# Введение

Данный курсовой проект выполняется на тему «Разработка интерфейса информационной системы управления контрагентами организации».

Автоматизация учета контрагентов позволит сократить время на оформление и учет договоров, формирование документов и отчетов, вести электронный документооборот, устранить несоответствие данных в различных документах.

Целью курсового проекта является разработка программного продукта в среде Visual Studio на языке C#, позволяющее хранить данные о контрагентах, их договорах, расчетных счетах и кураторах в базе данных. Приложение помогает осуществлять поиск в таблицах, формировать и выводить отчеты о договорах на экран компьютера, на печатающее устройство или сохранять на жесткий диск в различных форматах.

# 1 Постановка задачи

## 1.1 Описание предметной области

Программное средство разработано для облегчения учета договоров заключенных организацией со своими контрагентами, получения быстрой справочной информации по договорам, контрагентам, их счетам и сотрудникам, формирования отчета по заключенным договорам с указанием куратора и его контактной информации, даты договора. Областью применения разрабатываемого приложения может являться небольшая организация, с любым основным видом деятельности, например небольшая торговая точка. Автоматизация учета договоров позволит быстро получать необходимую для руководства в работе информацию, упростить процесс контроля расхода денежных средств организации.

## 1.2 Актуальность решаемой задачи

Любая организация в своей деятельности так или иначе связана с контрагентами, будь то поставка сырья для производства или сбыт готовой продукции, или закупка и ремонт производственного оборудования. Существуют программы, реализующие учет контрагентов организации, например такие как 1С, Галактика, Парус и т.п., стоимость лицензии на использование таких программных средств высока для небольших организаций, и не всегда оправдана, а возможности, предоставляемые этими программами, превосходят потребности заказчика. Интерфейс лицензионных программных средств перегружен лишними элементами управления отвлекая пользователя от выполнения простых и несложных задач, что влечет за собой излишнюю нагрузку на пользователя. Применение программного средства, ориентированного на конкретную задачу, позволяет и сэкономить на средствах на лицензионные отчисления, и понизить нагрузку на оператора.

## 1.3 Характеристика решаемой задачи

Полное наименование программного средства «Учет контрагентов организации».

Краткое наименование программного средства: «ПС учета контрагентов».

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

Разработчик программного обеспечения – Верхов Михаил Геннадьевич, учащийся по специальности 2-40 01 01, «Программное обеспечение информационных технологий».

Основанием для проведения работ является задание на курсовое проектирование.

Информационная система для учета контрагентов будет выполнять следующие функции:

* хранение информации о контрагентах;
* хранение информации о банковских счетах и сотрудниках контрагентов;
* оформление договоров;
* формирование отчета по договорам.

Основными целями создания программного средства для учета являются:

* замещение ручного учета контрагентов, их данных и договоров на автоматизированный, что уменьшит затраты времени;
* повышение качества выполняемой работы, что уменьшит риск ошибок при редактировании информации.

Система представляет собой базу данных, содержащую информацию о контрагентах, их счетах и сотрудниках, договорах организации.

Объектом автоматизации является рабочее место бухгалтера организации.

Видом автоматизируемой деятельности являются задачи учета выданных договоров заключенных с контрагентами.

Критерием оценки достижения целей создания программного средства является упрощение ведения учета договоров, списка контрагентов организации.

Программное средство разрабатывается как однопользовательское программное обеспечение.

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

Структурой хранения данных будет являться собственная база данных. Клиентский интерфейс прикладной программы будет разрабатываться в Visual Studio 2019, поддерживающий интерфейс операционной системы Windows. Взаимодействие с пользователем будет осуществляться через главное диалоговое окно с вкладками, функционально разделяющими его на рабочие пространства.

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

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

* добавлять данные о контрагентах и их реквизитах и договорах;
* производить редактирование информации о договорах;

# 2 Проектирование программного продукта

## 2.1 Разработка модели данных

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

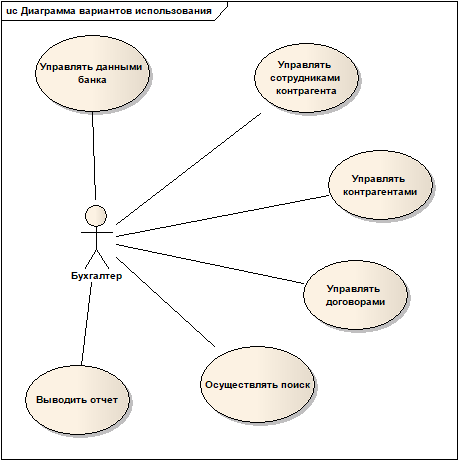


Рисунок 1 – Диаграмма вариантов использования

Описание вариантов использования представлено в таблице 1.

