МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БАРАНОВИЧСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет \_\_\_\_\_\_инженерный \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра информационные системы и технологии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность 1-40 01 02 «Информационные системы и технологии»

код наименование

Направление специальности 1-40 01 02-01\_«Информационные системы и технологии (в проектировании и производстве)»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

код наименование

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой

\_\_\_\_\_\_\_ ­­­­­\_\_А.В. Шах\_\_

(подпись) (инициалы и фамилия)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_20 \_\_ г.

**ДИПЛОМНЫЙ ПРОЕКТ**

Тема: Разработать приложение для учета и управления активов в ЦССО

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Руководитель | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  |  |  |  |
|  | (подпись) |  |  | (Ф. И. О.) |  |
| Консультант по охране труда | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  |  |  |  |
|  | (подпись) |  |  | (Ф. И. О.) |  |
| Консультант по экономической части | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  |  |  |  |
|  | (подпись) |  |  | (Ф. И. О.) |  |
|  |  |  |  |  |  |
| Нормоконтролер | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  |  |  |  |
|  | (подпись) |  |  | (Ф. И. О.) |  |
| Студент | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  |  |  |  |
|  | (подпись) |  |  | (Ф. И. О.) |  |

Барановичи 2018

АВТОМАТИЗИРОВАННАЯ СИСТЕМА, РУКОВОДИТЕЛЬ, ПРАКТИКАНТ, UML, ЯЗЫК C#,БАЗА ДАННЫХ, SQL ЗАПРОСЫ, СУБД MICROSOFTSQLSERVER, ADO.NET, AES, ПЛАТФОРМА .NET

Объектом разработки является процесс автоматизация учета и управления активами ЦССО.

Предметом исследования является программный продукт, обеспечивающий учет и управление активов предприятия.

Цель дипломного проекта: разработка приложения для учета и управления активами ЦССО.

В процессе проектирования исследована структура и деятельность предприятия, изучен и проанализирован процесс прохождения практики студентами в компании, определены функциональные требования к программному средству, разработаны логическая и физическая модели данных, разработано приложение, которое работает с реляционной БД, проведено тестирование и определен экономический эффект от внедрения программного продукта.

Элементами практической значимости полученных результатов является автоматизация учета и управления активами.

Областью возможного практического применения являются Центр современных средств обучения, а также иные предприятия.

Технико–экономическая и социальная значимость: программный продукт автоматизирует учет, помогает эффективно хранить и систематизировать информацию, сокращает время на составление документов, а также упрощает учет движения активов на предприятии.

Студент–дипломник подтверждает, что приведенный в дипломном проекте расчетно-аналитический материал объективно отражает состояние исследуемого процесса (разрабатываемого объекта), все заимствованные из литературных и других источников теоретические и методологические положения и концепции сопровождаются ссылками на их авторов.

Дипломный проект: 83 с., 49 рис., 4табл., 26 источника, 2 прил.

**Ведомость объема дипломного проекта**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Формат | Обозначение | | Наименование | | | Количестволистов | | | Примечание | |
| А4 | - | | Заданиеподипломномупроектированию | | | 1 | | |  | |
|  |  | |  | | |  | | |  | |
| А4 | ДП 2ИСТ31–ДО -2017 ПЗ | | Пояснительная записка | | | 83 | | |  | |
|  |  | |  | | |  | | |  | |
| А4  А4 | ДП 2ИСТ31–01–ДО -2017  ДП 2ИСТ31–02–ДО -2017 | | Приложение А – Код функциональных частей программы  Приложение Б – Презентация проекта | | | 21  21 | | |  | |
|  |  | |  | | |  | | |  | |
|  |  | |  | | |  | | |  | |
|  |  | |  | | |  | | |  | |
|  |  | |  | | |  | | |  | |
|  |  | |  | | |  | | |  | |
|  |  | |  | | |  | | |  | |
|  |  | |  | | |  | | |  | |
|  |  | |  | | |  | | |  | |
|  |  | |  | | |  | | |  | |
|  |  | |  | | |  | | |  | |
|  |  |  |  |  | ДП 2ИСТ31–ДО -2017 | | | | | |
|  |  |  |  |  |
|  |  | Ф.И.О | Подпись | Дата |
| Дипломник | | . |  |  | Ведомость объема дипломного проекта | | Стадия | Лист | | Листов |
| Руководитель ДП | |  |  |  | ТП | 1 | | 127 |
| Нормоконтролёр | |  |  |  | БарГУ, ИСТ,  г. Барановичи | | | |
| Зав. кафедрой | |  |  |  |

ОГЛАВЛЕНИЕ

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ 5](#_Toc483954193)

[ВВЕДЕНИЕ 6](#_Toc483954194)

[1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ 9](#_Toc483954195)

[1.1Анализ предметной области 9](#_Toc483954196)

[1.2Постановка задачи 10](#_Toc483954197)

[1.3Алгоритм шифрования AES 10](#_Toc483954198)

[1.4Краткая характеристика Microsoft Visual Studio 2012 12](#_Toc483954199)

[1.5Краткая характеристика Microsoft SQL Server 2012 14](#_Toc483954200)

[1.6Краткая характеристика языка программирования C# 15](#_Toc483954201)

[1.7Краткая характеристика платформы .NETFramework 16](#_Toc483954202)

[1.8Краткая характеристика технологии ADO.NET 18](#_Toc483954203)

[2 ПРАКТИЧЕСКИЙ РАЗДЕЛ 20](#_Toc483954204)

[2.1Определение объектов БД, концептуальное, логическое и физическое проектирование БД 20](#_Toc483954205)

[2.2Построение диаграмм в среде RationalRose 23](#_Toc483954206)

[2.2.1 Диаграмма вариантов использования UseCase 24](#_Toc483954207)

[2.2.2 Диаграмма последовательности Sequence 25](#_Toc483954208)

[2.2.3 Диаграмма деятельности Activity 27](#_Toc483954209)

[2.3Диаграмма классов 28](#_Toc483954210)

[2.4 Тестирование программного продукта 32](#_Toc483954211)

[2.5 Руководство пользователя 37](#_Toc483954212)

[2.5.1 Установка программного продукта 37](#_Toc483954213)

[2.5.2 Работа с программой 40](#_Toc483954214)

[3 ЭКОНОМИЧЕСКИЙ РАЗДЕЛ 59](#_Toc483954215)

[3.1 Определение трудоёмкости разработки программного продукта 59](#_Toc483954216)

[3.2 Определение себестоимости создания программного продукта 60](#_Toc483954217)

[3.2.1 Определение затрат на оплату труда разработчика 60](#_Toc483954218)

[3.2.2 Определение стоимости машино-часа работы ЭВМ (для разработанного программного продукта) 61](#_Toc483954219)

[3.2.3 Определение затрат на отладку программного продукта 64](#_Toc483954220)

[3.3 Определение отпускной цены программного продукта 64](#_Toc483954221)

[3.4 Определение ожидаемого прироста прибыли в результате внедрения программного продукта 66](#_Toc483954222)

[3.4.1 Определение годовых эксплуатационных расходов при ручном решении задачи…................ 66](#_Toc483954223)

[3.4.2 Определение годовых текущих затрат, связанных с эксплуатацией программного продукта 66](#_Toc483954224)

[3.4.3 Определение ожидаемого прироста прибыли в результате внедрения программного продукта 68](#_Toc483954225)

[3.5 Расчёт показателей эффективности использования программного продукта 69](#_Toc483954226)

[4 ОХРАНА ТРУДА 71](#_Toc483954227)

[4.1 Необходимость охраны труда 71](#_Toc483954228)

[4.2 Анализ состояния охраны труда 72](#_Toc483954229)

[4.2.1 Организация работы по охране труда 72](#_Toc483954230)

[4.2.2 Техника безопасности 73](#_Toc483954231)

[4.2.3 Производственная санитария 75](#_Toc483954232)

[4.2.4 Пожарная безопасность 75](#_Toc483954233)

[4.3 Мероприятия по улучшению условий и безопасности труда 76](#_Toc483954234)

[4.4 Разработка инструкции по охране труда для операторов ПЭВМ 77](#_Toc483954235)

[4.4.1 Общие требования техники безопасности 77](#_Toc483954236)

[4.4.2 Требования техники безопасности перед началом работы 78](#_Toc483954237)

[4.4.3 Требования техники безопасности при выполнении работы 78](#_Toc483954238)

[4.4.4 Требования безопасности по окончании работы 80](#_Toc483954239)

[ЗАКЛЮЧЕНИЕ 82](#_Toc483954240)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 84](#_Toc483954241)

[ПРИЛОЖЕНИЯ 87](#_Toc483954242)

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ADO – ActiveX Data Objects;

SQL – Structured Query Language;

UML – Unified Modeling Language;

AES – Advanced Encryption Standard;

ПО – Программное обеспечение;

ПП – Программный продукт;

AC – Автоматизированная система;

СУБД – Система управления базами данных;

БД – База данных;

ПЭВМ – Персональная электронная вычислительная машина;

ПК – Персональный компьютер;

ОС – Операционная система;

ОТ – Охрана труда;

ЦССО – Центр современных средств обучения.

ВВЕДЕНИЕ

В настоящем времени трудно назвать те области человеческой деятельности, успехи в которых не были бы связаны с использованием разнообразных информационных систем и компьютера. Сфера применения компьютера постоянно расширяется, существенно влияя на развитие производительных сил нашего общества. Непрерывно изменяются технико-экономические характеристики компьютера в сторону увеличения производительности и уменьшения стоимости вычислительных мощностей.

Передача информации о положении и деятельности предприятия, движении продукции, материальных и нематериальных активов и взаимный обмен информацией между всеми взаимными подразделениями осуществляются на базе современной электронно-вычислительной техники и других технических средствах связи.

В наше время информационные технологии являются основными в управлении организации всех типов, действующих во всех сферах жизнедеятельности человека. Основными функциями информационных технологий управления предприятия являются сбор, поиск, хранение, систематизация и обработка необходимых данных для всех сфер общественной жизни, выработка новой информации, решение тех или иных оптимизационных задач.

**Основными задачами центра являются:**

* администрирование Web-сайта отдела по образованию
* администрирование образовательных интернет ресурсов учреждений образования;
* создание банка компьютерных программ, аудио-видео материала, электронных документов и учебников;
* организация обучения информационным и компьютерным технологиям;
* техническое обслуживание компьютерной и организационной техники и ТСО, находящейся в учреждениях образования;
* списание технически устаревших средств обучения, подготовка извлеченного лома, содержащего драгметаллы для сдачи;
* техническое сопровождение мероприятий отела по образованию, в том числе разработка презентаций, тиражирование учебно-методических пособий и средств обучения;
* видеосъемка районных мероприятий, открытых уроков;

Целью дипломного проектирования является разработка алгоритмов и программного комплекса для учета и управления активами ЦССО.

В рамках сформулированной цели необходимо было решить следующие задачи:

* изучить особенности учета активов в ЦССО;
* выбрать пакет прикладных программ, необходимый для реализации проекта;
* построить и описать концептуальную, логическую и физическую модели данных;
* разработать реляционную базу данных;
* разработать алгоритмы и ПП;
* провести тестирование ПП;
* определить экономический эффект от внедрения разработанного ПП.

Объект исследования – автоматизированная система учета и управления активами ЦССО.

Предмет исследования – основные теоретические аспекты, алгоритмические механизмы и программный продукт, обеспечивающий учет и управление активами ЦССО.

При выполнении дипломного проекта использовались следующие методы исследований: аналитический метод, метод синтеза и анализа, а также метод монографических публикаций и статей.

Программный продукт взаимодействует с базой данных, так как на сегодняшний день их использование становится неотъемлемой частью функционирования любых организаций и предприятий.

База данных помогает систематизировать, упарядочить и хранить информацию из определенной предметной области, облегчает доступ к данным, поиск и предоставление необходимых сведений, обеспечивает их целостность и безопасность [1].

Для реализации приложения была выбрана мощная интегрированная среда разработки профессионального уровня – MS Visual Studio 2017. При написании программы был использован объектно-ориентированный язык высокого уровня C# и платформа .Net Framework 4.6. Для проектирования и реализации работы с базой данных была выбрана одна из самых популярных систем управления базами данных – Microsoft SQL Server 2016. Построение диаграмм проекта проходило в программе Rational Rose 7.0, которая представляет собой CASE средство проектирования информационных систем и программного обеспечения.

1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ

* 1. Анализ предметной области

Начальным этапом проектирования базы данных является определение ее предметной области.

Предметная область – это часть реального мира, данные о которой необходимо отразить в базе данных [1, c. 13].

В данном дипломном проекте рассматриваются задачи, выполняемые компанией JazzTeam,при организации и проведении практики у студентов.

База данных обязательно должна содержать информацию о практике, которая будет проводиться в компании, об организации,отправляющей студента на практику, о практиканте, который будет ее проходить, о руководителе, который будет заниматься обучением студента и контролем прохождения практики.

Помимо этого в базе данных необходимо хранить разнообразную информацию о практических заданиях и их выполнении, дневник прохождения практики, данные авторизации, список должностей руководителей и вакансий, по которым обучаются практиканты.

Разрабатываемый программный продукт должен осуществлять работу с базой данных и позволять получить студенту и его руководителю все возможные сведения о прохождении практики, а также производить добавление, редактирование, удаление и поиск необходимой информации, осуществлять планирование выполняемых задач и взаимодействие практиканта со своим руководителем.

Пользователи, работающие с приложением, делятся на два типа:

* «Руководитель»
* «Практикант»

В зависимости от типа, пользователю предоставляются те или иные права доступа и рабочий интерфейс программы. Данные меры представляют собой простой и эффективный способ защиты данных от несанкционированных действий пользователя.

Для защиты данных от взлома и хищения информации используется шифрование логина и пароля методом AESшифровки.

1.2 Постановка задачи

Задачей дипломного проекта является разработка автоматизированной системы контроля прохождения практики студентами в компании JazzTeam.

Требования, предъявляемы к разрабатываемому программному обеспечению:

* удобный пользовательский интерфейс;
* осуществление работы с программой только после прохождения авторизации;
* ограничение функциональных возможностей программного продукта в зависимости от типа пользователя;
* подключение к базе данных в сетевом режиме;
* работа с операционной системой Windows 7 и выше;
* добавление, редактирование и удаление записей;
* поиск и сортировка данных;
* обеспечение защиты и целостности данных;
* автоматическое удаление устаревших записей в базе данных;
* экспорт данныхв MicrosoftWordи MicrosoftExcel.

1.3 Алгоритм шифрования AES

Для защиты данных от взлома и хищения информациив программе используется шифрование логина и пароля методом AESшифровки.

Advanced Encryption Standard — симметричный алгоритм блочного шифрования, принятый правительством США в качестве стандарта в результате конкурса, проведенного между технологическими институтами. Он заменил устаревший Data Encryption Standard, который больше не соответствовал требованиям сетевой безопасности, усложнившимся в XXI веке. Этот алгоритм, кроме аббревиатуры AES, иногда называют еще Rijndael — это анаграмма из частей имен бельгийских программистов Joan Daemen и Vinent Rijmen, которые его разработали.

Алгоритм AES был одобрен Агентством национальной безопасности США как пригодный для шифрования особо секретной информации. Однако, правительство постановило, что AES должен периодически подвергаться проверкам и улучшениям, чтобы надежно хранить зашифрованные данные.

Суть AES в том, что любая «лобовая атака» на защищенные данные — то есть подбор всех возможных паролей – в перспективе очень сильно растягивается. Если представить, что взломщик располагает огромными ресурсами, то есть целой коллекцией суперкомпьютеров, то при усердном старании доступ к зашифрованным данным он мог бы получить через десятки лет. Если же в его распоряжении ничего этого нет, то взлом AES займет астрономически долгое время [2].

Преимущества использование AESшифрования:

* выполняется быстрее, чем обычный блочный алгоритм шифрования. Выполнена оптимизация размера таблицы и скорости выполнения;
* алгоритм шифрования не использует арифметические операции, поэтому тип архитектуры процессора не имеет значения;
* алгоритм шифрования полностью «само определяемый». Он не использует других криптографических компонентов, взятых из хорошо известных алгоритмов, битов, полученных из специальных таблиц и тому подобных уловок;
* алгоритм не основывает свою безопасность или часть ее на неясностях или плохо понимаемых итерациях арифметических операций;
* определение алгоритма позволяет специфицировать варианты длины блока и длины ключа в диапазоне от 128 до 256 бит с шагов в 32 бита;
* хотя число раундов Rijndael для AES фиксировано, в случае возникновения проблем с безопасностью алгоритм можно модифицировать, сделав число раундов параметром алгоритма[3];
* не подвержен многим видам криптоаналитических атак, таких как дифференциальный и линейный криптоанализ, Square-атака, метод интерполяции и др.

1.4 Краткая характеристика Microsoft Visual Studio 2017

Для создания приложения, работающего с базой данных, была выбрана интегрированная среда разработки профессионального уровняMicrosoft Visual Studio 2012.

Она позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых [Microsoft Windows](http://ru.bmstu.wiki/Microsoft_Windows), Windows Mobile, Windows CE, NET Framework, Xbox, Windows Phone.NET Compact Framework и [Microsoft Silverlight](http://ru.bmstu.wiki/Microsoft_Silverlight) [4].

MS Visual Studio 2012 меняет отношение к процессу разработки, делая его увлекательным и хорошо организованным. Продукт отличается повышенной скоростью загрузки рабочей среды и открывает разработчикам доступ к конкретным проектам буквально в считанные секунды. Кроме того, все длительные процессы выполняются в фоновом режиме, что не замедляет скорость работы среды и не отвлекает разработчика от основных задач.

Также рабочая среда Visual Studio 2012 имеет новый контекстно-зависимый интерфейс. Главная его особенность заключается в том, что он предлагает разработчику только те функции и инструменты, которые ему нужны на данном этапе работы. Таким образом, панель инструментов не содержит ничего лишнего и не затрудняет поиск нужных функций.

Функциональная сторона: среда для разработки Visual Studio 2012 создана так, чтобы помогать разработчику и подсказывать ему наилучшие решения. Для этого в продукт были включены специальные технологии, которые упрощают и ускоряют работу. Например, функция Code Clone ищетодинаковые по функционалу коды и объединяет их в соответствующие модули для того, чтобы разработчик впоследствии мог воспользоваться уже готовым кодом при создании нового приложения [5].

Функциональные возможности Visual Studio 2012[6]:

* возможность работы с базами данных: включает инструменты развертывания и управления изменениями, которые дают возможность снабдить постоянную синхронизацию базы данных и приложения;
* наличие средств тестирования: включает стандартные инструменты тестирования, которые дают возможность обеспечить высокое качество программного кода;
* отладка и диагностика: средства оценки кода позволяют избежать обычных ошибок программирования до старта приложений в производство. Настроить оценку кода элементарно просто благодаря наборам правил, которые предназначены для конкретных сценариев или же важных участков кода. Кроме того, метрики кода обеспечивают простоту и удобство обслуживания создаваемого программного кода;
* наличие интегрированной среды разработки: предоставляет полный набор инструментов управления цифровым рабочим местом. Система включает такие настраиваемые функции, как поддержка нескольких мониторов, которая позволяет более удобно осуществить работу;
* поддержка различных платформ проектирования: охватывает такие платформы как Windows, Windows Server, Office, SharePoint, веб-среду, облачную среду и многие другие, благодаря единой интегрированной среде проектирования;
* управление жизненным циклом приложений (ALM): создание качественных приложений подразумевает четкий и бесперебойный процесс, благоприятный для всех участников рабочей группы. Вмонтированные инструменты управления жизненным циклом приложений в Visual Studio 2012 обеспечивают эффективное взаимодействие и совместную работу на всех этапах, а также визуальное представление состояния проекта, делая проще экономичное создание высококачественных решений.

1.5 Краткая характеристикаMicrosoftSQL Server2012

Для проектирования, разработки и создания базы данных, с которой работает программа, была использована мощная система управления реляционными базами данных MicrosoftSQLServer 2012.

Вее состав входит компонент DatabaseEngine, службы анализа AnalysisServices, службы отчетов ReportingServices, интеграционные службы IntegrationServices и расширение SQLXML.

SQLServer является наилучшим выбором для широкого диапазона конечных пользователей и программистов баз данных по двум причинам:

* SQL Server – несомненно, наилучшая система для операционных систем Windows, вследствие ее тесной интеграции с ними (а также вследствие низкой стоимости). Благодаря огромному и все возрастающему количеству установленных систем Windows, SQL Server является широко применяемой системой управления базами данных;
* будучи составляющей системы реляционной базы данных, компонент Database Engine является самой легкой в использовании системой баз данных. Кроме хорошо знакомого пользовательского интерфейса, разработчики предоставляют несколько разных инструментов для создания объектов баз данных, настраивания приложений баз данных и управления задачами системного администрирования.

В целом, SQL Server является больше, чем просто системой управления реляционными базами данных. Это платформа, которая не только позволяет управлять структурированными, полуструктурированными и неструктурированными данными, но также предоставляет комплексное, интегрированное системное программное обеспечение и программное обеспечение для аналитических исследований, которые позволяют организациям надежно управлять критически важными данными [8, c. 5].

Кроме того, она включает в себя мощный графический управленческий потенциал и интерфейс доступа для основных платформ разработки, в том числе .NET.

1.6 Краткая характеристикаязыка программирования C#

Для написания программного продукта был выбран объектно-ориентированный язык программирования C#.

