания = Convert.ToInt32(tbStaffN.Text),

ДатаРасписания = dtPickerStaffDate.Value,

НомерПриказа = Convert.ToInt32(tbStaffOrder.Text),

ДатаПриказа = dtPickerOrderDate.Value,

Ставка1 = Convert.ToInt32(tbTarif1.Text),

ДатаСтавки1 = dtPickerDateTarif.Value

};

if (JournalOfEmployees.conditionStringStaff == "новый")

{

// 3. Добавление сущностного объекта в таблицу

context.ШтатноеРасписание.Add(staffList);

MessageBox.Show("Добавлено новое штатное расписание");

}

else if (JournalOfEmployees.conditionStringStaff == "просмотр")

{

MainForm.staffList.НомерРасписания = Convert.ToInt32(tbStaffN.Text);

MessageBox.Show("расписание №" + MainForm.staffList.НомерРасписания.ToString());

MainForm.staffList.ДатаРасписания = dtPickerStaffDate.Value;

MainForm.staffList.НомерПриказа = Convert.ToInt32(tbStaffOrder.Text);

MainForm.staffList.ДатаПриказа = dtPickerOrderDate.Value;

MainForm.staffList.Ставка1 = Convert.ToInt32(tbTarif1.Text);

MainForm.staffList.ДатаСтавки1 = dtPickerDateTarif.Value;

MainForm.ОтделКадров.Entry(MainForm.staffList).State = EntityState.Modified;

MainForm.ОтделКадров.SaveChanges();

MessageBox.Show("Штатное расписание изменено");

}

else if (MainForm.conditionStringStaff == "новый")

{

context.ШтатноеРасписание.Add(staffList);

MessageBox.Show("Добавлено новое штатное расписание");

}

// 4. Вызов метода SaveChanges

context.SaveChanges();

int tbStaffN\_Text = Convert.ToInt32(tbStaffN.Text);

// Проверяем наличие вставленной записи

ШтатноеРасписание stafList = (from c in context.ШтатноеРасписание

where c.НомерРасписания == tbStaffN\_Text

select c).SingleOrDefault<ШтатноеРасписание>();

if (stafList == null) MessageBox.Show("не");

else MessageBox.Show("ok");

}

catch (DbUpdateException ex) { MessageBox.Show(ex.ToString()); return; }

}

private void btnOK\_Click(object sender, EventArgs e)

{

try

{

// 1. Создание ObjectContext

OKDBEntities context = new OKDBEntities();

// 2. Создание экземпляра сущностного объекта

СоставШтатного staffMembers = new СоставШтатного

{

ID = Convert.ToInt32(tbID.Text),

НомерРасписания = Convert.ToInt32(tbStaffN.Text),

Должность = comboBoxPost.Text,

КоличествоШтатныхЕдиниц = Convert.ToDouble(tbKol.Text),

ЗанятоВакансий = Convert.ToDouble(tbZanVac.Text),

ТарифныйРазряд = Convert.ToInt32(comboBoxRazriad.Text),

ТарифныйКоэфициент = coefficient,

ДолжностнойОклад = oklad

};

if (this.Text == "Добавить строку в штатное расписание")

{

// 3. Добавление сущностного объекта в таблицу

context.СоставШтатного.Add(staffMembers);

MessageBox.Show("Новая строка добавлена");

}

else

{

MainForm.staffMembers.ID = Convert.ToInt32(tbID.Text);

MainForm.staffMembers.НомерРасписания = Convert.ToInt32(tbStaffN.Text);

MainForm.staffMembers.Должность = comboBoxPost.Text;

MainForm.staffMembers.КоличествоШтатныхЕдиниц = Convert.ToDouble(tbKol.Text);

MainForm.staffMembers.ЗанятоВакансий = Convert.ToDouble(tbZanVac.Text);

MainForm.staffMembers.ТарифныйРазряд = Convert.ToInt32(comboBoxRazriad.Text);

MainForm.staffMembers.ТарифныйКоэфициент = coefficient;

MainForm.staffMembers.ДолжностнойОклад = oklad;

MainForm.ОтделКадров.Entry(MainForm.staffMembers).State = EntityState.Modified;