Таблица 1 – Реестр вариантов использования

|  |  |  |
| --- | --- | --- |
| Основной актер | Наименование варианта использования | Краткое описание |
| Бухгалтер | Управлять данными банка | Ввод, изменение и удаление данных о счетах |
| Бухгалтер | Управлять сотрудниками контрагента | Ввод, изменение и удаление данных о сотрудниках |
| Бухгалтер | Управлять контрагентами | Ввод, изменение и удаление данных о контрагентах |
| Бухгалтер | Управлять договорами | Ввод, изменение и удаление данных о договорах |
| Бухгалтер | Осуществлять поиск | Осуществление поиска по критериям в базе данных |
| Бухгалтер | Вывести отчет по договорам | Сформировать отчет по договорам |

Диаграмма классов форм и базы данных приложения представлена на рисунке 2.

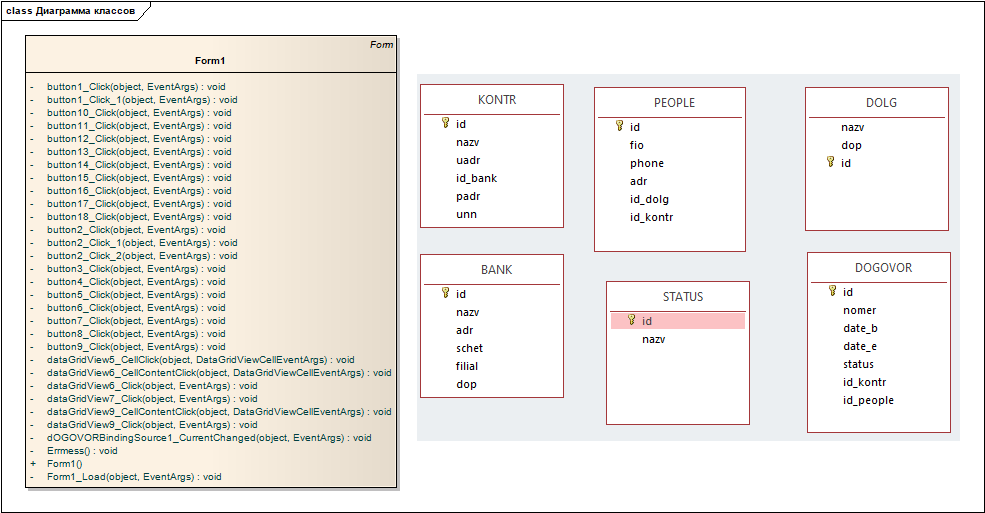


Рисунок 2 Диаграмма классов форм и базы данных приложения

Диаграмма состояния приложения представлена на рисунке 3.

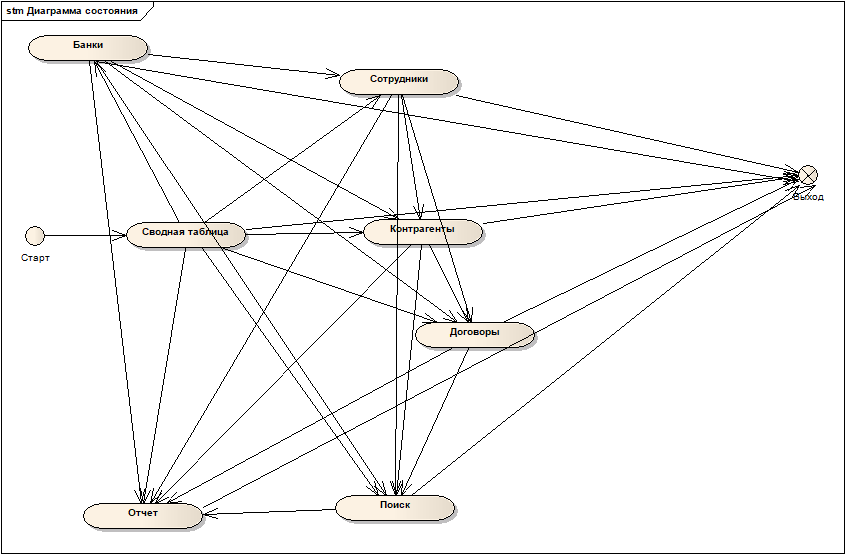


Рисунок 3. Диаграмма состояния приложения.