C# – это язык программирования, предназначенный для разработки самых разнообразных приложений, предназначенных для выполнения в среде .NET Framework. Язык C# прост, типобезопасен и объектно-ориентирован. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую языкам C [9].

На сегодняшний момент язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в IT-отрасли. В настоящий момент на нем пишутся самые различные приложения: от небольших десктопных программок до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей.

По сравнению с другими языками C# достаточно молодой, но в то же время он уже прошел большой путь. Первая версия языка вышла вместе с релизом Microsoft Visual Studio .NET в феврале 2002 года.

C# является объектно-ориентированным языком программирования, а значит, поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений[10].

1.7 Краткая характеристикаплатформы .NETFramework

Разработка программы велась на платформе .NetFramework.

Платформа .NETFramework– это технология, которая поддерживает создание и выполнение нового поколения приложений и веб-служб XML. При разработке платформы .NETFramework учитывались следующие цели:

* обеспечение согласованной объектно-ориентированной среды программирования для локального сохранения и выполнения объектного кода, для локального выполнения кода, распределенного в Интернете, либо для удаленного выполнения;
* обеспечение среды выполнения кода, минимизирующей конфликты при развертывании программного обеспечения и управлении версиями программного продукта;
* обеспечение среды выполнения кода, гарантирующей безопасное выполнение кода, включая код, созданный неизвестным или не полностью доверенным сторонним изготовителем;
* обеспечение среды выполнения кода, исключающей проблемы с производительностью сред выполнения сценариев или интерпретируемого кода;
* обеспечение единых принципов работы разработчиков для разных типов приложений, таких как приложения Windows и веб-приложения;
* разработка взаимодействия на основе промышленных стандартов, которое обеспечит интеграцию кода платформы .NETFramework с любым другим кодом[11].

Когда говорят C#, нередко имеют в виду технологии платформы .NET. И, наоборот, когда говорят .NET, нередко имеют в виду C#. Однако, хотя эти понятия связаны, отождествлять их неверно. Язык C# был создан специально для работы с фреймворком .NET, однако само понятие .NET несколько шире.

.NET Framework представляет собой мощную платформу для создания приложений. Можно выделить следующие ее основные черты:

1. Поддержка нескольких языков. Основой платформы является общеязыковая среда исполнения Common Language Runtime (CLR), благодаря чему .NET поддерживает несколько языков: наряду с C# это также VB.NET, C++, F#, а также различные диалекты других языков, привязанные к .NET, например, Delphi.NET. При компиляции код на любом из этих языков компилируется в сборку на общем языке CIL (Common Intermediate Language) – своего рода ассемблер платформы .NET. Поэтому есть возможность сделать отдельные модули одного приложения на разных языках;
2. Кроссплатформенность. Платформа .NET является переносимой (с некоторыми ограничениями). Например, последняя версия платформы на данный момент .NET Framework поддерживается на большинстве современных ОС Windows (Windows 10/8.1/8/7/Vista). А благодаря проекту Mono можно создавать приложения, которые будут работать и на других ОС семейства Linux, в том числе на мобильных платформах Android и iOS;
3. Мощная библиотека классов. .NET представляет единую для всех поддерживаемых языков библиотеку классов. И какое бы приложение мы не собирались писать на C# - текстовый редактор, чат или сложный веб-сайт – так или иначе мы задействуем библиотеку классов .NET;
4. Разнообразие технологий. Общеязыковая среда исполнения CLR и базовая библиотека классов являются основой для целого стека технологий, которые разработчики могут задействовать при построении тех или иных приложений. Например, для работы с базами данных в этом стеке технологий предназначена технология ADO.NET. Для построения графических приложений с богатым насыщенным интерфейсом – технология WPF. Для создания веб-сайтов – ASP.NET и т.д.

Также еще следует отметить такую особенность языка C# и фреймворка .NET, как автоматическая сборка мусора. А это значит, что в большинстве случаев не придется, в отличие от С++, заботиться об освобождении памяти. Вышеупомянутая общеязыковая среда CLR сама вызовет сборщик мусора и очистит память [10].

1.8 Краткая характеристикатехнологии ADO.NET

Чтобы осуществлять связь между базой данных и приложением на C# необходим посредник. И именно таким посредником является технология ADO.NET.

ADO.NET предоставляет собой технологию работы с данными, которая основана на платформе .NET Framework. Эта технология представляет собой набор классов, через которые можно отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Причем важно отметить, что систем управления баз данных может быть множество. В своей сущности они могут различаться. Разные системы баз данных могут иметь разные типы данных. Также могут различаться какие-то другие моменты. Однако функционал ADO.NET построен таким образом, чтобы предоставить разработчикам унифицированный интерфейс для работы с самыми различными СУБД.

Основу интерфейса взаимодействия с базами данных в ADO.NET представляет ограниченный круг объектов:Connection,Command, DataReader,DataSet и DataAdapter. С помощью объектаConnectionпроисходитустановкаподключениякисточникуданных. ОбъектCommandпозволяетвыполнятьоперациисданнымииз БД. ОбъектDataReaderсчитываетполученные в результате запроса данные. ОбъектDataSetпредназначен для хранения данных из БД и позволяет работать с ними независимоот БД. Иобъект DataAdapter является посредником между DataSet и источником данных. Главным образом, через эти объекты осуществляется работа с базой данных.

Однако чтобы использовать один и тот же набор объектов для разных источников данных, необходим соответствующий провайдер данных. Собственно черезпровайдер данных в ADO.NET и осуществляется взаимодействие с базой данных. Причем для каждого источника данных в ADO.NET может быть свой провайдер, который собственно и определяет конкретную реализацию вышеуказанных классов.

По умолчанию в ADO.NET имеются следующие встроенные провайдеры:

* провайдер для MS SQL Server;
* провайдер для OLE DB (Предоставляет доступ к некоторым старым версиям MS SQL Server, а также к БД Access, DB2, MySQL и Oracle);
* провайдер для ODBC (Провайдер для тех источников данных, для которых нет своих провайдеров);
* провайдер для Oracle;
* провайдер EntityClient.Провайдер данных для технологии ORM Entity Framework;
* провайдер для сервера SQL Server Compact 4.0.

Кроме этих провайдеров, которые являются встроенными, существует также множество других, предназначенных для различных баз данных.

Функционально классы ADO.NET можно разбить на два уровня: подключенный и отключенный. Каждый провайдер данных .NET реализует свои версии объектов Connection, Command, DataReader, DataAdapter и ряда других, которые составляют подключенный уровень.

Другие классы, такие как DataSet, DataTable, DataRow, DataColumn и ряд других составляют отключенный уровень, так как после извлечения данных в DataSet можно продолжать работу с этими данными независимо от того, установлено ли подключение или нет. То есть после получения данных из БД приложение может быть отключено от источника данных[12].

2 ПРАКТИЧЕСКИЙ РАЗДЕЛ

2.1 Определение объектов БД, концептуальное, логическое и физическое проектирование БД

Концептуальная модель – это отражение предметной области, для которой разрабатывается база данных[13].

Средством моделирования предметной области на этапе концептуального проектирования является модель "сущность-связь". Часто ее называют ER-моделью (Entity-сущность, Relation-связь). В ней моделирование структуры данных предметной области базируется на использовании графических средств – ER-диаграмм.

Основные понятия ER-диаграммы – сущность, атрибут, связь.

Сущность – это некоторый объект реального мира, который может существовать независимо. Сущность имеет экземпляры, отличающиеся друг от друга значениями атрибутов и допускающие однозначную идентификацию.

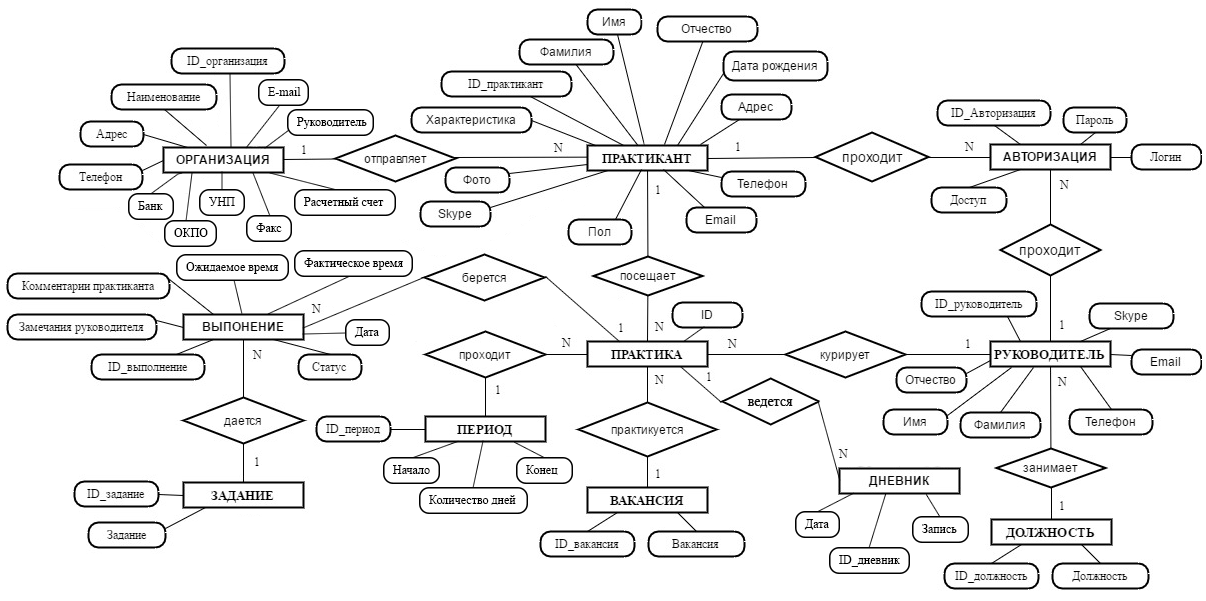
Атрибут – это свойство сущности. Атрибут, который уникальным образом идентифицирует экземпляры сущности, называется ключом. Может быть составной ключ, представляющий комбинацию нескольких атрибутов.

Связь – представляет взаимодействие между сущностями. Она характеризуется мощностью, которая показывает, сколько сущностей участвует в связи [14].

В соответствии с предметной областью система строится с учётом следующих особенностей:

* организация может отправить на практику несколько человек;
* практикант ведет свой дневник, делая множество записей;
* за каждую практику ведется отдельный дневник;
* на каждый день (на каждую дату) может быть дано несколько заданий для выполнения;
* за определенный период времени может проводиться несколько разных практик;
* на одну и ту же вакансию может прийти несколько практикантов;
* по каждой должности в компании может работать несколько человек;
* практикант может брать на выполнение несколько разных заданий;
* одно задание может быть выдано нескольким практикантам;
* руководитель может обучать одновременно несколько практикантов;
* практикант может пройти в компании несколько разных практик.

На рисунке 2.1 представлена концептуальная диаграмма данных.

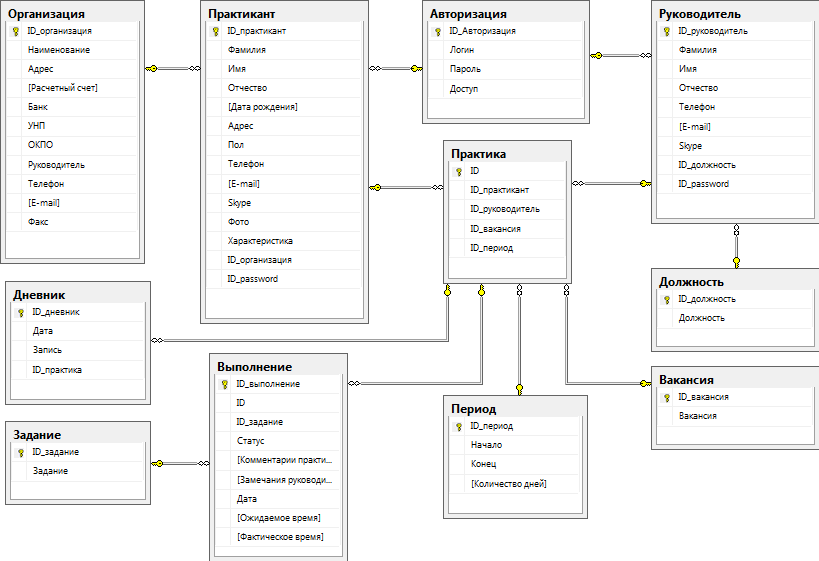


**Рисунок 2.1 – Концептуальная диаграмма данных**

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Она является начальным прототипом будущей базы данных[15].

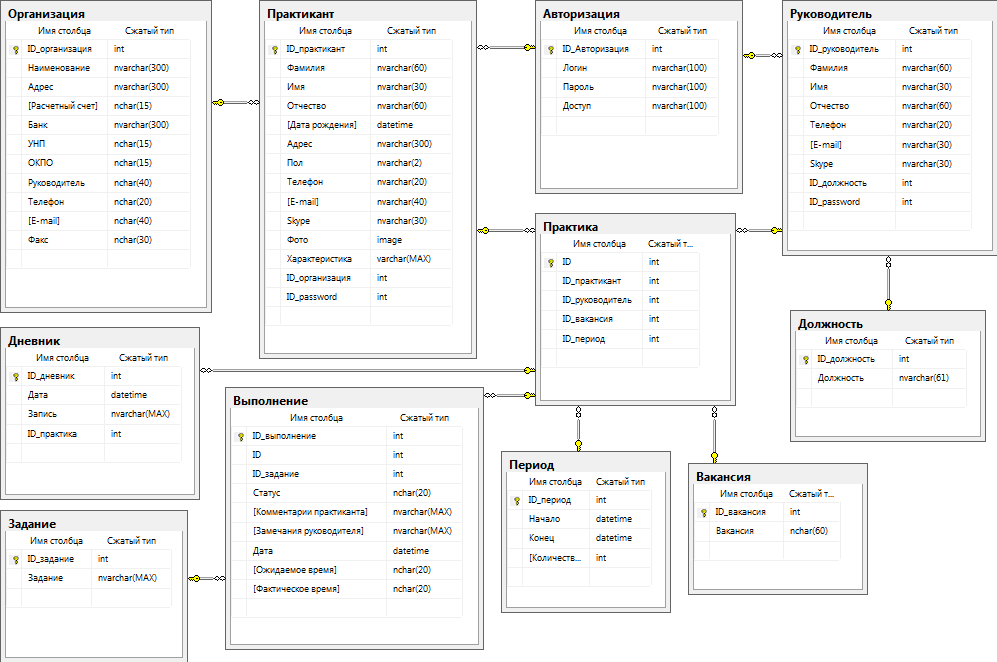
Для построения логической модели данных выделяем сущности и атрибуты данной предметной области.Получим следующие сущности: «Организация», «Практикант», «Руководитель», «Авторизация», «Практика», «Дневник», «Задание», «Выполнение», «Период», «Вакансия», «Должность».

Логическая диаграмма данных приедставлена на рисунке 2.2.



**Рисунок 2.2 – Логическая диаграмма данных**

Физическая модель баз данных определяет способы размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне [16].Физическая модель представлена на рисунке 2.3.



**Рисунок 2.3 – Физическая диаграмма данных**

2.2 Построение диаграмм в среде RationalRose

Rational Rose –мощное CASE-средство визуального моделирования объектно-ориентированных информационных систем компании Rational Software Corp. Работа продукта основана на универсальном языке моделирования UML (Universal Modeling Language). Благодаря уникальному языку моделирования Rational Rose способен решать практически любые задачи в проектировании информационных систем: от анализа бизнес процессов до кодогенерации на определенном языке программирования.

Только Rational Rose позволяет разрабатывать как высокоуровневые, так и низкоуровневые модели, осуществляя тем самым либо абстрактное проектирование, либо логическое.

Одно из достоинств этого программного продукта – возможность использования диаграмм на языке UML. Можно сказать, что Rational Rose является графическим редактором UML диаграмм [17].

### 2.2.1 Диаграмма вариантов использованияUseCase

Диаграммы вариантов использования (use case diagrams) представляют собой графическое представление взаимодействия пользователя и компьютерной системы. Каждый вариант использования охватывает некоторую очевидную для пользователей функцию системы и решает некоторую дискретную задачу пользователя. Список всех вариантов использования фактически определяет функциональные требования к системе.

Действующее лицо (actor) – это роль, которую пользователь играет по отношению к системе.

Вариант использования (use case) представляет собой типичное взаимодействие пользователя и компьютерной системы и решает некую дискретную задачу пользователя. Каждый вариант использования – это потенциальное требование к системе [18].

В данномдипломномпроекте действующими лицами выступают руководительи практикант.

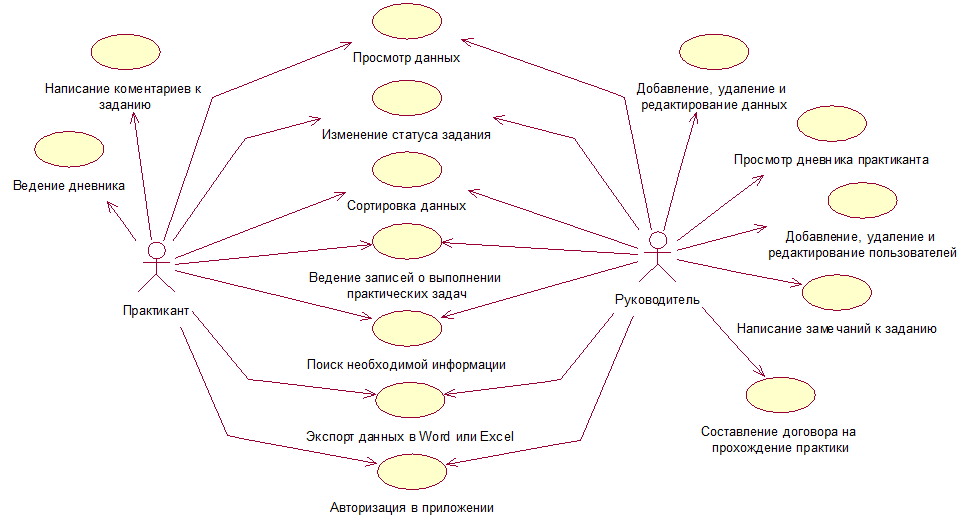
Руководителюприложение предоставляет следующие возможности:

* просмотр любых данных хранящихся в базе данных;
* изменения статуса задания;
* добавление, удаление и редактирование любых данных.
* сортировкаи поиск данных;
* экспорт данных в Wordи Excel;
* ведение записей о выполнении практических задач;
* добавление, удаление и редактирование пользователей;
* составление договора на прохождение практики;
* написание замечаний к заданию;
* редактирование списка практических задач.

Практиканту предоставляются следующие возможности работы с приложением:

* просмотр данных только связанных с ним;
* изменение статуса задания на «в очереди», «в работе», «на проверку»;
* ведение дневника;
* просмотр своей характеристики;
* сортировкаи поиск данных;
* ведение записей о выполнении практических задач;
* экспорт дневника, характеристики и списка выполняемых практических задач в Wordи Excel;
* написание комментариев к заданию.

UseCase диаграмма проекта представлена на рисунке 2.4.



**Рисунок 2.4 – UseCase диаграмма**

**2.2.2 Диаграмма последовательности Sequence**

Диаграмма последовательности является одной из разновидности диаграмм взаимодействия и предназначена для моделирования взаимодействия объектов системы во времени, а также обмена сообщениями между ними.

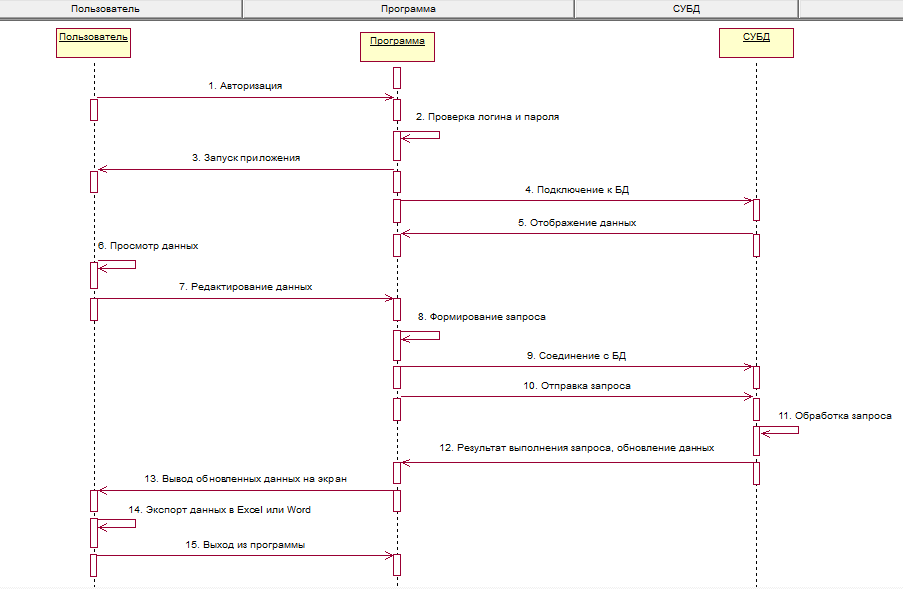
Одним из основных принципов объектно-ориентированного программирования является способ информационного обмена между элементами системы, выражающийся в отправке и получении сообщений друг от друга. Таким образом, основные понятия диаграммы последовательности связаны с понятием объект и сообщение.

На диаграмме последовательности объекты в основном представляю экземпляры класса или сущности, обладающие поведением.