MainForm.ОтделКадров.SaveChanges();

MessageBox.Show("Строка изменена");

}

// 4. Вызов метода SaveChanges

context.SaveChanges();

int tbID\_Text = Convert.ToInt32(tbID.Text);

// Проверяем наличие вставленной записи

СоставШтатного getcust = (from c in context.СоставШтатного

where c.ID == tbID\_Text

select c).SingleOrDefault<СоставШтатного>();

if (getcust == null) MessageBox.Show("не");

else MessageBox.Show("ok");

}

catch (InvalidOperationException ex)

{

MessageBox.Show("Строка не добавлена");

MessageBox.Show(ex.ToString());

}

finally { /\*MessageBox.Show("btnOK\_Click finally");\*/ Close(); }

staffListForm.FillDataGridViewStaffList();//обновить датаГрид

}

Приложение Д  
(обязательное)  
Программный код приказа о переводе сотрудника

private void btnOK\_Click(object sender, EventArgs e)

{

try

{

if (JournalOfEmployees.conditionStringOrder == "новый")

{

// 1. Создание ObjectContext

OKDBEntities context = new OKDBEntities();

// 2. Создание экземпляра сущностного объекта

Перевод orderTransfer = new Перевод

{

Приказ = tbOrderN.Text,

ДатаПриказа = dtPickerOrderDate.Value,

Табельный = Convert.ToInt32(lblTabN.Text),

КодПодразделенияОткуда = callsMethods.GetOfficeKodFromOfficeName(lblOffice.Text),

КодДолжностиОткуда = callsMethods.GetPostKodFromPostName(lblPost.Text),

КодПодразделенияКуда = callsMethods.GetOfficeKodFromOfficeName(comboBoxOffice.Text),

КодДолжностиКуда = callsMethods.GetPostKodFromPostName(comboBoxPost.Text),

Причина = textBox10.Text

};

// 3. Добавление сущностного объекта в таблицу

context.Перевод.Add(orderTransfer);

MessageBox.Show("Новая строка добавлена");

// 4. Вызов метода SaveChanges

context.SaveChanges();

int lblTabN\_Text = Convert.ToInt32(lblTabN.Text);

// Проверяем наличие вставленной записи

Перевод getcust = (from c in context.Перевод

where c.Табельный == lblTabN\_Text

select c).SingleOrDefault<Перевод>();

//Console.WriteLine("\n Новый приказ {0} вставлен в базу данных", getcust == null ? "не" : "успешно");

if (getcust == null) MessageBox.Show("не");

else MessageBox.Show("ok");

}

else

{

MessageBox.Show(JournalOfEmployees.conditionStringOrder);

IQueryable<Перевод> Table = from cellvalue in MainForm.ОтделКадров.Перевод

select cellvalue;

foreach (Перевод rowCell in Table)

{

if (rowCell.Приказ == tbOrderN.Text)//тогда обновляем

{

MessageBox.Show("Проверка перевода (" + tbOrderN.Text + "). Табельный " + rowCell.Приказ.ToString() + " есть в таблице!");

//rowCellT1.Приказ = tbOrderN.Text;

rowCell.ДатаПриказа = dtPickerOrderDate.Value;

//rowCellT1.КодДолжностиОткуда = GetPostKodFromPostName(lblPost.Text);

//rowCellT1.КодПодразделенияОткуда = GetOfficeKodFromOfficeName(lblOffice.Text);

rowCell.КодДолжностиКуда = callsMethods.GetPostKodFromPostName(comboBoxPost.Text);

rowCell.КодПодразделенияКуда = callsMethods.GetOfficeKodFromOfficeName(comboBoxOffice.Text);

rowCell.Причина = textBox10.Text;

MainForm.ОтделКадров.SaveChanges();

}

}

}

//обновить данные в таблице employees

IQueryable<Сотрудники> TableEmp = from cellvalue in MainForm.ОтделКадров.Сотрудники

select cellvalue;

foreach (Сотрудники rowCell in TableEmp)

{

if (rowCell.Табельный == Convert.ToInt32(lblTabN.Text))//тогда обновляем

{

rowCell.ТекущееПодразделение = callsMethods.GetOfficeKodFromOfficeName (comboBoxOffice.Text);

rowCell.ТекущаяДолжность = callsMethods.GetPostKodFromPostName(comboBoxPost.Text);