Диаграмма последовательности «Банки» представлена на рисунке 4.

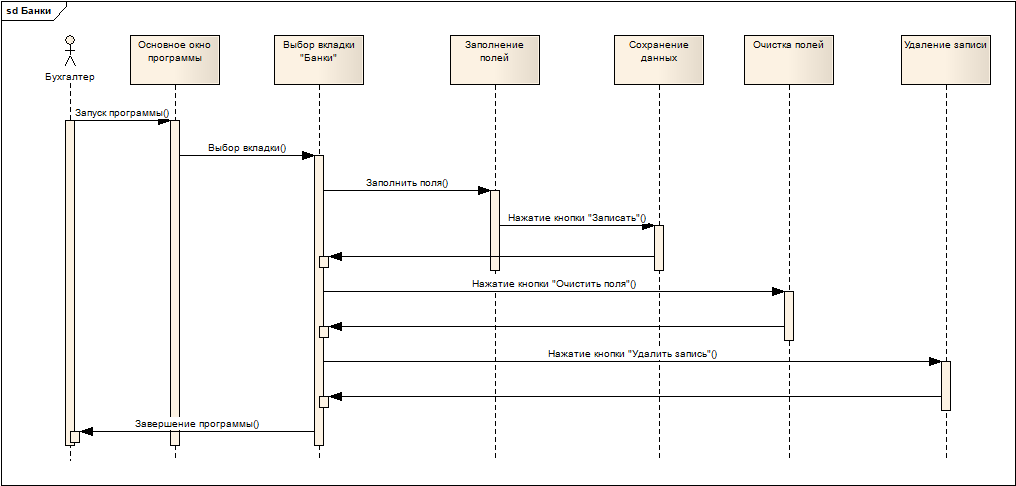


Рисунок 4. Диаграмма последовательности «Банки».

Таблица 2 - Описание процесса выполнения варианта использования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пользователь | Система | Экранная форма | Условие:  последующий шаг |
| 1 | Запуск программы | Запускается программа | Главная форма приложения | Ожидание действия |
| 2 | Выбор вкладки «Банки» | Открывается вкладка «Банки» | Вкладка приложения «Банки» | Ожидание действия |
| 3 | Внесение данных банка | Ожидание действия | Вкладка приложения «Банки» | Ожидание действия |
| 4 | Нажатие кнопки «Записать» | Осуществляет сохранение данных в базе данных | Вкладка приложения «Банки» | Ожидание действия |
| 5 | Нажатие кнопки «Очистить поля» | Очищает поля ввода экранной формы | Вкладка приложения «Банки» | Ожидание действия |
| 6 | Нажатие кнопки «Удалить запись» | Удаляет выделенную в табличной форме запись | Вкладка приложения «Банки» | Ожидание действия |
| 7 | Нажатие кнопки закрытия главной формы программы или выбор пункта «Закрыть» главного меню | Завершает работу приложения |  |  |

Диаграмма последовательности «Контрагенты» представлена на рисунке 5.

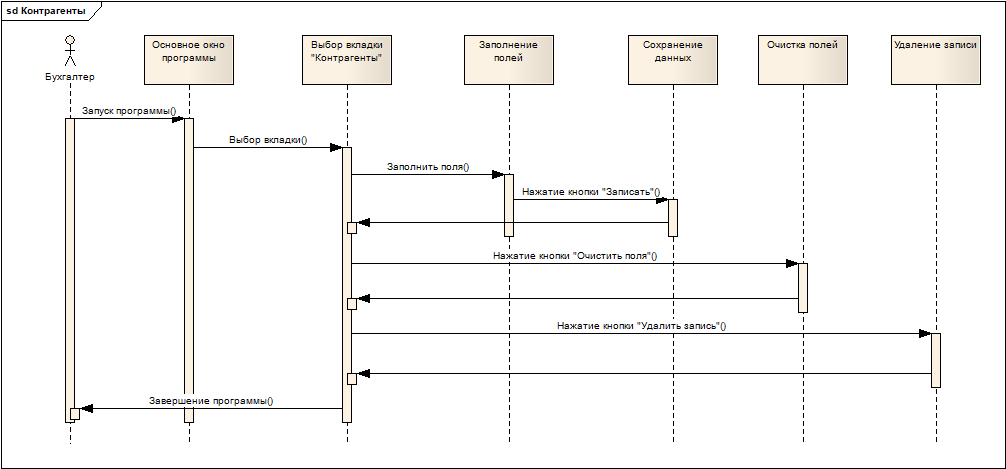


Рисунок 5. Диаграмма последовательности «Контрагенты».