Неотъемлемой частью объекта на диаграмме последовательности является линия жизни объекта. Линия жизни показывает время, в течение которого объект существует в системе.

Прием сообщения инициирует выполнение определенных действий, направленных на решение отдельной задачи тем объектом, которому это сообщение отправлено. Сообщение в большинстве случаев – это вызов методов отдельных объектов[19].

Описание взаимодействия объектов системы при помощи диаграммы Sequence представлено на рисунке 2.5.



**Рисунок 2.5 – Sequence диаграмма**

**2.2.3 Диаграмма деятельностиActivity**

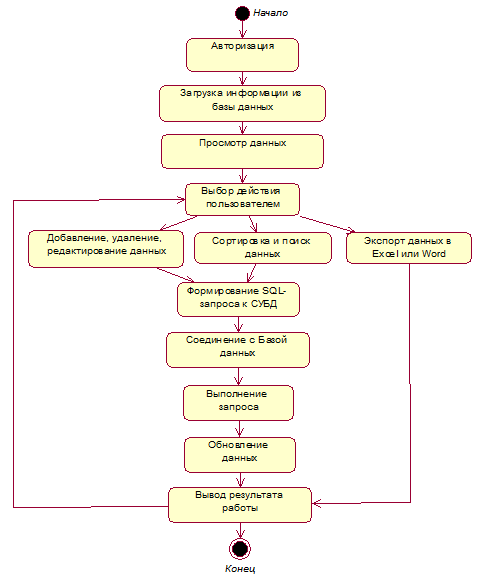
Диаграмма деятельности – это, по существу, блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой.

Диаграммы деятельности можно использовать для моделирования динамических аспектов поведения системы. Как правило, они применяются, чтобы промоделировать последовательные (а иногда и параллельные) шаги вычислительного процесса. С помощью диаграмм деятельности можно также моделировать жизнь объекта, когда он переходит из одного состояния в другое в разных точках потока управления.

Деятельность (Activity) – это некоторый относительно продолжительный этап выполнения в автомате. В конечном итоге деятельность сводится к некоторому действию (Action), которое составлено из атомарных вычислений, приводящих к изменению состояния системы или возврату значения.

Диаграммы деятельности важны не только для моделирования динамических аспектов поведения системы, но и для построения выполняемых систем посредством прямого и обратного проектирования [20].

Модель поведения системы при помощи диаграммы Activity представлена на рисунке 2.6.



**Рисунок 2.6 – Activityдиаграмма**

2.3 Диаграмма классов

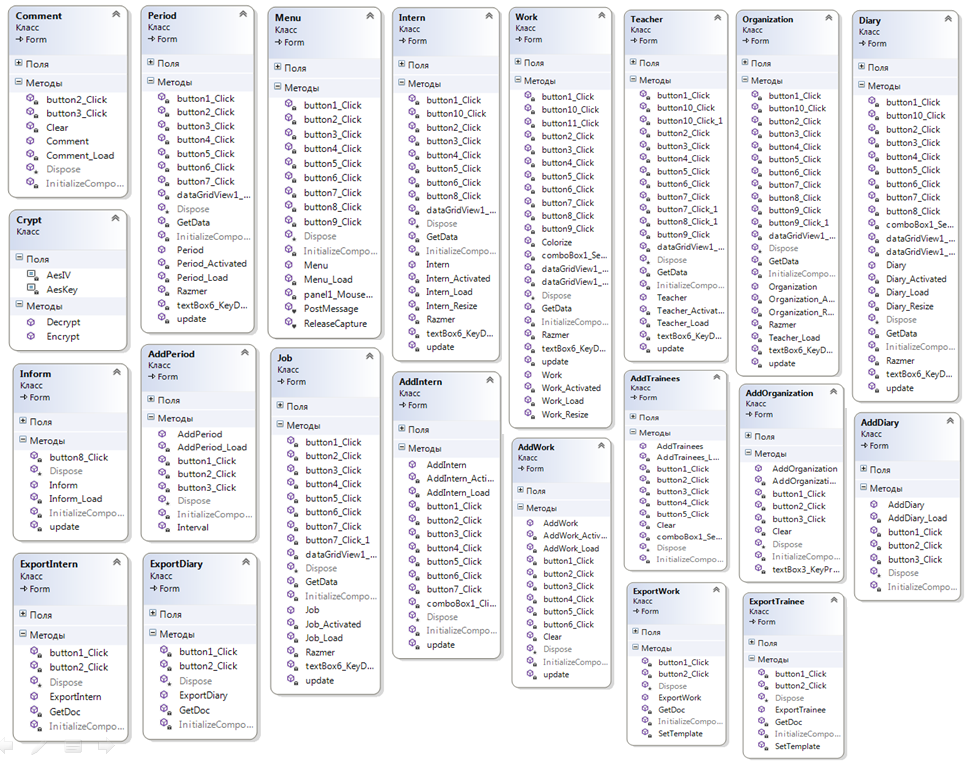
Схемы классов могут выводить всевозможные отношения классов, но в самом простом варианте этот тип схем изображает набор классов или объектов, а также статические отношения между ними[21].

Диаграммы классов при моделировании объектно-ориентированных систем встречаются чаще других. Диаграммы классов важны не только для визуализации, специфицирования и документирования структурных моделей, но также для прямого и обратного проектирования исполняемых систем[22].

При разработке приложения, было реализованы следующие классы:

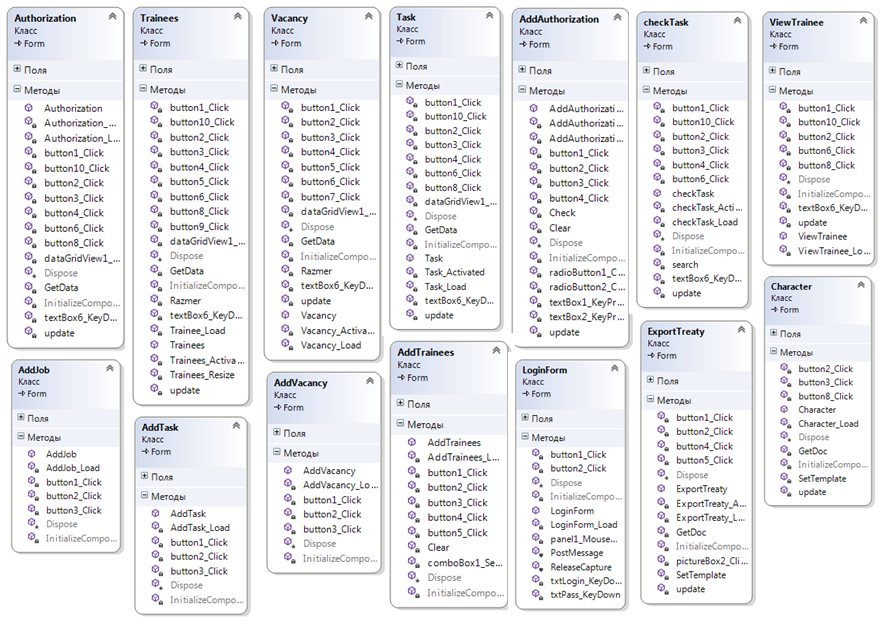
* Teacher – организовывает интерфейс работы с таблицей «Руководитель»;
* Trainees – организовывает интерфейс работы с таблицей «Практиканты»;
* Crypt – отвечает за шифровку логина и пароля в базе данных алгоритмом AES;
* Organization – организовывает интерфейс работы с таблицей «Организация»;
* Comment – отвечает за добавление, редактирование и удаление комментариев и замечаний к заданию;
* Intern – организовывает интерфейс работы с таблицей «Практика»;
* Task – отвечает за ведения списка практических задач;
* Work – организовывает интерфейс работы с таблицей «Выполнение»;
* Inform – выводит дополнительную информацию о практиканте;
* Diary – организовывает интерфейс работы с таблицей «Дневник»;
* Character – отвечает за добавление, редактирование и удаление характеристики практиканта;
* Vacancy – отвечает за ведение справочника вакансий;
* Menu – реализовывает главное меню программы;
* Job – отвечает за ведение справочника должностей;
* Period – организовывает интерфейс работы с таблицей «Период»;
* ViewTrainee – отвечает за вывод практикантов выбранной организации или руководителя;
* Authorization – организовывает интерфейс работы с таблицей «Авторизация»;
* CheckTask – реализует выбор практикантом задачи из списка задач;
* LoginForm – отвечает за реализацию процесса авторизации в программе;
* AddTeacher – выполняет операции по редактированию данных о руководителях практики;
* AddAuthorization – выполняет операции по редактированию данных необходимых для авторизации пользователя;
* AddTrainees – выполняет операции по редактированию данных о практикантах;
* AddVacancy – выполняет операции по редактированию данных справочника вакансий;
* AddIntern – выполняет операции по редактированию данных о практике;
* AddWork – выполняет операции по редактированию данных плана выполнения практических задач;
* AddJob – выполняет операции по редактированию данных справочника должностей;
* AddTask – выполняет операции по редактированию данных практических задач;
* AddOrganization – выполняет операции по редактированию данных об организациях;
* AddDiary – отвечает за редактирование данных дневника;
* AddPeriod – выполняет операции по редактированию данных о периодах проведения практик;
* ExportTreaty – отвечает за составление договора о прохождении практики в Word;
* ExportTrainee – отвечает за экспорт списка практикантов выбранной организации или руководителя в Word и Excel;
* ExportWork – отвечает за экспорт отчета о выполнении практических задач в Word и Excel;
* ExportIntern – отвечает за экспорт информации о практике в Word и Excel;
* ExportDiary – отвечает за экспорт дневника практиканта в Word и Excel;

Первая часть диаграммыклассовпредставленанарисунке2.7.



**Рисунок 2.7 – Диаграмма классов**

Вторая часть диаграммыклассовпредставленанарисунке2.8.



**Рисунок 2.8 – Диаграмма классов**

2.4**Тестирование программного продукта**

Тестирование программного обеспечения (ПО) – это процесс исследования ПО с целью выявления ошибок и определения соответствия между реальным и ожидаемым поведением ПО, осуществляемый на основе набора тестов, выбранных определённым образом. В более широком смысле, тестирование ПО – это техника контроля качества программного продукта, включающая в себя проектирование тестов, выполнение тестирования и анализ полученных результатов.

Очень часто современные программные продукты разрабатываются в сжатые сроки и при ограниченных бюджетах проектов. Программирование сегодня перешло из разряда искусства в разряд ремесел для многих миллионов специалистов. Но, к сожалению, в такой спешке разработчики зачастую игнорируют необходимость обеспечения защищённости своих продуктов, подвергая тем самым пользователей неоправданному риску. Контроль качества (тестирование) считается важным в процессе разработки ПО, потому что обеспечивает безопасность, надёжность, удобство создаваемого продукта. В настоящее время существует великое множество подходов и методик к решению задачи тестирования ПО, но эффективное тестирование сложных программных систем - процесс творческий, не сводящийся к следованию строгим и чётким правилам[23].

Журнал тестирования программного продукта представлен в таблице 2.1.

Таблица 2.1 - Журнал тестирования программного продукта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название  модуля/экрана | Описание тестового случая | Ожидаемые результаты | Тест пройден?  Да/Нет | Комментарии |
| Кнопка «Войти» | Одинарный клик левой кнопкой мыши | Соединение с базой данных и вход в приложение | Да | Работает корректно |
| Кнопка «Войти» | Одинарный клик левой кнопкой мыши при введенных неверных данных логина/пароля | Запуск сообщения «В доступе отказано!» | Да | Работает корректно |
|  | | | | |
| Продолжение таблицы 2.1 | | | | |
| Кнопка «Войти» | Одинарный клик левой кнопкой мыши при пустом поле «Сервер» | Запуск сообщения «Введите имя сервера!» | Да | Работает корректно |
| Кнопка «Отмена» | Одинарный клик левой кнопкой мыши | Закрытие формы | Да | Работает корректно |
| Кнопка «» | Одинарный клик левой кнопкой мыши | Закрытие приложения | Да | Работает корректно |
| Строка «Поиск» | Ввод данных с клавиатуры | Поиск в БД | Да | Работает корректно |
| Кнопка «Добавитьзапись» | Одинарный клик левой кнопкой мыши | Запуск формы на добавление данных в соответствующей активной таблице | Да | Работает корректно |
| Кнопка «Добавить» | Одинарный клик левой кнопкой мыши | Добавление новой записи в соответствующую активную таблицу | Да | Работает корректно |
| Кнопка «Добавить» | Одинарный клик левой кнопкой мыши при незаполненном обязательном поле | Запуск сообщение «Поле должно быть заполнено!» | Да | Работает корректно |
| Кнопка «Редактироватьзапись» | Одинарный клик левой кнопкой мыши | Запуск формы на редактирование данных в соответствующей активной таблице | Да | Работает корректно |
| Кнопка «Редактировать» | Одинарный клик левой кнопкой мыши | Редактирование записи в соответствующей активной таблице | Да | Работает корректно |
| Кнопка «Удалитьзапись» | Одинарный клик левой кнопкой мыши | Запуск диалогового окна «Удалить выбранную запись?» | Да | Работает корректно |
| Ответ «Да» на сообщение диалогового окна «Удалить выбранную запись?» | Одинарный клик левой кнопкой мыши | Удаление записи из соответствующей активной таблицы | Да | Работает корректно |
| Ответ «Нет» на сообщение диалогового окна «Удалить выбранную запись?» | Одинарный клик левой кнопкой мыши | Отмена удаления записи из соответствующей активной таблицы | Да | Работает корректно |
| Продолжение таблицы 2.1 | | | | |
| Кнопка «Поиск» | Одинарный клик левой кнопкой мыши | Показать/скрыть панель поиска | Да | Работает корректно |
| Кнопка «Начать поиск» | Одинарный клик левой кнопкой мыши | Поиск данных в соответствующей активной таблице | Да | Работает корректно |
| Кнопка «Обновить таблицу» | Одинарный клик левой кнопкой мыши | Возврат исходного состояния соответствующей активной таблицы после проведения поиска | Да | Работает корректно |
| Кнопка «Словарь «Должность» » | Одинарный клик левой кнопкой мыши | Запуск формы «Словарь «Должность» » | Да | Работает корректно |
| Кнопка «Показать практикантов» | Одинарный клик левой кнопкой мыши | Запуск формы «Список практикантов» | Да | Работает корректно |
| Кнопка «Вернуться в главное меню» | Одинарный клик левой кнопкой мыши | Закрытие текущей формы и отображение формы «Меню» | Да | Работает корректно |
| Кнопка «Составить договор» | Одинарный клик левой кнопкой мыши | Запуск формы «Составление договора» | Да | Работает корректно |
| Кнопка «Договор» | Одинарный клик левой кнопкой мыши | Составление договора в Word | Да | Работает корректно |
| Кнопка редактирования «  » | Одинарный клик левой кнопкой мыши | Открытие формы для редактирования данных, представленных в выпадающем списке | Да | Работает корректно |
| Кнопка «Посмотреть характеристику» | Одинарный клик левой кнопкой мыши | Запуск формы «Характеристика» | Да | Работает корректно |
| Кнопка «Экспорт данных» | Одинарный клик левой кнопкой мыши | Запуск формы «Экспорт данных» | Да | Работает корректно |
| Кнопка «Экспорт» | Одинарный клик левой кнопкой мыши | Открытие диалогового окна «Экспортировать данные в Word/Excel?» | Да | Работает корректно |
| Продолжение таблицы 2.1 | | | | |
| Ответ «Да» на сообщение диалогового окна«Экспортировать данные в Word/Excel?» | Одинарный клик левой кнопкой мыши | Открывает Word/Excel с экспортированными данными | Да | Работает корректно |
| Ответ «Нет» на сообщение диалогового окна«Экспортировать данные в Word/Excel?» | Одинарный клик левой кнопкой мыши | Предлагает сохранить файл Word/Excel с экспортированными данными | Да | Работает корректно |
| Ответ «Отмена» на сообщение диалогового окна«Экспортировать данные в Word/Excel?» | Одинарный клик левой кнопкой мыши | Отменяет экспорт данных | Да | Работает корректно |
| Кнопка «Сохранить изменения» | Одинарный клик левой кнопкой мыши | Сохранить изменения в текстовом поле | Да | Работает корректно |
| Кнопка «Дополнительная информация» | Одинарный клик левой кнопкой мыши | Открытие формы «Дополнительная информация о практиканте» | Да | Работает корректно |
| Выпадающий список «Выберите практиканта» | Одинарный клик левой кнопкой мыши + одинарный клик левой кнопкой мыши на пункте списка | Отображение в таблице связанных с выбранным пунктом списка данных | Да | Работает корректно |
| Выпадающий список «Выберите практику» | Одинарный клик левой кнопкой мыши + одинарный клик левой кнопкой мыши на пункте списка | Отображение в таблице связанных с выбранным пунктом списка данных | Да | Работает корректно |
| Кнопка «Комментарии и замечания» | Одинарный клик левой кнопкой мыши | Открытие формы «Комментарии и замечания» | Да | Работает корректно |
| Кнопка «Сохранить» | Одинарный клик левой кнопкой мыши | Сохранить данные в полях: «Комментарии практиканта» и «Замечания руководителя» | Да | Работает корректно |
| Кнопка «Выбрать задание» | Одинарный клик левой кнопкой мыши | Открытие формы «Задания» | Да | Работает корректно |
| Окончаниетаблицы 2.1 | | | | |
| Кнопка «Выбрать» | Одинарный клик левой кнопкой мыши | Выбирает выделенное задание и закрывает форму «Задания» | Да | Работает корректно |

Помимо своих функциональных возможностей программа должна быть удобной, понятной и иметь приятный интерфейс. Результаты данного тестирования приведены в таблице 2.2.

Таблица 2.2 – Тестирование удобства использования

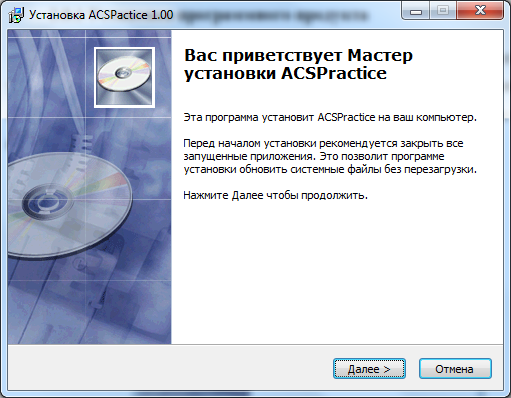
|  |  |
| --- | --- |
| Свойства программы | Их наличие и функционирование в программе |
| Графический интерфейс | Да |
| Удобство навигации приложения | Да |
| Интуитивно понятный интерфейс | Да |
| Легкая установка | Да |
| Возможность выбора каталога установки | Да |
| Запуск с ярлыка | Да |
| Небольшие системные требования | Да |
| Наличие всплывающих подсказок | Да |
| Защита от некорректных действий пользователя | Да |
| Приятная цветовая схема оформления | Да |
| Легко читаемый шрифт | Да |
| Возможность перетаскивания формы программы | Да |
| Возможность изменения размера окна формы | Главное меню, форма авторизации, диалоговые окна – нет, остальные формы – да |
| Наличие главного меню | Да |
| Наличие звуковых эффектов | Нет |
| Наличие кнопочной панели управления | Да |

Исходя из результатов тестирования, можно сказать, что испытания прошли успешно, все выявленные ошибки были устранены. Основные функции программы работают нормально, интерфейс и удобство использования соответствуют поставленным требованиям.

2.5Руководство пользователя

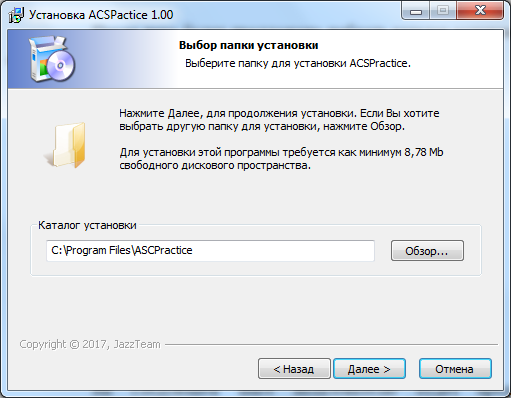
2.5.1 Установка программного продукта

Для установки приложения на компьютере необходимо запустить файл установки SetupACSPractice.exe, после чего появится окно приветствия мастера установки программы, представленное нарисунке 2.9, и нажать кнопку «Далее».



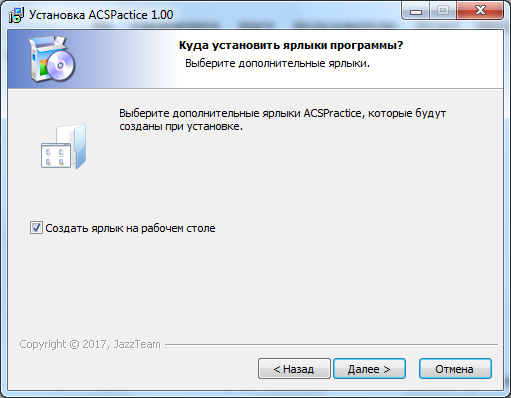
**Рисунок 2.9 – Запуск процесса установки приложения**

После чего будет предложено выбрать каталог установки. Вид окна представлен на рисунке 2.10.