}

}

MainForm.ОтделКадров.SaveChanges();

//чтоб оставаться на позиции

int currentRow = MainForm.journalOfEmployees.dataGridViewOrders.CurrentRow.Index;

MainForm.journalOfEmployees.JournalOfEmployees\_Load(sender, e);//обновить датаГрид

MainForm.journalOfEmployees.rBtnOrderTrans\_CheckedChanged(sender, e);

MainForm.journalOfEmployees.dataGridViewOrders.CurrentCell = MainForm.journalOfEmployees.dataGridViewOrders.Rows[currentRow].Cells[2];

Close();

}

catch (DbUpdateException ex)

{

MessageBox.Show("Приказ не добавлен в базу данных.");

MessageBox.Show(ex.ToString());

} }

Приложение Е  
(обязательное)  
Программный код приказа об увольнении сотрудника

private void btnOK\_Click(object sender, EventArgs e)

{ try {

if (JournalOfEmployees.conditionStringOrder == "новый")

{

// 1. Создание ObjectContext

OKDBEntities context = new OKDBEntities();

// 2. Создание экземпляра сущностного объекта

Увольнение orderOUT = new Увольнение

{

Табельный = Convert.ToInt32(lblTabN.Text),

Приказ = tbOrderN.Text,

ДатаПриказа = dtPickerOrderDate.Value,

ДатаУвольнения = dtPickerDateOut.Value,

КодПодразделенияОткуда = callsMethods.GetOfficeKodFromOfficeName(lblOffice.Text),

КодДолжностиОткуда = callsMethods.GetPostKodFromPostName(lblPost.Text),

Статья = comboBox1.Text

};

// 3. Добавление сущностного объекта в таблицу Customers

context.Увольнение.Add(orderOUT);

MessageBox.Show("Новая строка добавлена");

// 4. Вызов метода SaveChanges

context.SaveChanges();

int tbTabN\_Text = Convert.ToInt32(lblTabN.Text);

// Проверяем наличие вставленной записи

Увольнение getrowOUT = (from c in context.Увольнение

where c.Табельный == tbTabN\_Text

select c).SingleOrDefault<Увольнение>();

if (getrowOUT == null) MessageBox.Show("getrowOUT не");

else MessageBox.Show("getrowOUT ok");

}

else//если не новый, а просмотр, тогда обновить

{ //обновляем таблицу Увольнение

IQueryable<Увольнение> Table = from cellvalue in MainForm.ОтделКадров.Увольнение

select cellvalue;

foreach (Увольнение rowCell in Table)

{ if (rowCell.Приказ == tbOrderN.Text)//тогда обновляем

{

MessageBox.Show("Увольнение (" + tbOrderN.Text + "). Табельный " + rowCell.Приказ.ToString() + " есть в таблице!");

rowCell.ДатаПриказа = dtPickerOrderDate.Value;

rowCell.ДатаУвольнения = dtPickerDateOut.Value;

rowCell.КодПодразделенияОткуда = callsMethods.GetOfficeKodFromOfficeName(lblOffice.Text);

rowCell.КодДолжностиОткуда = callsMethods.GetPostKodFromPostName(lblPost.Text);

rowCell.Статья = comboBox1.Text;

MainForm.ОтделКадров.SaveChanges();

}

}

}

MainForm.ОтделКадров.SaveChanges(); MessageBox.Show("Приказ об увольнении изменён");

//чтоб оставаться на позиции

int currentRow = MainForm.journalOfEmployees.dataGridViewOrders.CurrentRow.Index;

MainForm.journalOfEmployees.rBtnOrderOut\_CheckedChanged(sender, e);//обновить датаГрид

MainForm.journalOfEmployees.dataGridViewOrders.CurrentCell = MainForm.journalOfEmployees.dataGridViewOrders.Rows[currentRow].Cells[2];

Close();

}

catch (DbUpdateException ex)

{ MessageBox.Show(“ Приказ не добавлен в базу данных.”); MessageBox.Show(ex.ToString()); }

}