Таблица 3 - Описание процесса выполнения варианта использования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пользователь | Система | Экранная форма | Условие:  последующий шаг |
| 1 | Запуск программы | Запускается программа | Главная форма приложения | Ожидание действия |
| 2 | Выбор вкладки «Контрагенты» | Открывается вкладка «Контрагенты» | Вкладка приложения «Контрагенты» | Ожидание действия |
| 3 | Внесение данных контрагента | Ожидание действия | Вкладка приложения «Контрагенты» | Ожидание действия |
| 4 | Нажатие кнопки «Записать» | Осуществляет сохранение данных в базе данных | Вкладка приложения «Контрагенты и» | Ожидание действия |
| 5 | Нажатие кнопки «Очистить поля» | Очищает поля ввода экранной формы | Вкладка приложения «Контрагенты» | Ожидание действия |
| 6 | Нажатие кнопки «Удалить запись» | Удаляет выделенную в табличной форме запись | Вкладка приложения «Контрагенты» | Ожидание действия |
| 7 | Нажатие кнопки закрытия главной формы программы или выбор пункта «Закрыть» главного меню | Завершает работу приложения |  |  |

Диаграмма последовательности «Сотрудники» представлена на рисунке 6.

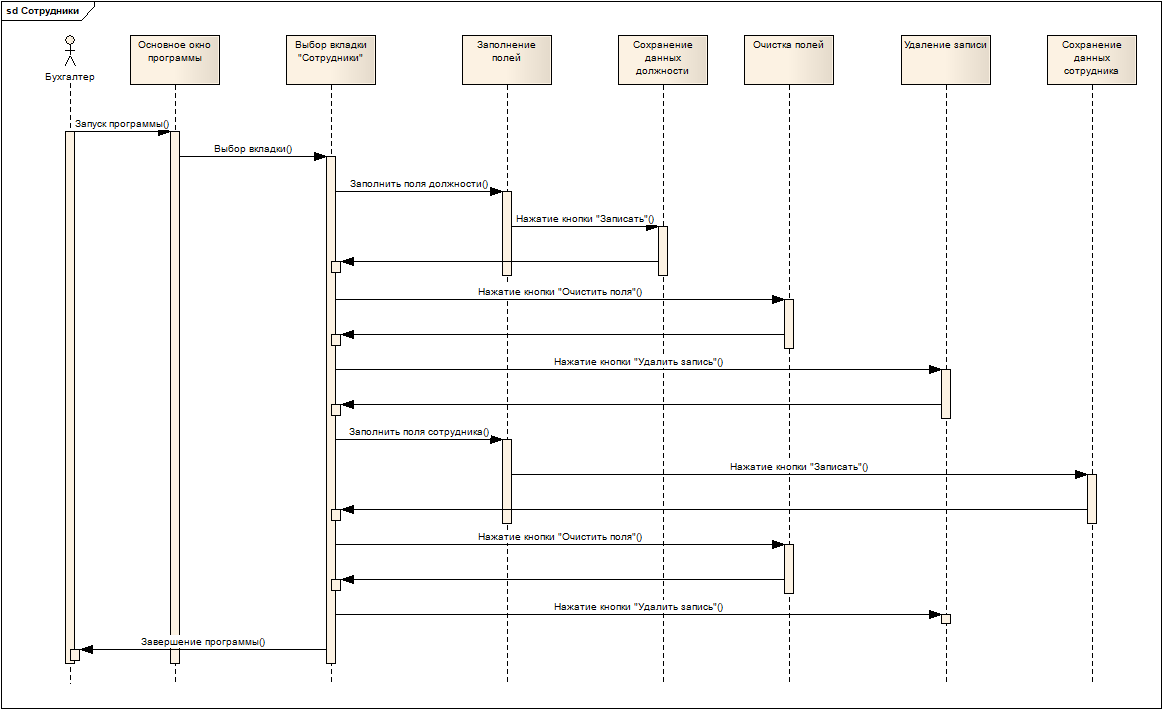


Рисунок 6. Диаграмма последовательности «Сотрудники».