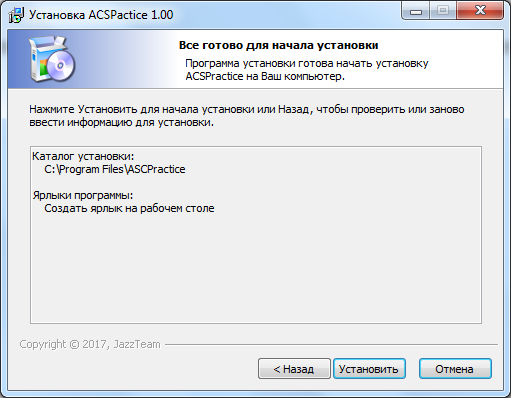
**Рисунок 2.10 – Окно для выбора каталога установки**

На следующем шаге пользователю будет предложена возможность создания ярлыка программы–рисунок 2.11.



**Рисунок 2.11 – Окно для создания ярлыка программы**

Далее будет предоставлена информация о процессе готовностик установке программного продукта–рисунок 2.12.



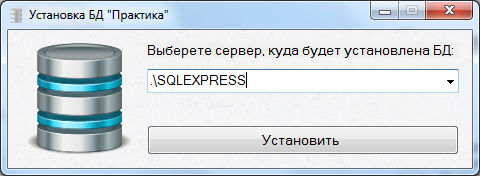
**Рисунок 2.12 – Информация о процессе готовности установки программы**

После нажатия кнопки «Установить» запустится процесс установки, по окончанию которого пользователь получит сообщение об окончании процесса установки. На рабочем столе появятся ярлыки для запуска приложения и установки базы данных–рисунок 2.13.



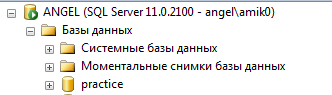
**Рисунок 2.13 – Ярлыки для запуска приложения и установки базы данных**

Для установки базы данных на сервер необходимо запустить ярлык «installBD». Появится окно установки БД, представленное нарисунке 2.14, после чего необходимо подтвердить установку, нажатием кнопки «Установить».



**Рисунок 2.14 – Окно установки базы данных**

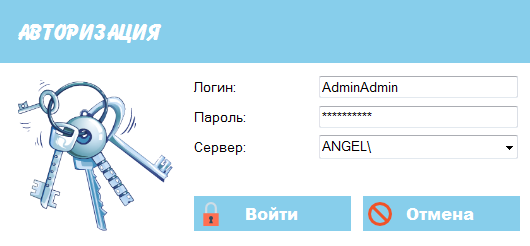
Результат добавления БД на сервер представлен на рисунке 2.15.



**Рисунок 2.15 – Результат установки базы данных на сервер пользователя**

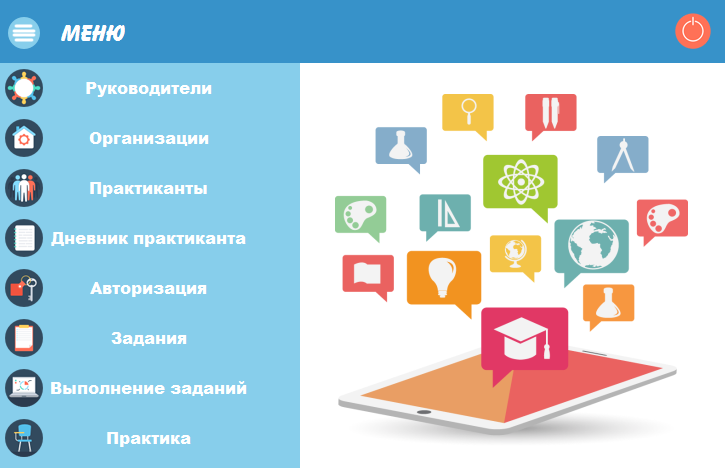
2.5.2Работа с программой

Первым, при запуске программы, появляется окно «Авторизация», представленное на рисунке 2.16, которое служит для подключения к базе данных через введенные логин и пароль. По умолчанию стоит подключение к локальному серверу.



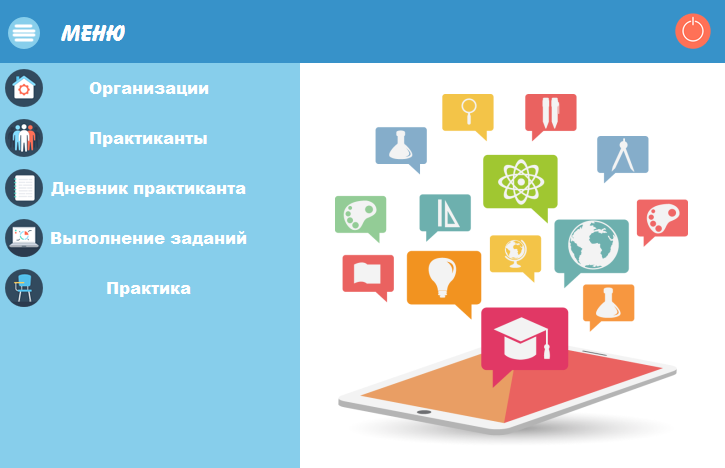
**Рисунок 2.16–Окно авторизации пользователя**

При успешной авторизации появится окно главой формы «Меню». При авторизации пользователя с правами администратора, данная форма примет вид, представленный на рисунке 2.17.



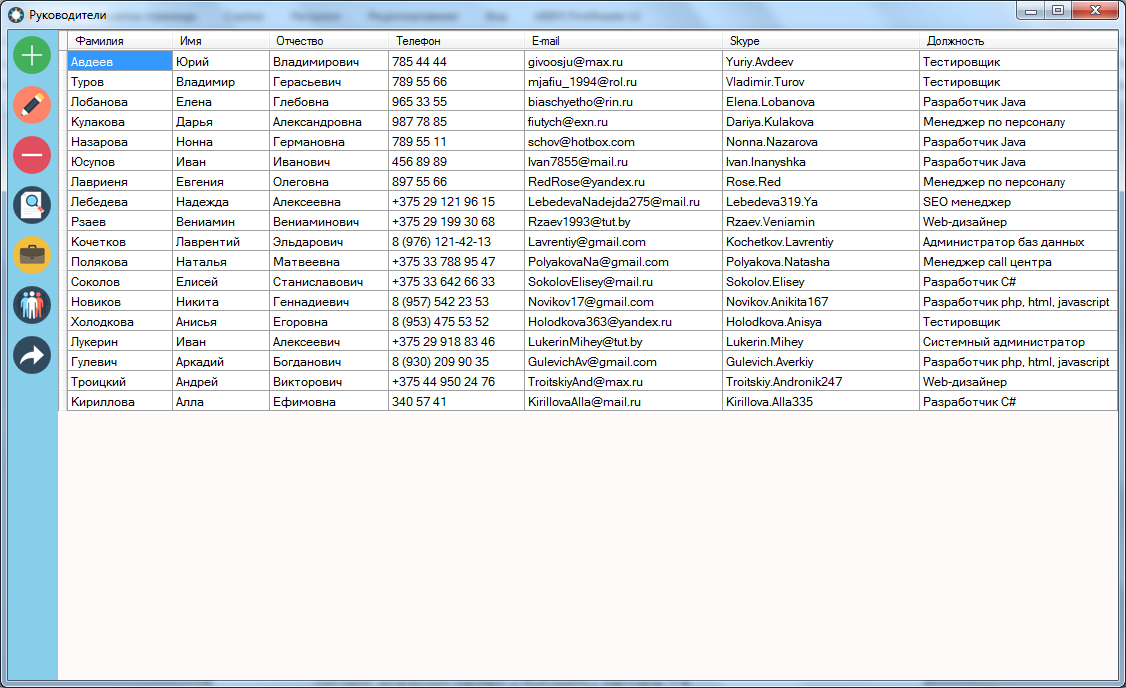
**Рисунок 2.17–Вид главной формы «Меню» для пользователя с правами администратора**

При авторизации пользователя с правами практиканта, форма примет вид, представленный на рисунке 2.18.



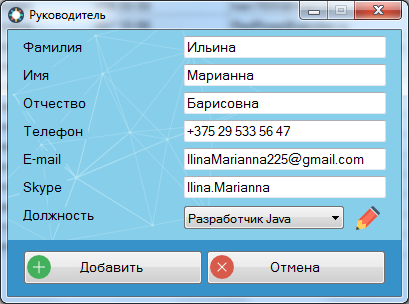
**Рисунок 2.18–Вид главной формы «Меню» для пользователя с правами администратора**

В программе есть набор основных функций, которые присутствуют на каждой форме, работающей с таблицей. Рассмотрим работу этих функций на примере взаимодействия администратора с формой «Руководители». Для этого, на главной форме приложения нажмем кнопку «Руководители», появится форма, представленная на рисунке 2.19.



**Рисунок 2.19–Вид формы «Руководители»**

Добавления нового руководителя происходит по нажатию кнопки «Добавить запись», в результате чего появится форма «Руководитель», где пользователь заполняет необходимые поля, и чтобы добавить запись в таблицу нажимает кнопку «Добавить» рисунок 2.20.



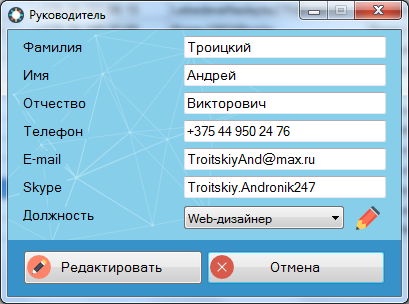
**Рисунок 2.20–Добавление новой записи при помощи формы «Руководитель»**

Если при добавлении записи не будет заполнено одно из обязательных полей, об этом появится сообщение, показанное на рисунке 2.21.



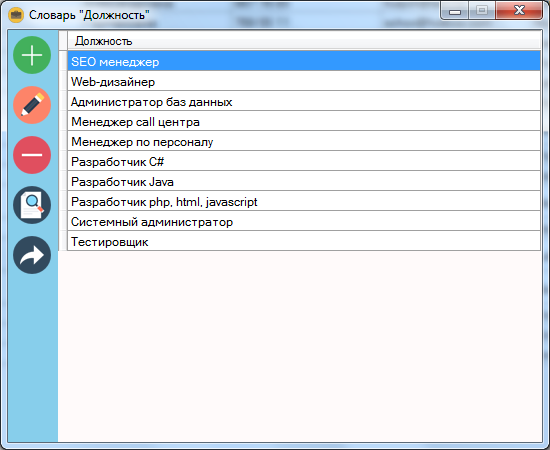
**Рисунок 2.21–Сообщение о необходимости заполнения обязательного поля**

Для редактирования записи необходимо выбрать в таблице любую ячейку в строке, которую необходимо редактировать и нажать на кнопку «Редактировать запись», или сделать два быстрых щелчка на интересующей записи в таблице. Появится форма «Руководитель», но уже с кнопкой «Редактировать», которая отвечает за сохранения произведенных в записи изменений рисунок 2.22.



**Рисунок 2.22–Редактирование записи при помощи формы «Руководитель».**

Также при добавлении или редактировании записи мы видим автозаполняемый список выбора «Должность». Аналогичные списки присутствуют и на других формах добавления или редактирования. Список можно редактировать по своему усмотрению, для этого необходимо нажать кнопку «» напротив списка. Откроется форма, отвечающая за ведение записей данного списка – рисунок 2.23.



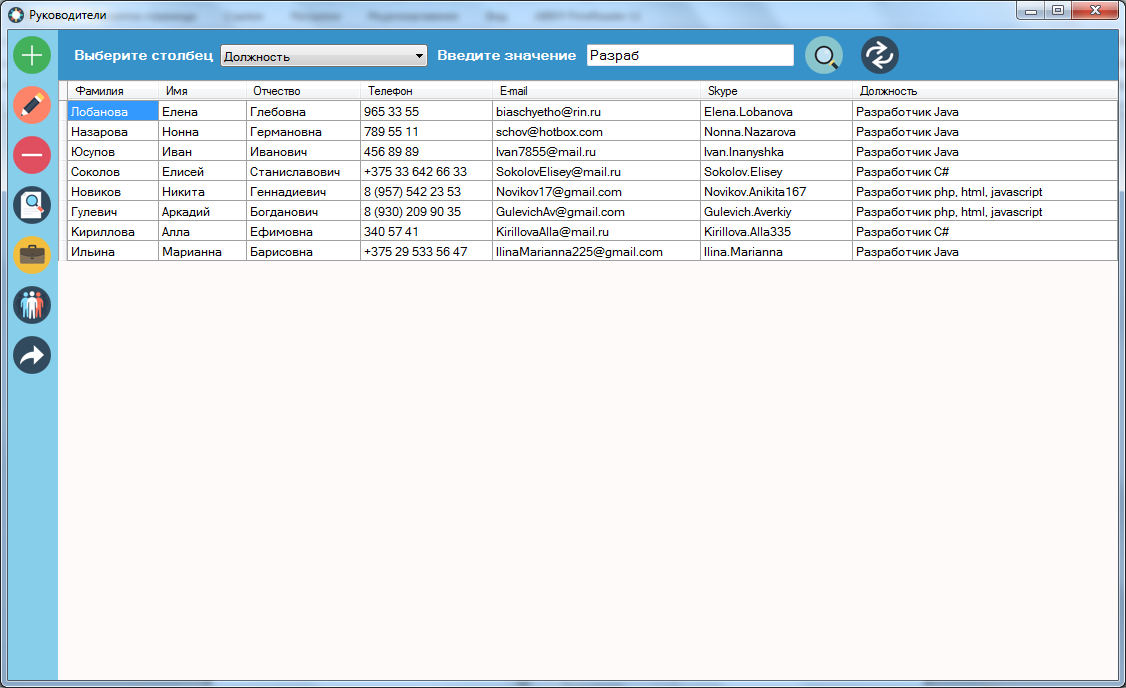
**Рисунок 2.23–Форма ведения записей о должностях руководителей**

Для удаления данных, необходимо выбрать запись в таблице и нажать кнопку «Удалить запись». Появится запрос на подтверждение удаления, приводится на рисунке 2.24. После подтверждения запроса, запись будет удалена.



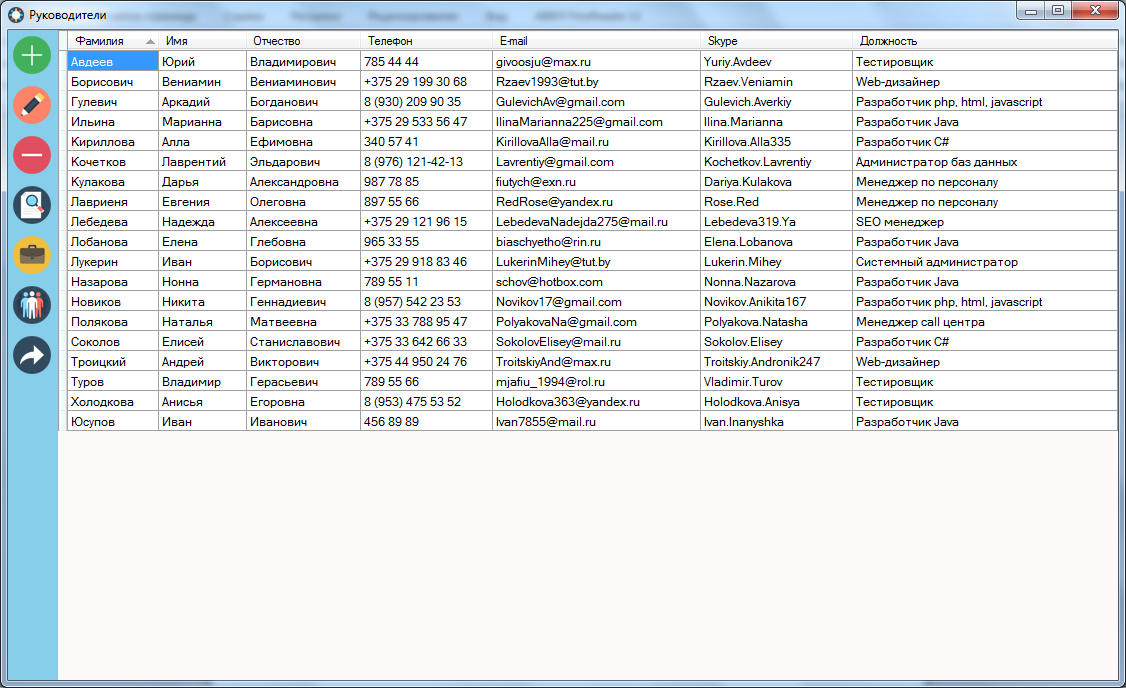
**Рисунок 2.24–Запрос на подтверждение удаления записи в таблице**

Для поиска данных нажимаем на кнопку «Поиск», сверху формы появится панель, для реализации поиска по таблице. Далее на панели поиска выбираем поле, по которому будет произведен поиск, в текстовое поле вводим данные, которые будем искать, после чего нажимаем кнопку «Начать поиск». Результат поиска представлен на рисунке 2.25. Для возвращения всех записей таблицы к исходному виду используется копка «Обновить таблицу». Если на кнопку «Поиск» нажать опять, панель скроется.



**Рисунок 2.25–Элементы формы для проведения поиска по данным**

Для сортировки записей по алфавиту в порядке возрастания или убывания, нажмите на заголовок столбца таблицы, по которому необходимо произвести сортировку. Первое нажатие сортирует данные по алфавиту в порядке возрастание, второе –в порядке убывания. Результат сортировки таблицы по столбцу «Фамилия» в порядке возрастания представлен на рисунке 2.26.

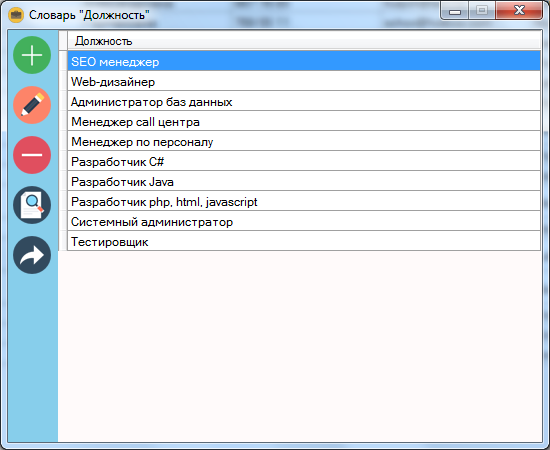


**Рисунок 2.26–Результат сортировки по столбцу «Фамилия»**

Чтобы вернуться в главное меню программы нажимаем кнопку «Вернуться в главное меню».

В программе есть также функции, которые встречаются только на определенной форме. Для формы «Руководители» это кнопки «Словарь «Должность»» и «Показать практикантов».

По нажатию на кнопку «Словарь «Должность»» появится форма, представленная на рисунке 2.27 и отвечающая за ведения списка должностей руководителей практики.



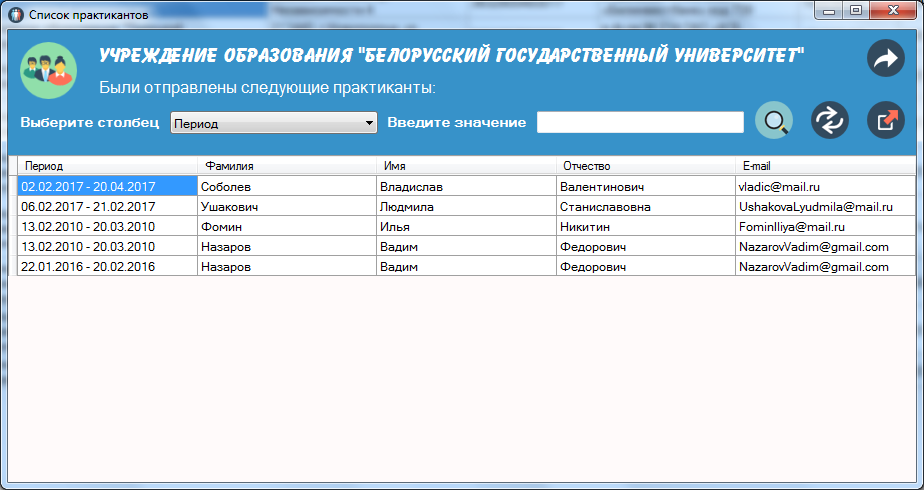
**Рисунок 2.27–Форма ведения записей о должностях руководителей**

По нажатию на кнопку «Показать практикантов» для выбранной записи руководителя, откроется форма, представленная на рисунке 2.28, которая отображает список студентов, у которых вел практику выбранный руководитель.



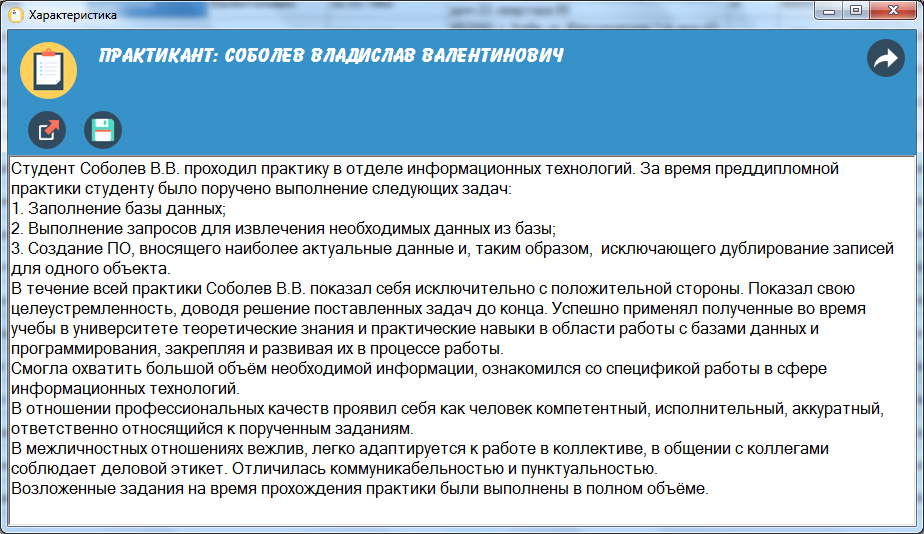
**Рисунок 2.28–Форма «Список практикантов» для руководителей**

Аналогичная функция действует на форме «Организации», только выводит список студентов, которых выбранная организация, когда-либо отправляла на практику – рисунок 2.29.



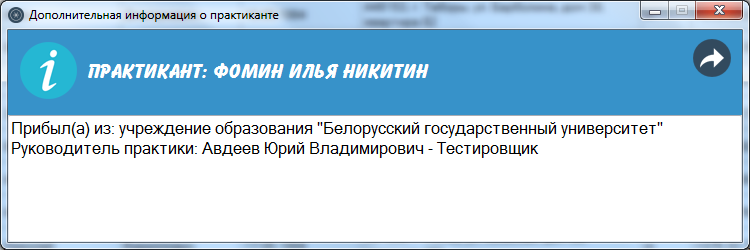
**Рисунок 2.29–Форма «Список практикантов» для организаций**

На форме «Практиканты» для выбранного студента можно посмотреть, или, если вы руководитель, написать характеристику. Для этого необходимо выбрать в таблице запись интересующего студента и нажать на кнопку «Посмотреть характеристику». Откроется форма «Характеристика» – рисунок 2.30. Написанную характеристику можно сохранить нажатием на кнопку «Сохранить изменения». Для практиканта эта кнопка не отображается.



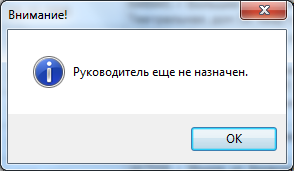
**Рисунок 2.30–Форма «Характеристика»**

Дополнительная информация о выбранном практиканте (из какой организации он прибыл и кто был назначен его руководителем практики) появится, если нажать кнопку «Дополнительная информация» –рисунок 2.31.



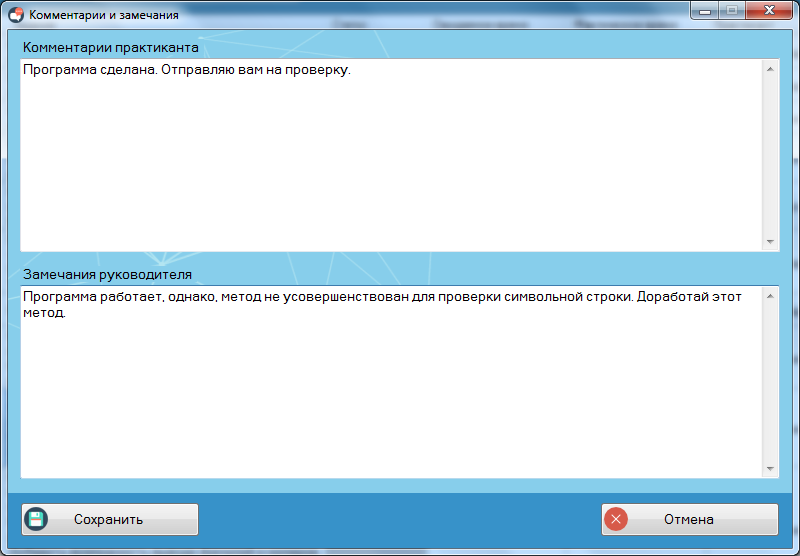
**Рисунок 2.31–Форма «Дополнительная информация о практиканте»**

Если руководитель ещё не был назначен, то перед открытием данной формы появится соответствующее сообщение, представленное на рисунке 2.32, а на самой форме отобразится только информация об учреждения, из которого прибыл практикант.



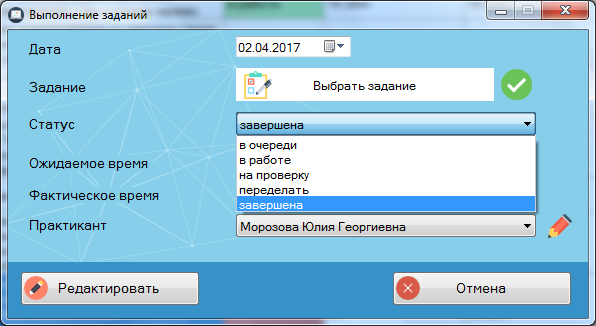
**Рисунок 2.32–Информационное сообщение о том, что руководитель практики для выбранного студента ещё не назначен**

На форме «Выполнение заданий», помимо основных функций реализован просмотр замечаний руководителя и комментариев практиканта для выбранного задания. Для этого в таблице надо выбрать интересующее задание и нажать на кнопку «Комментарии и замечания». Появится форма, представленная на рисунке 2.33. При этом поле «Замечания руководителя» не может быть отредактировано практикантом, а поле «Комментарии практиканта» не может быть отредактировано руководителем.



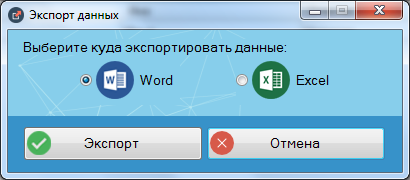
**Рисунок 2.33–Форма «Комментарии и замечания»**

При создании или редактировании выполняемой задачи интерес представляет выбор статуса выполнения. Руководитель может изменять статус на значение: в очереди, в работе, на проверку, переделать, завершена. Студенту доступны только статусы: в очереди, в работе и на проверку. Форма, где можно выбрать статус, представлена на рисунке 2.34.



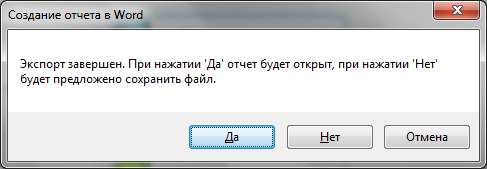
**Рисунок 2.34–Форма «Выполнение заданий»**

Также в программе предусмотрена возможность экспорта данных в Wordи Excel.Там, где данные можно экспортировать есть кнопка «Экспорт данных». При нажатии на эту кнопкупоявится окно «Экспорт данных», представленное на рисунке 2.35.



**Рисунок 2.35–Форма «Экспорт данных»**

В данном окне можно выбрать, куда именно вы хотите экспортировать данные, и нажать на кнопку «Экспорт». При этом появится диалоговое окно «Создание отчета Word/Excel»с тремя кнопками рисунок 2.36. При выборе кнопки «Да» – отчетcэкспортированными данными будет открыт, при нажатии кнопки «Нет» – будет предложено сохранить сформированный отчет, при нажатии кнопки «Отмена» – операция экспорта будет отменена.



**Рисунок 2.36–Окно выбора действия с созданным отчетом**

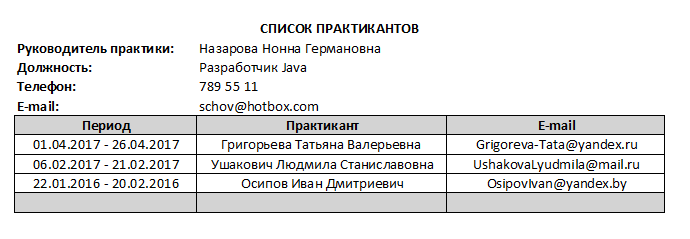
В программе существует возможность экспорта различных данных.

Экспорт списка практикантов выбранного руководителя в Word представлен нарисунке 2.37.



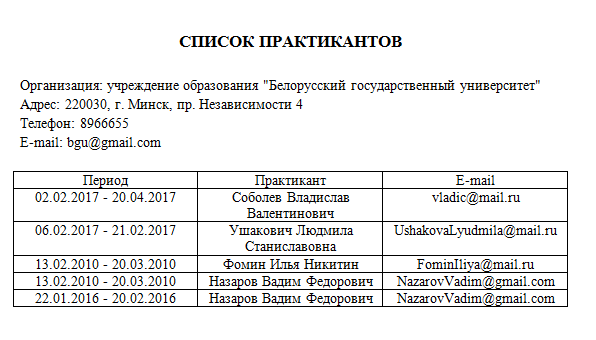
**Рисунок 2.37–Список практикантов выбранного руководителя в Word**

Экспорт списка практикантов выбранного руководителя в Excel представлен нарисунке 2.38.



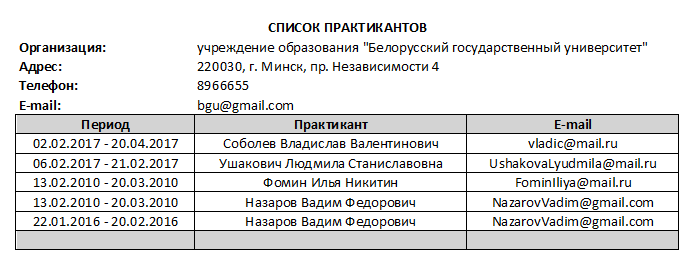
**Рисунок 2.38–Список практикантов выбранного руководителя в Excel**

Экспорт списка практикантов выбранной организации в Wordпредставлен на рисунке 2.39.



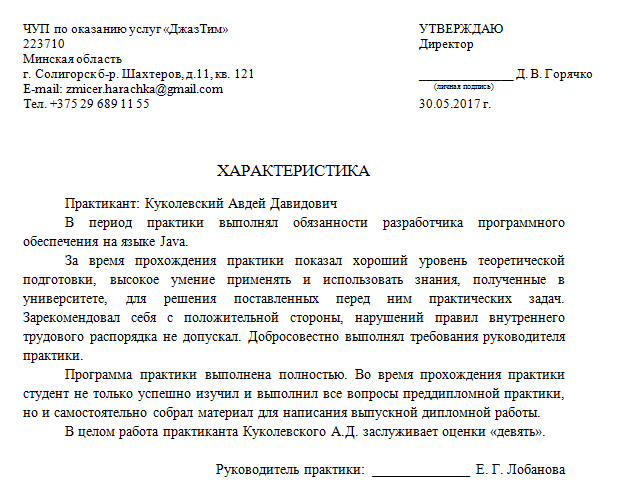
**Рисунок 2.39–Список практикантов выбранной организации в Word**

Экспорт списка практикантов выбранной организации в Excelпредставлен на рисунке 2.40.



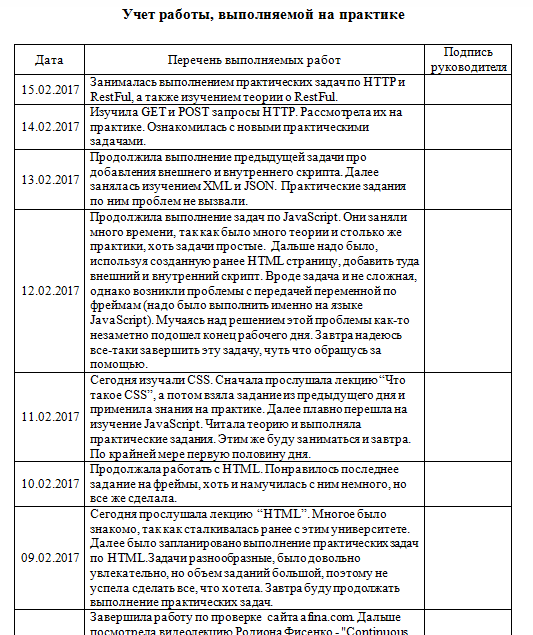
**Рисунок 2.40–Список практикантов выбранной организации в Excel**

Создание документа характеристики выбранного практиканта в Wordпредставлено на рисунке 2.41.



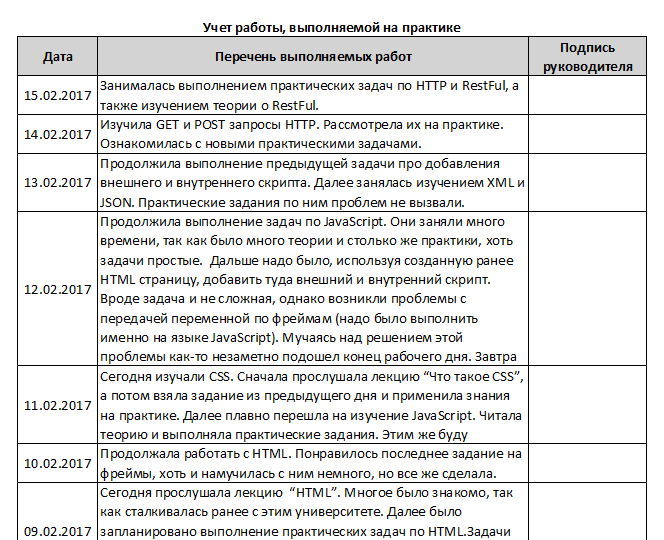
**Рисунок 2.41–Характеристика выбранного практиканта в Word**

Результат экспорта дневника практики выбранного практиканта в Word представлен на рисунке 2.42.



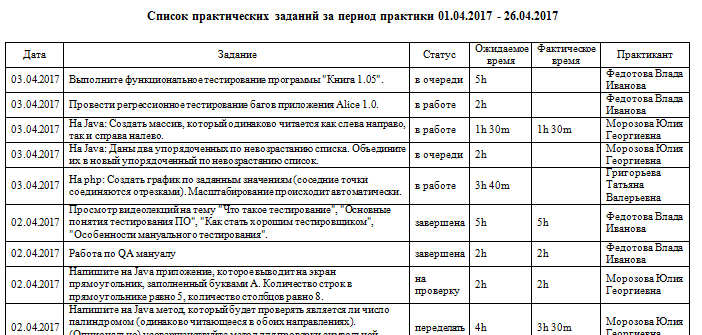
**Рисунок 2.42–Дневник практики выбранного практиканта в Word**

Результат экспорта дневника практики выбранного практиканта в Excel представлен на рисунке 2.43.



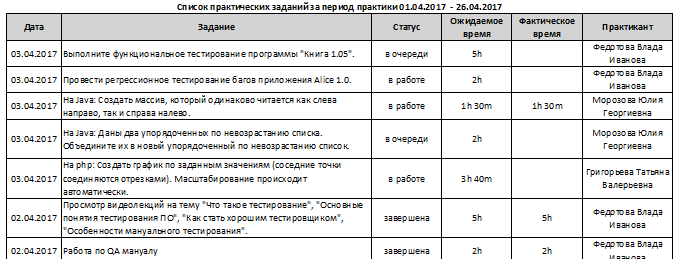
**Рисунок 2.43–Дневник практики выбранного практиканта в Excel**

Экспорт спискапрактических задач за определенный период практики в Wordпредставлен на рисунке 2.44.



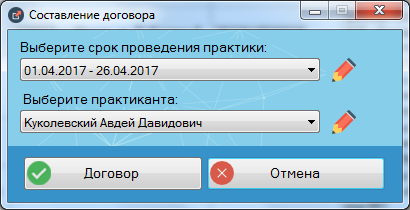
**Рисунок 2.44–Список практических задач за период практики в Word**

Экспорт списка практических задач за определенный период практики в Excelпредставлен на рисунке 2.45.



**Рисунок 2.45–Список практических задач за период практики в Excel**

К экспорту данных относится создание договора на прохождение практики в данной компании. Чтобы создать договор, на форме «Организации», нажмите кнопку «Составить договор», появится форма, представленная на рисунке 2.46.



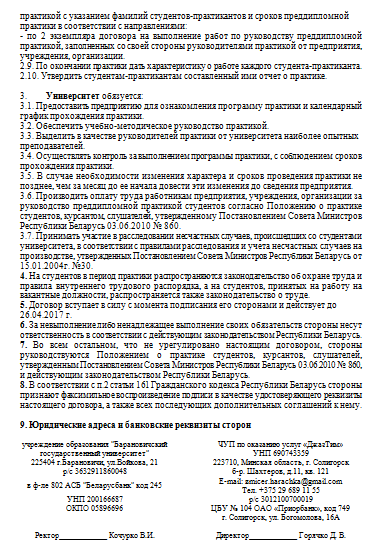
**Рисунок 2.46–Форма «Составление договора»**

Далее необходимо выбрать период практики, студента и нажать на кнопку «Договор». Составленный договор будет выведен в Word. Первая страница договора представлена нарисунке 2.47.



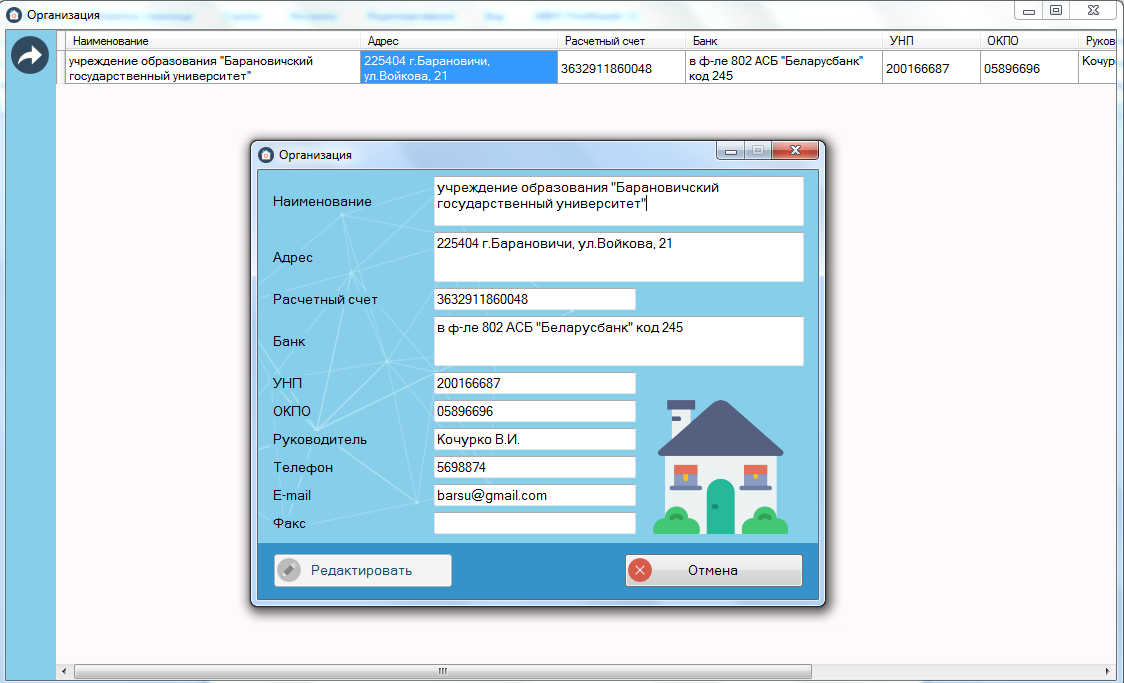
**Рисунок 2.47–Первая страница договора о прохождении практики в Word**

Вторая страница договора представлена нарисунке 2.48.



**Рисунок 2.48–Вторая страница договора о прохождении практики в Word**

Студенту в программе доступен просмотр только записей связанных непосредственно с ним. Например, на рисунке 2.49 представлен вид формы «Организации», если авторизацию прошел пользователь с правами доступа «практикант».



**Рисунок 2.49–Форма «Организации» для пользователя с правами доступа «практикант»**

Также студент может добавлять, редактировать, удалять и просматривать только записи своего дневника, а также добавлять и редактировать записи о выполнении своих заданий без возможности удаления, так как удалить выбранную для выполнения задачу имеет право только руководитель. Из экспорта данных доступен экспорт своей характеристики, дневника и списка выполненных заданий.

Для выхода из программы необходимо нажать на кнопку «» в главном меню программы.

3 ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

3.1 Определение трудоёмкости разработки программного продукта

Трудоёмкость разработки программного продукта может быть определена укрупненным методом. При этом необходимо воспользоваться формулой(3.1):

, (3.1)

где ТОА – трудоемкость подготовки описания задачи и исследования алгоритма решения;

ТБС – трудоемкость разработки блок–схемы алгоритма;

ТП – трудоемкость программирования по готовой блок–схеме;

ТОТЛ – трудоемкость отладки программы на ЭВМ;

ТДР – трудоемкость подготовки документации по задаче в рукописи;

ТДО – трудоемкость редактирования, печати и оформления документации по задаче.

Составляющие приведённой формулы определяются, в свою очередь, через условное число операторов Q в разрабатываемом ПП по формуле(3.2):

, (3.2)

где q– число операторов в программе (q = 1495);

С – коэффициент сложности программы (С = 1,1);

p–коэффициент коррекции программы в ходе её разработки (р = 0,20).

Тогда составляющие трудоёмкости разработки программы определятся по формулам(3.3) – (3.8):

(3.3)

(3.4)

(3.5)

(3.6)

(3.7)

(3.8)

где W– коэффициент увеличения затрат труда вследствие недостаточного описания задачи (W = 1,3);

K–коэффициент квалификации разработчика алгоритмов и программ (при стаже от двух до трех лет K = 1, до двух лет = 0,8).