Таблица 4 - Описание процесса выполнения варианта использования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пользователь | Система | Экранная форма | Условие:  последующий шаг |
| 1 | Запуск программы | Запускается программа | Главная форма приложения | Ожидание действия |
| 2 | Выбор вкладки «Сотрудники» | Открывается вкладка «Сотрудники» | Вкладка приложения «Сотрудники» | Ожидание действия |
| 3 | Внесение данных должности | Ожидание действия | Вкладка приложения «Сотрудники» | Ожидание действия |
| 4 | Нажатие кнопки «Записать» | Осуществляет сохранение данных в базе данных | Вкладка приложения «Сотрудники» | Ожидание действия |
| 5 | Нажатие кнопки «Очистить поля» | Очищает поля ввода экранной формы | Вкладка приложения «Сотрудники» | Ожидание действия |
| 6 | Нажатие кнопки «Удалить запись» | Удаляет выделенную в табличной форме запись | Вкладка приложения «Сотрудники» | Ожидание действия |
| 7 | Внесение данных сотрудника | Ожидание действия | Вкладка приложения «Сотрудники» | Ожидание действия |
| 8 | Нажатие кнопки «Записать» | Осуществляет сохранение данных в базе данных | Вкладка приложения «Сотрудники» | Ожидание действия |
| 9 | Нажатие кнопки «Очистить поля» | Очищает поля ввода экранной формы | Вкладка приложения «Сотрудники» | Ожидание действия |
| 10 | Нажатие кнопки «Удалить запись» | Удаляет выделенную в табличной форме запись | Вкладка приложения «Сотрудники» | Ожидание действия |
| 11 | Нажатие кнопки закрытия главной формы программы или выбор пункта «Закрыть» главного меню | Завершает работу приложения |  |  |

Диаграмма последовательности «Договоры» представлена на рисунке 7.

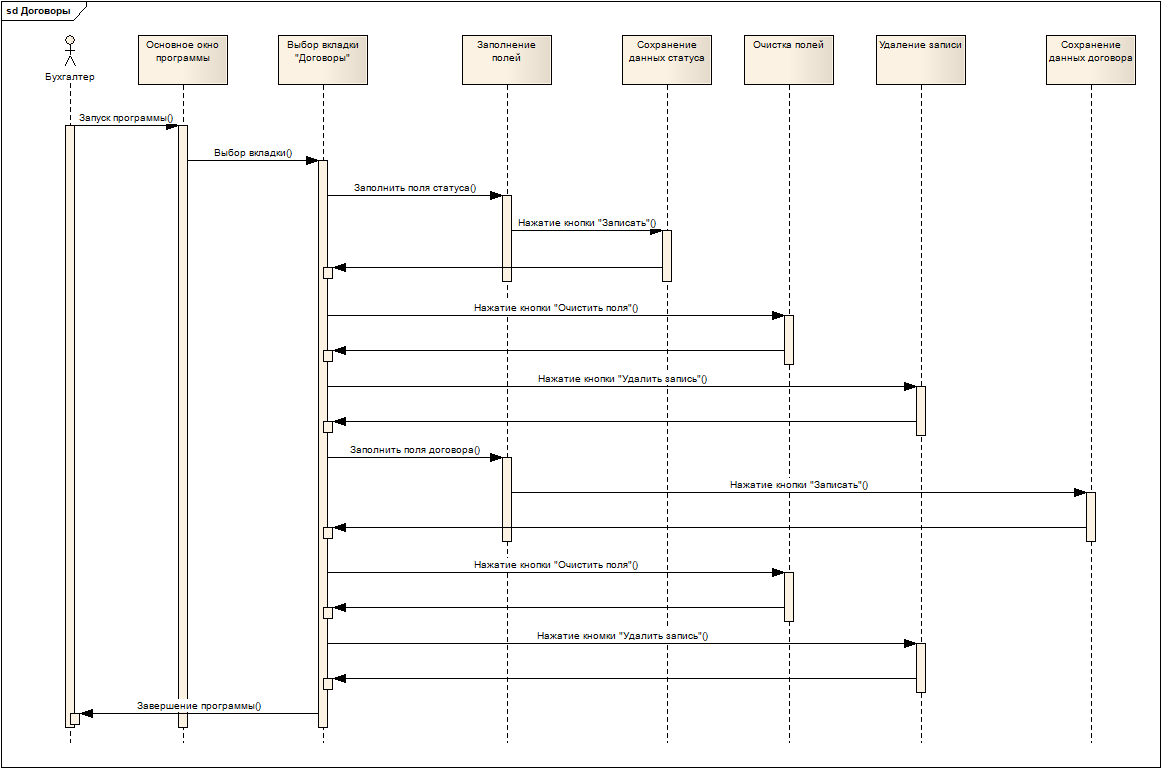


Рисунок 7. Диаграмма последовательности «Договоры».