3.2 Определение себестоимости создания программного продукта

### 3.2.1 Определение затрат на оплату труда разработчика

Для определения себестоимости создания программного продукта необходимо определить затраты на заработную плату разработчика по формуле(3.9):

(3.9)

гдеТР – трудоемкость разработки программного продукта, чел/ч;

tЧР – среднечасовая ставка работника, осуществлявшего разработку программного продукта, р;

q – коэффициент, учитывающий процент премий и доплат к тарифной ставке в организации разработки (q=0,25);

а – коэффициент, учитывающий дополнительную заработную плату (а=0,15);

b – коэффициент, учитывающий начисления на заработную плату, включая отчисления в фонд социальной защиты населения, пенсионный фонд и др. (b=0,4).

Среднечасовая ставка работника определяется, исходя из Единой тарифной сетки в Республике Беларусь, по следующей формуле(3.10):

(3.10)

где З1 — месячная тарифная ставка 1 разряда (З1 = 31), р.;

kТ — тарифный коэффициент работника 1–го разряда (kТ = 2,74);

170 — нормативное количество рабочих часов в месяце.

### 3.2.2 Определение стоимости машино-часа работы ЭВМ (для разработанного программного продукта)

Стоимость машино-часа работы ЭВМ определяются по формуле(3.11):

(3.11)

где СЭ– расходы на электроэнергию за час работы ЭВМ, р.;

АЭВМ– годовые амортизационные отчисления на обновление ЭВМ, р.;

РЭВМ– годовые затраты на ремонт и техническое обслуживание ЭВМ, р.;

АПЛ– годовая величина амортизационных отчислений на реновацию производственных площадей, занимаемых ЭВМ, р.;

РПЛ– годовые затраты на ремонт и содержание производственных площадей, р;

НН– годовая величина налога на недвижимость, р.;

ФЭВМ– годовой фонд времени работы ЭВМ, час.

Расходы на электроэнергию за час работы ЭВМ определяются по формуле(3.12):

(3.12)

где NЭ–расход энергии на час работы ЭВМ, кВт(NЭ = 0,21);

kИС– коэффициент использования энергоустановок (kИС = 0,9);

ЦЭ– стоимость 1 кВт–часа электроэнергии, р. (ЦЭ = 0,155).

Годовая величина амортизационных отчислений на реновацию ЭВМ определяется по формуле (3.13):

(3.13)

где ЦЭВМ– цена ЭВМ на момент её выпуска, руб (ЦЭВМ = 830 р.);

kУ– коэффициент удорожания ЭВМ, (kУ = 1);

kМ– коэффициент, учитывающий затраты на монтаж ЭВМ (kМ = 1,05);

НАЭВМ– норма амортизационных отчислений на ЭВМ, % (НАЭВМ = 20%);

ЦБЭВМ– балансовая стоимость ЭВМ, р. (ЦБЭВМ = 871,5 р.).

Годовые затраты на ремонт и техническое обслуживание ЭВМ укрупнено определяются по формуле(3.14):

(3.14)

где kРО– коэффициент, учитывающий затраты на ремонт и техническое обслуживание ЭВМ, в том числе затраты на запчасти, зарплату ремонтного персонала (kРО = 0,1).

Годовая величина амортизационных отчислений на реновацию производственных площадей, занятых ЭВМ определяется по формуле(3.15):

(3.15)

где – балансовая стоимость площадей, р.;

– годовая норма амортизационных отчислений на производственные площади, % (НАПЛ = 1,8%);

– площадь, занимаемая ЭВМ, кв.м. (SЭВМ = 1 м2);

– коэффициент, учитывающий дополнительную площадь (kД =3);

– цена 1 кв.м. производственной площади, р. (ЦПЛ = 418 р.).

Годовые затраты на ремонт и содержание производственных площадей могут быть определены по формуле(3.16):

(3.16)

где kРЭ– коэффициент, учитывающий затраты на ремонт и эксплуатацию производственных площадей (kРЭ = 0,05).

Величина налога на недвижимость определяется по формуле(3.17):

(3.17)

где СНН– годовая ставка налога на недвижимость, % (СНН =1,2 %).

Годовой фонд времени работы ЭВМ определяется исходя из режима её работы и рассчитывается по формуле(3.18):

(3.18)

где – среднесуточная фактическая загрузка ЭВМ, час ( = 8);

– среднее количество дней работы ЭВМ в год, дней ( = 250).

Теперь можем определить стоимость машино-часа работы ЭВМ.

Теперь можем определить стоимость машино-часа работы ЭВМ:

### 3.2.3 Определение затрат на отладку программного продукта

Затраты на отладку программы определяются по формуле(3.19):

(3.19)

где ТОТЛ– трудоёмкость отладки программы, час;

SМЧ– стоимость машино-часа работы ЭВМ, р./час.

Себестоимость разработки ПП определяется по формуле(3.20):

(3.20)

где F– коэффициент накладных расходов проектной организации без учёта эксплуатации ЭВМ (F=0,6).



3.3 Определение отпускной ценыпрограммного продукта

В основу расчёта отпускной цены закладывается затратная цена организации-разработчика, которая складывается из себестоимости создания программного продукта и плановой прибыли на программу. Затратную цену ЦЗ находим по формуле(3.21):

(3.21)

где – плановая прибыль на программу, р.

Плановая прибыль на программу определяется по формуле(3.22):

(3.22)

где СПР– себестоимость программы;

НП– норма прибыли проектной организации (НП = 0,2).

Отпускная цена программы определяется в следующей последовательности:

а) определяются отчисления в фонды и местные налоги, если установлено местными органами власти, по формуле(3.23):

(3.23)

где НМБ – норматив отчислений в фонды и местные налоги, % (НМБ= 13%);

б) определяются отчисления в республиканский фонд поддержки производителей сельскохозяйственной продукции, продовольствия и аграрной науки по формуле (3.24):

(3.24)

где НРФ – норматив отчислений в республиканский фонд поддержки производителей сельхоз продукции, % (НРФ = 0 %);

в) определяется величина налога на добавленную стоимость по формуле(3.25):

(3.25)

где НДС – ставка налога на добавленную стоимость, % (НДС = 20);

г) определяется отпускная цена программы по формуле(3.26):

(3.26)

3.4 Определение ожидаемого прироста прибыли в результате внедрения программного продукта

### 3.4.1 Определение годовых эксплуатационных расходов при ручном решении задачи

Годовые эксплуатационные расходы при ручной обработке информации (ручном решении задачи) определяются по формуле(3.27):

(3.27)

где – трудоемкость разового решения задачи вручную, ч;

– периодичность решения задачи в течение года, раз/год;

– среднечасовая тарифная ставка работника, осуществляющего ручной расчет задачи, р.;

q – коэффициент, учитывающий процент премий (q=0,275);

а – коэффициент, учитывающий дополнительную заработную плату (a=0,15);

b – коэффициент, учитывающий начисления на заработную плату (b=0,4).

### 3.4.2 Определение годовых текущих затрат, связанных с эксплуатацией программного продукта

Для расчёта годовых текущих затрат, связанных с эксплуатацией ПП, необходимо определить время решения данной задачи на ЭВМ.

Время решения задачи на ЭВМ определяется по формуле(3.28):

(3.28)

где ТВВ– время ввода в ЭВМ исходных данных, необходимых для решения задачи, мин;

ТВЫЧ– время вычислений, мин (проектный:ТВЫЧ = 0,2 мин);

ТВЫВ– время вывода результатов решения задачи (включая время распечатки на принтере), мин (проектный:ТВЫВ =1 мин);

dПЗ– коэффициент, учитывающий подготовительно-заключительное время (dПЗ = 0,175).

Время ввода в ЭВМ исходных данных может быть определено по формуле(3.29):

(3.29)

где – среднее количество знаков, набираемых с клавиатуры при вводе исходных данных (проектный: =250);

– норматив набора 100 знаков, мин (проектный:= 0,5 мин).

Затраты на заработную плату пользователя определяются по формуле(3.30):

(3.30)

где ТЗ– время решения задачи на ЭВМ, час;

tЧП– среднечасовая ставка пользователя программы, р. (tЧП= 0,5);

k–периодичность решения задачи в течение года, раз/год (k = 195);

q–коэффициент, учитывающий процент премий (q = 0,275);

а – коэффициент, учитывающий дополнительную заработную плату (a = 0,15);

b–коэффициент, учитывающий начисления на заработную плату (b = 0,4).

Затраты на оплату аренды ЭВМ для решения задачи определяются по следующей формуле(3.31):

(3.31)

где – стоимость одного машино-часа работы ЭВМ, которая будет использоваться для решения задачи, р.;

Годовые текущие затраты, связанные с эксплуатацией задачи, определяются по формуле(3.32):

(3.32)

где ЗП – затраты на заработную плату пользователя программы;

ЗА – затраты на оплату аренды ЭВМ при решении задачи;

ЗД – дополнительные затраты, связанные с эксплуатацией ПП, в частности затраты на формирование базы данных, если она не поставляется вместе с программным продуктом. (ЗД=0).

В результате получим:

### 3.4.3 Определение ожидаемого прироста прибыли в результате внедрения программного продукта

Ожидаемый прирост чистой прибыли в результате внедрения задачи взамен ручного её расчёта укрупнено может быть определён по формуле(3.33):

(3.33)

где СНП – ставка налога на прибыль, % (СНП = 18%);

3.5 Расчёт показателей эффективности использования программного продукта

Для определения годового экономического эффекта от разработанной программы необходимо определить суммарные капитальные затраты (КС) на разработку и внедрение программы по формуле(3.34):

(3.34)

где КЗ – капитальные и приравненные к ним затраты;

СПР – себестоимость программы.

Капитальные и приравненные к ним затраты в случае, если ЭВМ, на которой предполагается решать рассматриваемую задачу, отслужила к моменту расчета 0,5 года, определяются по формуле(3.35):

(3.35)

где ЦБЭВМ – балансовая стоимость комплекта вычислительной техники, необходимого для решения задачи, р.;

Годовой экономический эффект от внедрения разработанного ПП определяется по формуле(3.36):

(3.36)

где Е – коэффициент эффективности, равный ставке за кредиты на рынке долгосрочных кредитов (Е = 0,15).

Срок возврата инвестиций определяется по формуле(3.37):

(3.37)

Систематизированные данные расчета представлены в таблице 3.1.

Таблица 3.1 – Результаты экономического расчета

|  |  |  |
| --- | --- | --- |
| Наименование показателя | Варианты | |
| базовый | проектный |
| Периодичность решения задачи, раз в год | 250 | 195 |
| Годовые текущие затраты, связанные с решением задачи, руб. | 1140,82 | 12,16 |
| Отпускная цена программы, руб. | 2695,38 | |
| Прирост условной прибыли, руб./ год | 925,5 | |
| Годовой экономический эффект пользователя руб. | 680,59 | |
| Срок возврата инвестиций (окупаемость), лет | 1.76 | |

Исходя из анализа данных, представленных в таблице3.1, проект является экономически выгодным и полностью окупитьсяпримерно через 21 месяц. Благодаря использованию данного приложения годовые текущие затраты можно будет сократить на 1128,66 р. Годовой экономический эффект при этомсоставит680,59 р, а прирост условной прибыли –925,5 р. в год.

4 ОХРАНА ТРУДА

## 4.1 Необходимость охраны труда

Согласно статье 221 Трудового кодекса Республики Беларусь «охра­на труда – это система обеспечения безопасности жизни и здоровья работников в процессе тру­довой деятельности, включающая правовые, социально-экономические, организационные, техни­ческие, психофизиологические, санитарно-гигиенические, лечебно-профилактические, реабилита­ционные и иные мероприятия и средства».

Обеспечение охраны труда имеет большое социально-экономическое значение. Известно, что неудовлетворенность работника условиями труда приводить к текучести кадров со многими отри­цательными последствиями как для самого работника, так и для организации. Неудовлетвори­тельные условия труда приводят к тому, что часть работников вынуждена заканчивать свою трудовую деятельность ранее общеустановленного пенсионного возраста либо менять профессию и место работы.

Общеизвестно также, что условия труда в значительной степени влияют на производитель­ность труда. При благоприятных условиях труда работник не затрачивает сил на защиту организ­ма от воздействия опасных и вредных факторов производственной среды и трудового процесса. Имеются данные о том, что рациональный комплекс мероприятий, направленных на улучшение условий труда, может обеспечить прирост производительности труда на (6–25) и более процентов.

В то же время недостатки, упущения в работе по созданию надлежащих условий труда, нару­шения требований охраны труда приводят к травматизму на производстве, профессиональной за­болеваемости, необходимости производить дополнительные затраты на выплату компенсаций ра­ботникам по условиям труда. Особенно пагубно на экономику предприятия влияют случаи травматизма на производстве и профессиональных заболеваний со смертельным и тяжелым исходом, это в свою очередь объясняет актуальность рассмотрения вопросов охраны труда в дипломном проекте [24].

## 4.2 Анализ состояния охраны труда

### 4.2.1 Организация работы по охране труда

В соответствии с пунктом 15 статьи 226 Трудового кодекса Республики Беларусь наниматель обязан обеспечивать охрану труда работников.

В JazzTeam ответственность за соблюдение положений об охране труда несет директор компании.

Технические, санитарно-гигиенические, организационные и другие мероприятия по охране труда, направленные на обеспечение требований безопасности и гигиены труда, доведение санитарно-бытового обеспечения работников до установленных норм, осуществляемые нанимателем в плановом порядке, включаются в план мероприятий по охране труда, который оформляется в качестве приложения к коллективному договору.

Затраты на выделение и использование денежных и материальных средств для выполнения планов мероприятий за последние три года приведены в таблице 4.1.

Таблица 4.1 — Выделение и использование средств на мероприятия по охране труда

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Показатели | 2014 | | 2015 | | 2016 | |
| План | Факт | План | Факт | План | Факт |
| Всего затрат, руб. | 350 | 360 | 400 | 400 | 540 | 525 |
| В т.ч.: |  |  |  |  |  |  |
| на номенклатурные мероприятия, предусмотренные коллективным договором | 250 | 260 | 270 | 270 | 350 | 335 |
| на средства индивидуальной защиты | − | − | − | − | − | − |
| на лечебно−профилактическое питание и молоко | − | − | − | − | − | − |
| прочие мероприятия | 100 | 100 | 130 | 130 | 190 | 190 |
| ассигновано на одного работающего людей, руб. | 17,5 | 18 | 16 | 16 | 18 | 17,5 |

Исходя из данных, представленных в таблице 4.1, следует, что в 2014 году на мероприятия по охране труда было выделено 350 руб., фактические же затраты превысили выделенные средства на 10 руб., что является небольшой суммой. В 2015 году планируемые затраты были увеличены на 14% (50 руб.) и сошлись с фактическими. В 2016 году на мероприятия по охране труда выделили средств на 35% (140 руб.) больше чем в 2015 году, однако не все выделенные средства были потрачены.

Большая часть выделенных денежных средств уходит на проведение номенклатурных мероприятий, предусмотренных коллективным договором. В 2014 году фактические затраты на проведения этих мероприятий составили 260 рублей. В следующем году они возросли на 4% (10 руб.) и к 2015 году были увеличены еще на 24% (65 руб.).

Затраты на прочие мероприятия также были увеличены: в 2015 году на 30% (30 руб.) и в 2016 году на 46% (60 руб.) по сравнению с предыдущим годом.

Так как количество работников предприятия с каждым годом менялось, в зависимости от этого затраты ассигнованные на одного человека сначала уменьшились на 13% (2 руб.), а затем к 2015 году возросли на 9% (1,5 руб.).

На основе вышеизложенного можно сделать вывод, что с каждым годом выделение средств на мероприятия по охране труда увеличивается, что положительно сказывается на повышении ее качества. Компания уделяет большое внимание данному вопросу.

### 4.2.2 Техника безопасности

Компьютерное оборудование питается от электросети, поэтому среди основных опасных производственных факторов для оператора ЭВМ стоит выделить риск поражения электрическим током. По этой причине в компании JazzTeam проводят регулярные проверки целостности вилок, штепселей, розеток, соединительных шнуров, а также любых других токоведущих изделий, требующих надёжной изоляции. Питающие электрические кабели проверяются на ненарушенную изоляцию и сечение, соответствующее передаваемой мощности.

В компании соблюдаются нормы техники безопасности о положении оператора ЭВМ перед монитором. Повышенное внимание уделяется настройке яркости и контрастности работающего дисплея.

Специалисты, работающие на персональном компьютере, соблюдают режим труда и отдыха. При восьмичасовом рабочем дне, разделённом часовым обеденным перерывом, имеют два 15−минутных перерыва: через 2 часа после начала работы и спустя 2 часа после обеда. В качестве профилактики таким работникам дополнительно можно прерываться на 3−5 минут каждые полчаса работы за монитором [25].

Кроме технических устройств, в помещении операторов ПК имеются в наличии: инструкции по охране труда и пожарной безопасности; нормативно-техническая документация и методические материалы, необходимые для работы; плакаты по охране труда (в том числе и при работе на ПК); аптечка первой помощи.

В компании строго следят за тем, чтобы рабочее место оператора было аттестовано по условиям труда. Сам оператор был ознакомлен с результатами аттестации его рабочего места, знал параметры имеющихся вредных и опасных факторов, планы профилактических мероприятий, полагающиеся ему льготы и компенсации за работу во вредных условиях труда.

Основным действием на предприятии, направленным на профилактику производственного травматизма, является замена устаревшего и приобретение нового оборудования с высокой степенью защиты.

Согласно полученным официальным данным, в период с 2014−2016 гг. в компании JazzTeam случаев производственного травматизма зафиксировано не было, что в свою очередь говорит о безопасных условиях труда.

### 4.2.3 Производственная санитария

Компания JazzTeam расположена на первом этаже жилого комплекса, на территории которого выложены дорожки, обеспечен хороший подъезд к дому, организована автостоянка. Двор оборудован детской площадкой, беседками, скамейками для отдыха, проведено озеленение территории: посажены газоны, деревья и различные декоративные кустарники.

На предприятии имеется гардероб, предназначенный для хранения уличной одежды и обуви, оборудованный открытыми вешалками и стойками для обуви.

Оборудовано две уборные с умывальниками. При умывальниках есть жидкое мыло регулярно сменяемые полотенца.

В гардеробной и туалетах полы влагостойкие, с нескользкой поверхностью, светлых тонов, постоянно содержатся в сухом состоянии.

В компании также есть комната для приема пищи и отдыха, оборудованная умывальником, микроволновой печью, электрическим чайником, холодильником и кулером для снабжения питьевой водой.

Все санитарно-бытовые помещения ежесменно убираются с использованием дезинфицирующих средств (разрешенных к применению Министерством здравоохранения Республики Беларусь) и проветриваются.

 В компании JazzTeam каждый сотрудник проходит предварительные (при приеме на работу) и периодические медицинские осмотры. (Постановление Министерства здравоохранения Республики Беларусь от 8.08.2000 № 33.«О порядке проведения обязательных медицинских осмотров работников»).

### 4.2.4 Пожарная безопасность

Пожары и взрывы причиняют значительный материальный ущерб и в ряде случаев вызывают тяжелые травмы и гибель людей.

Ответственность за обеспечение пожарной безопасности в компании JazzTeam возлагается на руководителя предприятия.

Деятельность по обеспечению пожарной безопасности осуществляется в соответствии с Законом Республики Беларусь «О пожарной безопасности» от 15.06.1993 г.

В компании строго выполняются правила и требования противопожарного режима, со всеми вновь поступающими на работу проводиться противопожарный инструктаж.

В наличии имеются и готовы к использованию первичные и технические средства пожаротушения, такие как огнетушители и пожарные извещатели. Разработаны противопожарные инструкции, периодически проводится пожарная профилактика.

За анализируемый период с 2014−2016 гг. в компании не было зафиксировано ни одного пожароопасного случая, возгорания или пожара.

## 4.3 Мероприятия по улучшению условий и безопасности труда

На основании выше изложенного материала, можно сделать вывод, что в целом условия труда в компании JazzTeam соответствуют общепринятым нормам, сотрудникам обеспечены комфорт и благоприятные условия труда. В компании отсутствуют случаи производственного травматизма, профзаболеваний и несчастный случаев.

Для дальнейшего повышения качества условий труда на предприятии и улучшения безопасности труда, можно провести следующие мероприятия: улучшить контроль за выполнением требований и инструкций по охране труда и технике безопасности; для поддержания нормальной температуры и относительной влажности в помещении проводить регулярное проветривание, а так же установить системы ионизирования и кондиционирования воздуха; в течение рабочего дня менять род работы; устраивать разминку и выполнять специальные упражнения для глаз и для снятия общего утомления; для снижения уровня напряженности электростатического поля при необходимости применять экранные защитные фильтры; осуществить контроль за выполнением графиков планово-предупредительного ремонта оборудования; устранить в оборудовании конструктивные недостатки, последствия которых могут привести к травмам; выключать мониторы, на которых временно не работают, но находятся рядом с ними; протирать экран и рабочее место специальной антистатической салфеткой;

## 4.4 Разработка инструкции по охране труда для операторов ПЭВМ

### 4.4.1 Общие требования техники безопасности

* 1. К самостоятельной работе в качестве оператора персональных электронных вычислительных машин (ПЭВМ) допускаются лица не моложе 18 лет, прошедшие обучение по профессии, медицинский осмотр, инструктаж по охране труда на рабочем месте, освоившие безопасные методы и приемы выполнения работ, имеющие I квалификационную группу по электробезопасности.
  2. Женщины со времени установления беременности и в период кормления грудью к выполнению всех видов работ на ПЭВМ не допускаются.
  3. Соблюдать Правила внутреннего трудового распорядка предприятия, выполнять только порученную работу. Запрещается выполнение работы в состоянии алкогольного, токсического или наркотического опьянения. Курить разрешается только в специально установленных местах.
  4. Организация рабочего места должна учитывать требования безопасности, удобство положения, движений и действий работника.
  5. За невыполнение требований настоящей инструкции работник несет ответственность в соответствии с действующим законодательством.

### 4.4.2 Требования техники безопасности перед началом работы

* 1. Осмотреть рабочее место, убрать лишние и мешающие предметы, бумагу, документы и т.п.
  2. Проверить исправность подъемно–поворотного стула, для обеспечения рациональной рабочей позы отрегулировать взаиморасположение элементов стула в соответствии со своим ростом.
  3. Убедиться в том, что системный блок, монитор, клавиатура, печатающее и другие устройства не имеют видимых повреждений и установлены на ровных, устойчивых основаниях (столах) так, чтобы было невозможно наступить на электрические соединительные кабели, провода электропитания или зацепиться за них.
  4. Проверить исправность разъемов и правильность подключения оборудования, расположения монитора и клавиатуры.
  5. Протереть специальной салфеткой поверхность экрана.
  6. При необходимости отрегулировать положение стола, стула, угол наклона экрана монитора, расположение ПЭВМ в целях создания удобной рабочей позы, исключения длительных напряжений тела, выполнения зрительной работы вне зоны вредного воздействия излучений ПЭВМ и монитора.
  7. Перед включением оборудования проверить исправность розеток, вилок, изоляции проводов электропитания и т.п.
  8. Обо всех неисправностях, обнаруженных при проверке оборудования, мебели, инвентаря, сообщить непосредственному руководителю и до их устранения к работе не приступать.

### 4.4.3 Требования техники безопасности при выполнении работы

* 1. Содержать в чистоте и порядке рабочее место, соблюдать правила эксплуатации оборудования.
  2. Соблюдать оптимальное расстояние от экрана монитора до глаз – 600–700 мм, минимально допустимое расстояние – 500 мм.
  3. Если работник имеет нарушения зрения, пользоваться очками, подобранными окулистом с учетом рабочего расстояния от глаз до экрана монитора.
  4. При необходимости прекращения работы на некоторое время, когда отсутствует необходимость ввода информации, закрыть все активные задачи и выключить аппаратуру на рабочем месте.
  5. Соблюдать режим работы и отдыха.
  6. Во избежание поражения электрическим током, получения травмы иливыхода из строя оборудования выполнять следующие требования безопасности: не прикасаться одновременно к металлическим частям различных устройств, аппаратуры; не прикасаться к задней стенке системного блока при включенном питании; не переключать разъемы периферийных устройств работающей ПЭВМ; не допускать захламления рабочего места бумагой (в целях предотвращения накапливания органической пыли);не отключать электрическое питание ПЭВМ во время выполнения активной задачи; не прикасаться к поврежденным или неисправным выключателям, розеткам, вилкам, к проводам с поврежденной изоляцией; не допускать попадания воды или каких-либо жидкостей на поверхность системного блока, рабочую поверхность клавиатуры, дисководов, принтеров и других устройств; не допускать резких сгибов и защемления шнуров питания; не пытаться самостоятельно вскрывать и ремонтировать оборудование; не оставлять без присмотра включенное оборудование.
  7. При возникновении неисправностей во время работы выключить ПЭВМ, другие устройства, отключить их от электрической сети и сообщить об этом непосредственному руководителю.

### 4.4.4 Требования безопасности по окончании работы

1. Произвести закрытие всех активных задач.
2. Выключить источник бесперебойного питания (при его наличии), питание всех периферийных устройств и питание системного блока.
3. Для предотвращения попадания пыли закрыть клавиатуру крышкой.
4. Привести в порядок рабочее место.

**4.4.5 Требования техники безопасности в аварийных ситуациях**

* 1. К аварии или несчастному случаю могут привести следующие ситуации: выполнение работы с нарушением требований настоящей инструкции; неисправность используемого в работе оборудования; неосторожное обращение с огнем.
  2. При возникновении аварийной ситуации прекратить работу, отключить от сети электрооборудование и сообщить об этом непосредственному руководителю.
  3. В случае возникновения пожара действовать в соответствии с инструкцией о мерах пожарной безопасности.
  4. Немедленно отключить оборудование от электрической сети при появлении дыма, запаха горелой изоляции, необычного шума, повреждении розетки, вилки, изоляции провода питания, ощущении действия электрического тока при прикосновении к металлическим частям, при попадании воды или посторонних предметов внутрь оборудования и сообщить о случившемся непосредственному руководителю.
  5. При появлении боли в пальцах, кистях рук, рези в глазах, усилении сердцебиения и др. прекратить работу, выключить оборудование, сообщить о недомогании непосредственному руководителю и обратиться в здравпункт (поликлинику).
  6. При несчастном случае, который произошел с работником или очевидцем которого он стал работник обязан: оказать потерпевшему доврачебную помощь, используя препараты, имеющиеся в аптечке, и принять необходимые меры по оказанию потерпевшему медицинской помощи; принять меры к сохранению обстановки на месте происшествия (если это не создает угрозу здоровью и жизни других работников); сообщить о происшедшем непосредственному руководителю или другому должностному лицу нанимателя [26].

ЗАКЛЮЧЕНИЕ

Результатом выполнения дипломного проектирования стала автоматизированная система контроля прохождения практики студентами в компании «JazzTeam».

В ходе выполнения работы решены следующие задачи:

* разработана предметная область, определены сущности и атрибуты;
* построены и описаныконцептуальная, логическая и физическаямодели данных;
* разработана и спроектирована база данных;
* разработан программныйинтерфейс осуществляющийработу с БД;
* проведено тестирование созданного проекта;
* определен экономический эффект от внедрения разработанного программного продукта.

Были закреплены навыки работы в среде разработки программных продуктов MSVisualStudio 2012 на языке C#, а также навыки работы с системой управления базами данных MSSQLServer 2012.

Полученное программное средство позволяет:

* систематизировать и хранить информацию по выбранной предметной области;
* осуществлять работу с данными разработанной БД;
* обеспечивать целостность и безопасность данных;
* осуществлять поиск и сортировку данных;
* оптимизировать планы проведения практик;
* осуществлять контроль за выполнением студентами практических заданий;
* сократить время на составление и заполнение документации путем экспорта данных в Wordи Excel.

Разработанное приложение отвечает всем требованиям предметной области, имеет приятный интерфейс, интуитивно понятный пользователю.

Данный программный продукт разработан специально для компании JazzTeam, однако может быть легко адаптирован для использования и в других компаниях, где проводятся практики или стажировки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кириллов, В.В, Введение в реляционные базы данных: учеб. пособие / В.В. Кирилов, Г.Ю. Громов; — СПб.: БХВ–Петербург, 2009. — 454 с.
2. Алгоритм шифрования AES [Электронный ресурс]. – Режим доступа: https://www.opengsm.ru/blog/algoritm-shifrovaniya-aes/. – Дата доступа 20.05.2017.
3. Крипто графические основы безопасности [Электронный ресурс]. – Режим доступа: http://www.intuit.ru/studies/courses/28/28/lecture/ 20420?page=2. – Дата доступа 20.05.2017.
4. Microsoft Visual Studio [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_Visual\_Studio. – Дата доступа 21.05.2017.
5. Среда разработки Microsoft Visual Studio [Электронный ресурс]. – Режим доступа: http://ru.bmstu.wiki/Microsoft\_Visual\_Studio#Visual\_Studio\_2012. – Дата доступа 21.05.2017.
6. Visual Studio 2012 Professional [Электронный ресурс]. – Режим доступа: http://www.programrus.ru/other-programs/4964-visual-studio-2012-professional-110507271-rus.html. – Дата доступа 21.05.2017.
7. Visual Studio 2012 Professional [Электронный ресурс]. – Режим доступа: http://www.programrus.ru/other-programs/4964-visual-studio-2012-professional-110507271-rus.html. – Дата доступа 21.05.2017.
8. Петкович,Д. Microsoft SQL Server 2012. Руководство для начинающих: Пер. с анг./ Д. Петкович – СПб.: БХВ-Петербург, 2013 – 816 с.
9. Язык программирования C# [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/csharp/csharp. – Дата доступа 22.05.2017.
10. Введение в C#. Язык C# и платформа .NET [Электронный ресурс]. – Режим доступа: https://metanit.com/sharp/tutorial/1.1.php. – Дата доступа 22.05.2017.
11. Общие сведения о платформе .NET Framework [Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/ zw4w595w(v=vs.110).aspx. – Дата доступа 22.05.2017.
12. Введение в ADO.NET. Что такое ADO.NET? [Электронный ресурс]. – Режим доступа: https://metanit.com/sharp/adonet/1.1.php. – Дата доступа 23.05.2017.
13. Концептуальная модель базы данных [Электронный ресурс]. – Режим доступа: http://www.site-do.ru/db/db4.php. Дата доступа: 23.05.2017.
14. Модель "Сущность-связь" [Электронный ресурс]. – Режим доступа: http://www.bseu.by/it/tohod/lekcii4\_3.htm. Дата доступа: 23.05.2017.
15. Этапы разработки баз данных [Электронный ресурс]. – Режим доступа: http://inftis.narod.ru/db/db-10.htm. Дата доступа: 23.05.2017.
16. Физические модели баз данных [Электронный ресурс]. – Режим доступа: http://www.intuit.ru/studies/courses/1001/297/lecture/7415. Дата доступа: 23.05.2017.
17. Rational Rose [Электронный ресурс]. – Режим доступа: http://www.interface.ru/rational/rosemain.htm. Дата доступа: 23.05.2017.
18. Технология моделирования UML [Электронный ресурс]. – Режим доступа: http://www.exponenta.ru/soft/others/mvs/stud2/24.asp. Дата доступа 24.05.2017.
19. Теория и практика UML. Диаграмма последовательности. [Электронный ресурс]. – Режим доступа: http://it-gost.ru/articles/view\_ articles /94. Дата доступа: 24.05.2017.
20. Язык UML. Руководство пользователя. Диаграммы деятельности. [Электронный ресурс]. – Режим доступа: http://dit.isuct.ru/ivt /books/CASE/case11/ch19.htm. Дата доступа: 24.05.2017.
21. Схема классов [Электронный ресурс]. – Режим доступа: http://www.ngpedia.ru/id495554p1.html. Дата доступа: 25.05.2017.
22. Диаграммы классов [Электронный ресурс]. – Режим доступа: http://alice.pnzgu.ru/~dvn/uproc/books/uml\_user\_guide/gl\_08.htm. Дата доступа: 25.05.2017.
23. Тестирование программных продуктов [Электронный ресурс]. – Режим доступа: https://www.viva64.com/ru/t/0093/. Дата доступа: 27.05.2017.
24. Что такое охрана труда и в чем состоит ее социально-экономическое значение? [Электронный ресурс]. – Режим доступа: http://www.helper.by/chto-takoe-oxrana-truda-i-v-chem-sostoit-ee-socialno-ekonomicheskoe-znachenie.html. Дата доступа: 27.05.2017.
25. Техника безопасности при работе за ПК [Электронный ресурс]. – Режим доступа: http://www.znakcomplect.ru/poleznosti/example/ot-v-ofise/texnika-bezopasnosti-pri-rabote-za-pk.html. Дата доступа: 27.05.2017.
26. Инструкция по охране труда для оператора ПЭВМ [Электронный ресурс]. – Режим доступа: http://obrazec.by.ru.byru.info /baza03/blan2936.shtm. Дата доступа: 28.05.2017.

**ПРИЛОЖЕНИЯ**

**ПРИЛОЖЕНИЕА**

**Исходныйкод**

//классTheacher.cs

usingSystem;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Practice

{

public partial class Teacher : Form

{

public Teacher()

{

InitializeComponent();

}

//строкасоединениясБД

public string ConnectionString = "";

//строкадлясортировкиданных

private string sort = "order by [ID\_руководитель] asc";

//строка для поиска данных

publicstringpoisk = "";

//переменные для записи личного шифра руководителя, и статуса пользователя

publicstringshifr, status;

//вывод информации из БД

//источник данных куда будет выводиться информация

DataSet ds;

private void update()

{

try

{

//создаемсоединениесБД

SqlConnection connection = new SqlConnection(ConnectionString);

//открываем соединение

connection.Open();

//сложный запрос для извлечения данных

SqlCommand command = new SqlCommand("Select [ID\_руководитель],[Фамилия],[Имя],[Отчество],[Телефон],[E-mail],[Skype],[Должность] from Руководитель INNER JOIN Должность ON Руководитель.ID\_должность = Должность.ID\_должность " + poisk + " " + sort, connection);

//выполняет запрос и возвращает результат

IDataReader gh = command.ExecuteReader();

ds = new DataSet(); //объект для получения данных из таблицы

ds.Load(gh, LoadOption.Upsert, ""); //получаем данные

//присваиваем источник данных таблице

dataGridView1.DataSource = ds.Tables[0];

//скрываем в таблице столбец с личным шифром руководителя

dataGridView1.Columns[0].Visible = false;

//закрываемсоединение

gh.Close();

connection.Close();

}

//вывод сообщения при появлении непредвиденных ошибок

catch (Exception q)

{

MessageBox.Show(q.Message);

}

}

//запускфункцийназагрузкуданныхизБД;

private void Teacher\_Load(object sender, EventArgs e)

{

update();

}

//закрытиеформы

private void button7\_Click(object sender, EventArgs e)

{

Close();

}

//создаем объект для формы редактирования данных

AddTeacheraddTeach = newAddTeacher();

//открытие формы для работы с данными на добавление

private void button1\_Click(object sender, EventArgs e)

{

//передаем на другую форму переменную

addTeach.readoredit = 0;

addTeach.con = ConnectionString;

//открытие формы для работы с данными

addTeach.ShowDialog();

}

//обновление данных в таблице при активации формы

private void Teacher\_Activated(object sender, EventArgs e)

{

update();

}

//переменные для передачи на форму редактирования записи

string Fam, cName, Otch, Tel, Mail, Skype, Dolj;

//получение данных из выбранной строки в таблице

void GetData()

{

shifr = dataGridView1.CurrentRow.Cells[0].Value.ToString().TrimStart(' ').TrimEnd(' ');

Fam = dataGridView1.CurrentRow.Cells[1].Value.ToString().TrimStart(' ').TrimEnd(' ');

cName = dataGridView1.CurrentRow.Cells[2].Value.ToString().TrimStart(' ').TrimEnd(' ');

Otch = dataGridView1.CurrentRow.Cells[3].Value.ToString().TrimStart(' ').TrimEnd(' ');

Tel = dataGridView1.CurrentRow.Cells[4].Value.ToString().TrimStart(' ').TrimEnd(' ');

Mail = dataGridView1.CurrentRow.Cells[5].Value.ToString().TrimStart(' ').TrimEnd(' ');

Skype = dataGridView1.CurrentRow.Cells[6].Value.ToString().TrimStart(' ').TrimEnd(' ');

Dolj = dataGridView1.CurrentRow.Cells[7].Value.ToString().TrimStart(' ').TrimEnd(' ');

}

//открытие формы для работы с данными на редактирование

private void button2\_Click(object sender, EventArgs e)

{

if (dataGridView1.RowCount > 0)

{

GetData();

addTeach.readoredit = 1;

addTeach.con = ConnectionString;

addTeach.shifr = shifr;

addTeach.fam = Fam;

addTeach.name = cName;

addTeach.otch = Otch;

addTeach.tel = Tel;

addTeach.mail = Mail;

addTeach.skype = Skype;

addTeach.dolj = Dolj;

addTeach.ShowDialog();

}

}

//удалитьзаписьизтаблицы

private void button3\_Click(object sender, EventArgs e)

{

if (dataGridView1.RowCount != 0)

{

int p = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value);

DialogResult res = new DialogResult();

res = MessageBox.Show("Удалитьвыбраннуюзапись?",

"Внимание",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (res == DialogResult.Yes)

{

SqlConnection connection = new SqlConnection(ConnectionString);

connection.Open();

SqlCommand command = new SqlCommand("DELETE FROM Руководитель WHERE [ID\_руководитель]=" + p.ToString(), connection);

IDataReader gh = command.ExecuteReader();

gh.Close();

connection.Close();

update();

}

else

{ return; }

}

}

//открытиесловарядолжностейруководителей

private void button5\_Click(object sender, EventArgs e)

{

//открытие словаря "Должность"

Job work = new Job();

work.ConnectionString = ConnectionString;

work.ShowDialog();

}

//показать\скрыть панель поиска

bool flag=false;

private void button4\_Click(object sender, EventArgs e)

{

comboBox4.SelectedIndex = 0;

if (flag == false)

{

flag = true;

textBox6.Clear();

panel1.Height = 50;

}

else

{

flag = false;

panel1.Height = 0;

}

}

//организацияпоискапотаблице

private void button6\_Click(object sender, EventArgs e)

{

poisk = "where [" + comboBox4.SelectedItem + "] LIKE '%" + textBox6.Text + "%'";

update();

}

private void button7\_Click\_1(object sender, EventArgs e)

{

Close();

}

//возвращение таблицы к исходному виду после поиска

private void button10\_Click(object sender, EventArgs e)

{

sort = "order by [ID\_руководитель] asc";

poisk = "";

update();

}

private void button8\_Click\_1(object sender, EventArgs e)

{

Close();

}

//открытие формы для просмотра практикантов конкретного руководителя

private void button9\_Click(object sender, EventArgs e)

{

if (dataGridView1.RowCount != 0)

{

ViewTrainee myTeam = new ViewTrainee();

GetData();

myTeam.con = ConnectionString;

myTeam.shifr = shifr;

myTeam.flag = 0;

myTeam.fio = Fam + " " + cName + " " + Otch;

myTeam.work = Dolj;

myTeam.tel = Tel;

myTeam.email = Mail;

myTeam.ShowDialog();

}

}

private void button10\_Click\_1(object sender, EventArgs e)

{

sort = "order by [ID\_руководитель] asc";

poisk = "";

update();

}

//открытие формы для работы с данными на редактирование при двойном щелчке на записи в таблице

private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

button2\_Click(this, EventArgs.Empty);

}

//в поле поиска можно производить поиск по нажатию на Enter

private void textBox6\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter) {

button6\_Click(this, EventArgs.Empty);

}

}

}

}

//классMenu.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Threading;

namespace Practice

{

public partial class Menu : Form

{

public Menu()

{

InitializeComponent();

}

publicstringConStr;//строкасоединениясБД

publicstringstatus,id; //статус пользователя и его личный id

//выходизпрограммы

private void button9\_Click(object sender, EventArgs e)

{

//диалогподтверждениязакрытияпрограммы

DialogResult rsl = MessageBox.Show("Выдействительнохотитевыйтиизприложения?", "Внимание!", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

// если пользователь нажал кнопку да

if (rsl == DialogResult.Yes)

{

// выход из приложения

Application.Exit();

}

}

//перетаскивание окна при нажатии на panel

[DllImport("user32", CharSet = CharSet.Auto)]

internal extern static bool PostMessage(IntPtr hWnd, uint Msg, uint WParam, uint LParam);

[DllImport("user32", CharSet = CharSet.Auto)]

internal extern static bool ReleaseCapture();

const uint WM\_SYSCOMMAND = 0x0112;

const uint DOMOVE = 0xF012;

const uint DOSIZE = 0xF008;

private void panel1\_MouseDown(object sender, MouseEventArgs e)

{

ReleaseCapture();

PostMessage(this.Handle, WM\_SYSCOMMAND, DOMOVE, 0);

}

private void button1\_Click(object sender, EventArgs e)

{

//открытиеформы "Руководители"

Teacher teach = new Teacher();

teach.ConnectionString = ConStr;

teach.ShowDialog();

}

private void Menu\_Load(object sender, EventArgs e)

{

//если пользователь зашел как практикант, скрыть некоторые элементы управления

if (status == "Практикант")

{

button1.Visible = false;

button5.Visible = false;

button6.Visible = false;

}

}

private void button2\_Click(object sender, EventArgs e)

{

//Открытиеформы "Организация"

Organization org = new Organization();

org.ConnectionString = ConStr;

org.status = status;

org.idPractik = id;

org.ShowDialog();

}

private void button3\_Click(object sender, EventArgs e)

{

//Открытиеформы "Практиканты"

Trainees practic = new Trainees();

practic.ConnectionString = ConStr;

practic.status = status;

practic.idPractik = id;

practic.ShowDialog();

}

private void button4\_Click(object sender, EventArgs e)

{

//Открытие формы "Дневник практиканта"

Diary diar = new Diary();

diar.ConnectionString = ConStr;

diar.status = status;

diar.idPractik = id;

diar.ShowDialog();

}

private void button6\_Click(object sender, EventArgs e)

{

//Открытие формы "Задания"

Task tasks = new Task();

tasks.ConnectionString = ConStr;

tasks.ShowDialog();

}

private void button8\_Click(object sender, EventArgs e)

{

//Открытиеформы "Практика"

Intern inter = new Intern();

inter.ConnectionString = ConStr;

inter.status = status;

inter.idPractik = id;

inter.ShowDialog();

}

private void button7\_Click(object sender, EventArgs e)

{

Work newWork = new Work();

if (status == "Практикант")

{

try

{

//проверка, если практиканту не была назначена практика, не допускать его к ведению заданий

SqlConnection conn = new SqlConnection(ConStr);

conn.Open();