Таблица 5 - Описание процесса выполнения варианта использования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Пользователь | Система | Экранная форма | Условие:  последующий шаг |
| 1 | Запуск программы | Запускается программа | Главная форма приложения | Ожидание действия |
| 2 | Выбор вкладки «Договоры» | Открывается вкладка «Договоры» | Вкладка приложения «Договоры» | Ожидание действия |
| 3 | Внесение данных статуса | Ожидание действия | Вкладка приложения «Договоры» | Ожидание действия |
| 4 | Нажатие кнопки «Записать» | Осуществляет сохранение данных в базе данных | Вкладка приложения «Договоры» | Ожидание действия |
| 5 | Нажатие кнопки «Очистить поля» | Очищает поля ввода экранной формы | Вкладка приложения «Договоры» | Ожидание действия |
| 6 | Нажатие кнопки «Удалить запись» | Удаляет выделенную в табличной форме запись | Вкладка приложения «Договоры» | Ожидание действия |
| 7 | Внесение данных договора | Ожидание действия | Вкладка приложения «Договоры» | Ожидание действия |
| 8 | Нажатие кнопки «Записать» | Осуществляет сохранение данных в базе данных | Вкладка приложения «Договоры» | Ожидание действия |
| 9 | Нажатие кнопки «Очистить поля» | Очищает поля ввода экранной формы | Вкладка приложения «Договоры» | Ожидание действия |
| 10 | Нажатие кнопки «Удалить запись» | Удаляет выделенную в табличной форме запись | Вкладка приложения «Договоры» | Ожидание действия |
| 11 | Нажатие кнопки закрытия главной формы программы или выбор пункта «Закрыть» главного меню | Завершает работу приложения |  |  |

Диаграмма последовательности «Поиск» представлена на рисунке 8.

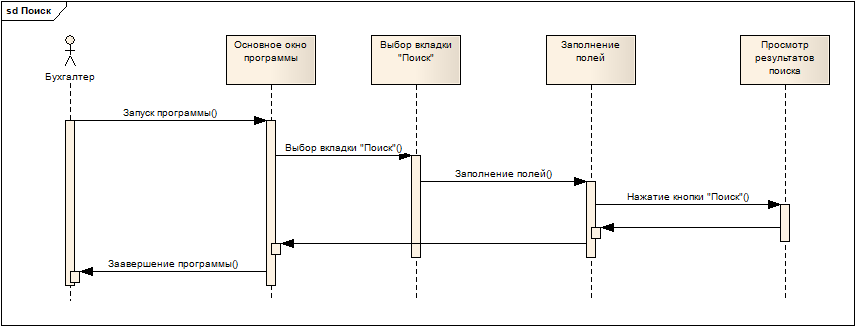


Рисунок 8. Диаграмма последовательности «Поиск».

Таблица 6 - Описание процесса выполнения варианта использования

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

Диаграмма последовательности «Отчет» представлена на рисунке 9.

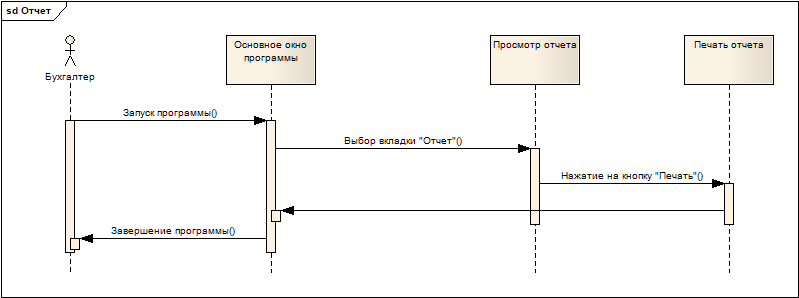


Рисунок 9. Диаграмма последовательности «Отчет».

Таблица 7 - Описание процесса выполнения варианта использования

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

## 2.2 Выбор программного обеспечения

Строгость, безопасность и надежность. Именно так характеризуют язык программирования Си шарп сторонники. Творение гиганта, корпорации Microsoft, неспроста пользуется такой репутацией среди разработчиков. Секрет успеха C# – обучение на недостатках предшественников. Язык взял лучшее от своих прародителей (Си++, Pascal, Модула, Smalltalk и Java), ликвидировав при этом проблемные места.

Язык программирования сочетает в себе объектно- и контекстно-ориентированные концепции. Он имеет строгую архитектуру со статической типизацией. Относится к семейству C-подобных. Его синтаксис схож с языками Си++ и Java. Он богат и разнообразен, но при этом прост в изучении. Особенно, если записаться на курсы C# в Днепре. Доверив свое образование профессионалам, вы улучшите качество знаний и ускорите освоение нужных для работы навыков.

Написание кода в среде Си шарп проще и понятнее, нежели в Си и Си++, так как создатели устранили многие неоднозначные места. При этом сам функционал системы более гибкий в сравнении с Java. В него включены недостающие возможности. Это такие нужные функции, как делегаты, перечисления, обнуляемые типы, прямой доступ к памяти и т.д.

Несомненными преимуществами среды C-шарп являются:

* удобный и функциональный редактор;
* продуманные пользовательские конструкторы;
* встроенный отладчик кода и прочие помощники разработчика.

Написанные на продвинутом языке программы выполняет платформа NET Framework, интегрированная в Windows. Для нее и был спроектирован C-шарп. Но нужно понимать, что понятие .NET гораздо шире обсуждаемой технологии. Платформа включает общеязыковую среду CLR и набор библиотек.

Язык называют одним из самых востребованных и перспективных в IT-индустрии. Его используют такие компании и корпорации:

* Microsoft (на Си-шарп написаны практически все продукты, разработанные и приобретенные);
* Siemens;
* Amazon;
* Oracle;
* Google;
* Tesco;
* Avanade;
* Ciklum;
* Softserve;
* AMC Bridge и другие.

Язык применяется в разработке всех видов продуктов: десктопных, мобильных и веб-приложений. С его помощью создаются игры для движков Unity 3D и CryEngine. Для разработки программ под Android и iOS подключаются дополнительные фреймворки Xamarin или Mono.

В среде возможна разработка любого компонента Windows. С недавних пор система работает также на Mac и Linux.

## 2.3 Определение требований к техническим средствам

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

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

- число ядер не менее 4;

2) системная плата с параметрами:

- форм-фактор - ATX;

- минимальный объем оперативной памяти - 4 Гб;

4) типы внешней памяти компьютера:

- минимальный объем жесткого диска: 100 ГБ;

6) параметры монитора:

- тип монитора - ЖК;

- ширина диагонали экрана – более 17 дюймов;

- разрешение по горизонтали и вертикали – не менее 1280x1024;

Диаграмма компонентов программного средства представлена на рисунке 10.

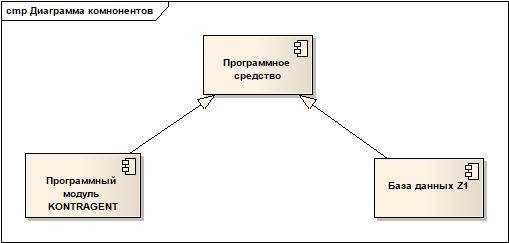


Рисунок 10 – Диаграмма компонентов программного средства.

# 3 Реализация программного модуля

## 3.1 Определение формы представления входных и выходных данных

При разработке программного модуля была выбрана модель, при которой в главном окне приложения отображается как сводная информация по путевкам, так выполнение дополнительных действий, таких как внесение и удаление вспомогательных данных, просмотр и печать отчетов, поиск в базе данных, все действия производятся в специальных вкладках приложения. Вкладки приложения разработаны однотипно, для того чтобы облегчить пользователю изучение и работу с приложением. На вкладках присутствуют поля ввода данных, поле отображения уже внесенной в таблицу информации и указания записи для удаления, и три кнопки – «Записать», «Стереть поля» и «Удалить запись» сохраняющие данные в базу, очищающие поля ввода и удаляющие запись, указанную в табличной части окна соответственно.

В качестве выходных форм, помимо сводной, главной вкладки приложения, выступают вкладки поиска и отчета. Вкладка поиска позволяет осуществить быстрый поиск в базе данных записей по указанным критериям, а вкладка отчета – сформировать отчет и вывести его, при необходимости, на печать. Вид вкладок и отчетов приложения размещен в приложении Б.

## 3.2 Тестирование программного средства

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

Таблица 8 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Примечание |
| Главная форма приложения | | | |
| Запуск приложения | Отображение главной формы приложения с заполненными таблицами из базы данных | Выполнено |
| Выбор контрагента в таблице контрагентов | Отображение в связанных таблицах соответствующей информации о контрагенте | Выполнено |
| Выбор вкладки для внесения изменений в таблицы БД | Открытие соответствующей действию вкладки приложения | Выполнено |
| Выбор вкладки «Отчет» | Открытие вкладки отчета, содержащей сформированный отчет по контрагентам | Выполнено |
| Выбор вкладки «Поиск» | Открытие вкладки поиска | Выполнено |
| Выбор пункта меню «Справка» | Открытие справочной системы приложения | Выполнено |
| Вкладки работы с базой данных | | | |
| Нажатие кнопки «Записать» | Внесение записи в базу данных, с проверкой пустых полей | Выполнено |
| Нажатие кнопки «Стереть» | Очистка полей ввода записи | Выполнено |
| Нажатие кнопки «Удалить» | Удаление записи выделенной в табличной части формы | Выполнено |
| Вкладка поиска | | | |
| Нажатие кнопки «Поиск» | Вывод результатов поиска в табличной части формы, в соответствии с указанными критериями | Выполнено |
| Нажатие кнопки «Стереть» | Очистка полей ввода поисковой строки | Выполнено |
| Вкладка отчета | | | |
| Нажатие кнопки «Печать» | Вывод отчета на принтер | Выполнено |
| Нажатие кнопки «Экспорт» | Сохранение отчета в указанном формате | Выполнено |

## 3.3 Защита информации

Разработка способов защиты от несанкционированного доступа является обязательным этапом написания программного продукта. Это позволяет защитить программный продукт от лишних и ненужных действий по отношению к программе, несанкционированного доступа к данным для их просмотра и копирования, охраны коммерческой и частной тайны контрагентов и самой организации.

Защита программы заключается, в первую очередь, в защите доступа к базе данных и осуществляется при необходимости соответствующей настройкой самого файла базы данных, Access позволяет защитить базу на уровне файла, задав пароль. С целью ограничения несанкционированного доступа при запуске программы производится запрос имени пользователя и его пароля. При некорректном вводе логина и пароля, система выдает сообщения об ошибке и прекращает работу с файлом

## 3.4 Разработка справочной системы

Справочная система - важная составная часть любой более-менее сложной (и даже простой) программы. Существуют разные форматы справочных систем. В проекте использована справочную систему в формате WinHelp. Из достоинств WinHelp можно назвать ее надежность и меньшие, чем у HTML-help, требования к ресурсам. Существуют различные программы для создания справочных систем названных типов.

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

# 4Энерго- и ресурсосбережение

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

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

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

(1)

где Тпк – время работы компьютера, ч;

Тпк =240,2 ч;

tрп – суммарное время регламентированных перерывов, в течение

рабочего дня, ч;

tрп =50 мин=5/6 часа.

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

(2)

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

Экономия электроэнергии рассчитывается по формуле

(3)

где Wпк – потребляемая мощность ПК, кВт;

Wсп – потребляемая мощность компьютера в ждущем режиме, кВт;

Wпк = 0,40 кВт;

Wсп = 0,16 кВт;

Сэ – стоимость 1 кВт электроэнергии, руб.

Таким образом сумма сэкономленной электроэнергии за время разработки программного обеспечения составила 2,07 рубля.

# Заключение

Результатом выполнения данного курсового проекта является разработанное программное средство для автоматизации учета контрагентов, их контактной информации, такой, как счета в банках, сотрудники и их координаты, договорах заключенных с контрагентами на поставку или закупку товаров, позволяющее хранить данные в базе данных и осуществлять управление, поиск информации и печать отчетов.

Программа выполнена в соответствии с техническим заданием, в полной мере реализует требуемые функции и предоставляет необходимые возможности.

Достоинствами данной программы являются:

* удобный и интуитивно-понятный интерфейс;
* легкость и простота в освоении и использовании;
* компактность и переносимость;
* возможность редактирования, добавления, удаления, изменения данных.

Разработанная программа нетребовательна к ресурсам компьютера, проста и удобна в использовании.

# Список используемых источников

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