SqlCommand command = new SqlCommand("SELECT Период.ID\_период, CONCAT(convert(varchar,Период.Начало,104),' - ', convert(varchar,Период.Конец,104)) AS Период FROM Период INNER JOIN Практика ON Период.ID\_период = Практика.ID\_период INNER JOIN Практикант ON Практика.ID\_практикант = Практикант.ID\_практикант WHERE Практикант.ID\_практикант='" + Convert.ToInt32(id) + "' ;", conn);

if (command.ExecuteScalar() == null)

{

MessageBox.Show("Вы не можете выбрать себе задание, так как вам еще не была назначена практика." + Environment.NewLine + "Для решения проблемы обратитесь к руководителю практики.", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

newWork.ConnectionString = ConStr;

newWork.ST = status;

newWork.idPractik = id;

newWork.ShowDialog();

}

conn.Close();

}

catch (Exception q)

{

MessageBox.Show(q.Message);

}

}

else

{

newWork.ConnectionString = ConStr;

newWork.ST = status;

newWork.idPractik = id;

newWork.ShowDialog();

}

}

private void button5\_Click(object sender, EventArgs e)

{

//открытиеформы «Авторизация»

Authorization newAuthorization = new Authorization();

newAuthorization.ConnectionString = ConStr;

newAuthorization.ShowDialog();

}

}

}

//классAddTeacher.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Practice

{

public partial class AddTeacher : Form

{

publicAddTeacher()

{

InitializeComponent();

}

//данные, которые форма получает из другой формы при загрузке

public int readoredit;

public string con;

public string shifr;

public string fam;

public string name;

public string otch;

public string tel;

public string mail;

public string skype;

public string dolj;

//процедура очистки текстовых полей

voidClear()

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

}

//закрытиеформы

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

//настройкиформыпризагрузке

private void AddTeacher\_Load(object sender, EventArgs e)

{

//загружаетсясписокдолжностейруководителей

SqlConnection conn = new SqlConnection(con);

conn.Open();

SqlCommand command = new SqlCommand("SELECT [ID\_должность], [Должность] FROM Должность order by [Должность] asc;", conn);

IDataReader g = command.ExecuteReader();

DataSet ds = new DataSet();

ds.Load(g, LoadOption.Upsert, "");

comboBox4.DataSource = ds.Tables[0];

comboBox4.DisplayMember = "Должность";

comboBox4.ValueMember = "ID\_должность";

g.Close();

conn.Close();

comboBox4.SelectedIndex = 0;

//если форма открыта на добавление

if (readoredit == 0)

{

button1.Visible = true;

button3.Visible = false;

Clear();

}

//если форма открыта на редактирование

if (readoredit == 1)

{

//полученные из другой формы данные вставляем в элементы формы

int index = comboBox4.FindString(dolj);

comboBox4.SelectedIndex = index;

textBox1.Text = fam;

textBox2.Text = name;

textBox3.Text = otch;

textBox4.Text = tel;

textBox5.Text = mail;

textBox6.Text = skype;

button1.Visible = false;

button3.Visible = true;

}

}

//добавитьзапись

private void button1\_Click(object sender, EventArgs e)

{

//проверка на отсутствие пустых обязательных полей

if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "")

{

MessageBox.Show("Поля: 'Фамилия', 'Имя', 'Отчество' должны быть заполнены!");

}

else

{

//организация выполнения запроса на добавление записи

SqlConnection conn = new SqlConnection(con);

conn.Open();

SqlCommand command = new SqlCommand("INSERT INTO [Руководитель] (Фамилия,Имя,Отчество,Телефон,[E-mail],Skype,ID\_должность) VALUES ('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "','" + comboBox4.SelectedValue.ToString() + "');", conn);

IDataReader g = command.ExecuteReader();

g.Close();

conn.Close();

Clear();

Close();

}

}

//Редактированиезаписи

private void button3\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "")

{

MessageBox.Show("Поля: 'Фамилия', 'Имя', 'Отчество' должны быть заполнены!");

}

else

{

//организация выполнения запроса на редактирование записи

SqlConnection conn = new SqlConnection(con);

conn.Open();

SqlCommand command = new SqlCommand("UPDATE [Руководитель] SET [Фамилия]='" + textBox1.Text + "',[Имя]='" + textBox2.Text + "',[Отчество]='" + textBox3.Text + "',[Телефон]='" + textBox4.Text + "',[E-mail]='" + textBox5.Text + "',[Skype]='" + textBox6.Text + "',[ID\_должность]='" + comboBox4.SelectedValue.ToString() + "' WHERE [ID\_руководитель]=" + Convert.ToInt32(shifr) + ";", conn);

IDataReader g = command.ExecuteReader();

g.Close();

conn.Close();

Clear();

Close();

}

}

//открытие словаря должностей

private void button4\_Click(object sender, EventArgs e)

{

//открытиесловаря "Должность"

Jobwork = newJob();

work.ConnectionString = con;

work.ShowDialog();

}

}

}

//класс Crypt.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Security.Cryptography;

using System.IO;

namespace Practice

{

class Crypt

{

// 128bit(16byte)IV and Key

private const string AesIV = @"!QAZ2WSX#EDC4RFV";

private const string AesKey = @"5TGB&YHN7UJM(IK<";

/// <summary>

/// AES Encryption

/// </summary>

public string Encrypt(string text)

{

// AesCryptoServiceProvider

AesCryptoServiceProvider aes = new AesCryptoServiceProvider();

aes.BlockSize = 128;

aes.KeySize = 128;

aes.IV = Encoding.UTF8.GetBytes(AesIV);

aes.Key = Encoding.UTF8.GetBytes(AesKey);

aes.Mode = CipherMode.CBC;

aes.Padding = PaddingMode.PKCS7;

// Convert string to byte array

byte[] src = Encoding.Unicode.GetBytes(text);

// encryption

using (ICryptoTransform encrypt = aes.CreateEncryptor())

{

byte[] dest = encrypt.TransformFinalBlock(src, 0, src.Length);

// Convert byte array to Base64 strings

return Convert.ToBase64String(dest);

}

}

/// <summary>

/// AES decryption

/// </summary>

public string Decrypt(string text)

{

// AesCryptoServiceProvider

AesCryptoServiceProvider aes = new AesCryptoServiceProvider();

aes.BlockSize = 128;

aes.KeySize = 128;

aes.IV = Encoding.UTF8.GetBytes(AesIV);

aes.Key = Encoding.UTF8.GetBytes(AesKey);

aes.Mode = CipherMode.CBC;

aes.Padding = PaddingMode.PKCS7;

// Convert Base64 strings to byte array

byte[] src = System.Convert.FromBase64String(text);

// decryption

using (ICryptoTransform decrypt = aes.CreateDecryptor())

{

byte[] dest = decrypt.TransformFinalBlock(src, 0, src.Length);

return Encoding.Unicode.GetString(dest);

}

}

}

}

//класс ExportTreaty

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using Microsoft.Office.Interop.Word;

using Word = Microsoft.Office.Interop.Word;

namespace Practice

{

public partial class ExportTreaty : Form

{

public ExportTreaty()

{

InitializeComponent();

}

public string cs;

public string nameOrg,shifr;

public string adr;

public string score;

public string bank;

public string ynp;

public string okpo;

public string master;

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

void update()

{

comboBox1.DataSource = null;

comboBox2.DataSource = null;

try

{

IDataReader g=null;

SqlConnection conn = new SqlConnection(cs);

conn.Open();

SqlCommand command = new SqlCommand("SELECT CONCAT(convert(varchar,Период.Начало,104),' - ', convert(varchar,Период.Конец,104)) AS Период FROM Период order by [Начало] desc;", conn);

if (command.ExecuteScalar() == null)

{

comboBox1.Items.Add("Ещенезаданниодинпериодпрактики!");

comboBox1.Enabled = false;

while (comboBox1.Text == "Еще не задан ни один период практики!")

{

toolTip1.IsBalloon = true;

toolTip1.ShowAlways = true;

toolTip1.Show("Добавьте запись о периоде.", button5);

}

}

else

{

comboBox1.Enabled = true;

g = command.ExecuteReader();

DataSet ds1 = new DataSet();

ds1.Load(g, LoadOption.Upsert, "");

comboBox1.DataSource = ds1.Tables[0];

comboBox1.DisplayMember = "Период";

}

command = new SqlCommand("SELECT CONCAT(Фамилия,' ',Имя,' ',Отчество) AS Практикант FROM Практикант WHERE ID\_организация='"+Convert.ToInt32(shifr)+"' order by [Фамилия] asc;", conn);

if (command.ExecuteScalar() == null)

{

comboBox2.Items.Add("Студентыэтойорганизацииещенедобавлены!");

comboBox2.Enabled = false;

if (comboBox2.Text == "Студенты этой организации еще не добавлены!")

{

toolTip1.IsBalloon = true;

toolTip1.ShowAlways = true;

toolTip1.Show("Добавьте запись о студенте.", button4);

}

}

else

{

comboBox2.Enabled = true;

g = command.ExecuteReader();

DataSet ds2 = new DataSet();

ds2.Load(g, LoadOption.Upsert, "");

comboBox2.DataSource = ds2.Tables[0];

comboBox2.DisplayMember = "Практикант";

}

g.Close();

conn.Close();

comboBox1.SelectedIndex = 0;

comboBox2.SelectedIndex = 0;

}

catch (Exception q)

{

MessageBox.Show(q.Message);

}

}

private void ExportTreaty\_Load(object sender, EventArgs e)

{

update();

}

//объявлениеобъектаWord

Word.\_Application oWord;

private void button1\_Click(object sender, EventArgs e)

{

// Считывает шаблон и сохраняет измененный в новом

oWord = new Word.Application();

Word.\_Document oDoc = GetDoc(Environment.CurrentDirectory + "\\act.dotx");

object oMissing = System.Reflection.Missing.Value;

objectoEndOfDoc = "\\endofdoc";

//открытие диалогового окна на экспорт данных в Word

SaveFileDialog saveFileDialog1 = new SaveFileDialog();

DialogResult res = MessageBox.Show("Экспорт завершен. При нажатии 'Да' отчет будет открыт, при нажатии 'Нет' будет предложено сохранить файл.", "Создание отчета в Word", MessageBoxButtons.YesNoCancel);

if (res == DialogResult.Yes)

{

oWord.Visible = true;

}

if (res == DialogResult.No)

{

string fileName = String.Empty;

saveFileDialog1.Filter = "doc files (\*.doc)|\*.doc|docx files (\*.docx)|\*.docx";

saveFileDialog1.FilterIndex = 1;

saveFileDialog1.RestoreDirectory = true;

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

fileName = saveFileDialog1.FileName;

}

else

return;

//сохраняем Workbook

oDoc.SaveAs(fileName, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

saveFileDialog1.Dispose();

oDoc.Close();

oDoc = null;

oWord.Quit(ref oMissing, ref oMissing, ref oMissing);

}

if (res == DialogResult.Cancel)

{

MessageBox.Show("Экспортданныхотменен.");

oDoc.Close(Word.WdSaveOptions.wdDoNotSaveChanges);

oDoc = null;

oWord.Quit(ref oMissing, ref oMissing, ref oMissing);

}

}

private Word.\_Document GetDoc(string path)

{

Word.\_Document oDoc = oWord.Documents.Add(path);

SetTemplate(oDoc);

return oDoc;

}

//установкапараметровшаблонаword

private void SetTemplate(Word.\_Document oDoc)

{

//удалить все слова после определенного знака

string data1 = comboBox1.Text.Substring(0, comboBox1.Text.IndexOf(" ") + 1);

//удалить подстроку из строки

string data2 = comboBox1.Text.Replace(data1 + "- ", "");

string data3 = DateTime.Now.ToString("dd.MM.yyyy");

oDoc.Bookmarks["name1"].Range.Text = nameOrg[0].ToString().ToLower() + nameOrg.Substring(1);

oDoc.Bookmarks["name2"].Range.Text = nameOrg[0].ToString().ToUpper() + nameOrg.Substring(1);

oDoc.Bookmarks["name3"].Range.Text = nameOrg[0].ToString().ToLower() + nameOrg.Substring(1);

oDoc.Bookmarks["name4"].Range.Text = nameOrg[0].ToString().ToLower() + nameOrg.Substring(1);

oDoc.Bookmarks["addr"].Range.Text = adr;

oDoc.Bookmarks["dt1"].Range.Text = data1+" г";

oDoc.Bookmarks["dt2"].Range.Text = data2 + " г";

oDoc.Bookmarks["dt3"].Range.Text = data2 + " г";

oDoc.Bookmarks["dt4"].Range.Text = data3 + " г";

oDoc.Bookmarks["fil"].Range.Text = bank;

oDoc.Bookmarks["okpo"].Range.Text ="ОКПО "+ okpo;

oDoc.Bookmarks["prac"].Range.Text =comboBox2.Text;

oDoc.Bookmarks["rs"].Range.Text = "р/с " + score;

oDoc.Bookmarks["ynn"].Range.Text ="УНП "+ ynp;

oDoc.Bookmarks["main1"].Range.Text = master;

oDoc.Bookmarks["main2"].Range.Text = master;

}

private void ExportTreaty\_Activated(object sender, EventArgs e)

{

update();

}

private void button5\_Click(object sender, EventArgs e)

{

Period addPeriod = new Period();

addPeriod.ConnectionString = cs;

addPeriod.ShowDialog();

}

private void button4\_Click(object sender, EventArgs e)

{

Trainees addTrainees = new Trainees();

addTrainees.ConnectionString = cs;

addTrainees.ShowDialog();

}

}

}

//класс ExportTrainee.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Microsoft.Office.Interop.Word;

using Word = Microsoft.Office.Interop.Word;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

namespace Practice

{

public partial class ExportTrainee : Form

{

public ExportTrainee()

{

InitializeComponent();

}

public string fio, work, tel, email;

public int flag = 0;

public System.Data.DataTable dt2 = new System.Data.DataTable();

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

//Экспорт данныхвword

Word.\_Application oWord;

private void button1\_Click(object sender, EventArgs e)

{

if (radioButton1.Checked == true)

{

if (dt2.Rows.Count > 0)

{

// Считывает шаблон и сохраняет измененный в новом

oWord = new Word.Application();

Word.\_Document oDoc = GetDoc(Environment.CurrentDirectory + "\\Otch01.dotx");

object oMissing = System.Reflection.Missing.Value;

object oEndOfDoc = "\\endofdoc";

Word.Paragraph oPara2;

object oRng = oDoc.Bookmarks.get\_Item(ref oEndOfDoc).Range;

oPara2 = oDoc.Content.Paragraphs.Add(ref oRng);

oPara2.Format.SpaceBefore = 3;

oPara2.Range.InsertParagraphAfter();

var t1 = Word.WdDefaultTableBehavior.wdWord9TableBehavior;

var t2 = Word.WdAutoFitBehavior.wdAutoFitFixed;

oWord.ActiveDocument.Tables.Add(oDoc.Paragraphs[8].Range, dt2.Rows.Count + 1, 3, t1, t2);

oWord.ActiveDocument.Tables[1].Cell(1, 1).Range.InsertAfter("Период");

oWord.ActiveDocument.Tables[1].Cell(1, 2).Range.InsertAfter("Практикант");

oWord.ActiveDocument.Tables[1].Cell(1, 3).Range.InsertAfter("E-mail");

for (int p = 2; p <= dt2.Rows.Count + 1; p++)

{

oWord.ActiveDocument.Tables[1].Cell(p, 1).Range.InsertAfter(dt2.Rows[p - 2][0].ToString());

oWord.ActiveDocument.Tables[1].Cell(p, 2).Range.InsertAfter(dt2.Rows[p - 2][1].ToString() + " " + dt2.Rows[p - 2][2].ToString() + " " + dt2.Rows[p - 2][3].ToString());

oWord.ActiveDocument.Tables[1].Cell(p, 3).Range.InsertAfter(dt2.Rows[p - 2][4].ToString());

}

SaveFileDialog saveFileDialog1 = new SaveFileDialog();

DialogResult res = MessageBox.Show("Экспорт завершен. При нажатии 'Да' отчет будет открыт, при нажатии 'Нет' будет предложено сохранить файл.", "Создание отчета в Word", MessageBoxButtons.YesNoCancel);

if (res == DialogResult.Yes)

{

oWord.Visible = true;

}

if (res == DialogResult.No)

{

string fileName = String.Empty;

saveFileDialog1.Filter = "doc files (\*.doc)|\*.doc|docx files (\*.docx)|\*.docx";

saveFileDialog1.FilterIndex = 1;

saveFileDialog1.RestoreDirectory = true;

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

fileName = saveFileDialog1.FileName;

}

else

return;

//сохраняем Workbook

oDoc.SaveAs(fileName, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

saveFileDialog1.Dispose();

oDoc.Close();

oDoc = null;

oWord.Quit(ref oMissing, ref oMissing, ref oMissing);

}

if (res == DialogResult.Cancel)

{

MessageBox.Show("Экспортданныхотменен.");

oDoc.Close(Word.WdSaveOptions.wdDoNotSaveChanges);

oDoc = null;

oWord.Quit(ref oMissing, ref oMissing, ref oMissing);

}

}

else

{

MessageBox.Show("В таблице нет записей!");

}

}

if (radioButton2.Checked == true)

{

if (dt2.Rows.Count > 0)

{

//создаем экземпляр класа Excel.Application

Excel.Application exApp = new Excel.Application();

//создаем новую книгу

Excel.Workbook Wb = exApp.Workbooks.Add();

Worksheet workSheet = exApp.ActiveSheet;

object \_missingObj = System.Reflection.Missing.Value;

// Выделяем диапазон ячеек от A1 до C1

Excel.Range \_excelCells = (Excel.Range)workSheet.get\_Range("A1", "C1").Cells;

// Производим объединение

\_excelCells.Merge(Type.Missing);

\_excelCells.Value = "СПИСОК ПРАКТИКАНТОВ";

if (flag == 0)

{

workSheet.Cells[2, 1] = "Руководитель практики: ";

workSheet.Cells[2, 2] = fio;

workSheet.Cells[3, 1] = "Должность: ";

workSheet.Cells[3, 2] = work;

}

else

{

workSheet.Cells[2, 1] = "Организация: ";

workSheet.Cells[2, 2] = fio;

workSheet.Cells[3, 1] = "Адрес: ";

workSheet.Cells[3, 2] = work;

}

workSheet.Cells[4, 1] = "Телефон: ";

workSheet.Cells[4, 2] = tel;

workSheet.Cells[5, 1] = "E-mail: ";

workSheet.Cells[5, 2] = email;

workSheet.Cells[6, 1] = "Период";

workSheet.Cells[6, 2] = "Практикант";

workSheet.Cells[6, 3] = "E-mail";

int rowExcel = 7;

//заполнение таблицы данными

for (int i = 0; i < dt2.Rows.Count; i++)

{

workSheet.Cells[rowExcel, "A"] = dt2.Rows[i][0].ToString();

workSheet.Cells[rowExcel, "B"] = dt2.Rows[i][1].ToString() + " " + dt2.Rows[i][2].ToString() + " " + dt2.Rows[i][3].ToString();

workSheet.Cells[rowExcel, "C"] = dt2.Rows[i][4].ToString();

++rowExcel;

}

//стиль заголовка таблицы

var cells2 = workSheet.get\_Range("A1", "E1");

cells2.Font.Bold = 3;

// cells2.RowHeight = 40;

cells2.WrapText = true;

//ширина столбцов таблицы

workSheet.get\_Range("A1", "A" + (dt2.Rows.Count + 1).ToString()).ColumnWidth = 40;

workSheet.get\_Range("B1", "B" + (dt2.Rows.Count + 1).ToString()).ColumnWidth = 50;

workSheet.get\_Range("C1", "C" + (dt2.Rows.Count + 1).ToString()).ColumnWidth = 50;

//цветные ячейки в таблице

workSheet.get\_Range("A" + (dt2.Rows.Count + 7).ToString(), "C" + (dt2.Rows.Count + 7).ToString()).Interior.Color = Color.LightGray;

workSheet.get\_Range("A6", "C6").Interior.Color = Color.LightGray;

//устанавливаем жирный шрифт

workSheet.get\_Range("A1", "A5").Font.Bold = 3;

workSheet.get\_Range("A6", "C6").Font.Bold = 3;

workSheet.get\_Range("A" + (dt2.Rows.Count + 7).ToString(), "C" + (dt2.Rows.Count + 7).ToString()).Font.Bold = 3;

//задаем границы таблицы

var cells = workSheet.get\_Range("A6", "C" + (dt2.Rows.Count + 7).ToString());

cells.Borders[Excel.XlBordersIndex.xlInsideVertical].LineStyle = Excel.XlLineStyle.xlContinuous; // внутренние вертикальные

cells.Borders[Excel.XlBordersIndex.xlInsideHorizontal].LineStyle = Excel.XlLineStyle.xlContinuous; // внутренние горизонтальные

cells.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle = Excel.XlLineStyle.xlContinuous; // верхняя внешняя

cells.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle = Excel.XlLineStyle.xlContinuous; // правая внешняя

cells.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlContinuous; // левая внешняя

cells.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = Excel.XlLineStyle.xlContinuous;

//установка выравнивания текста в ячейках

cells.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

cells.VerticalAlignment = Excel.XlHAlign.xlHAlignCenter;

workSheet.get\_Range("A1", "C1").HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

workSheet.get\_Range("A4", "B4").HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;

DialogResult res = MessageBox.Show("Экспорт завершен. При нажатии 'Да' отчет будет открыт, при нажатии 'Нет' будет предложено сохранить файл.", "Создание отчета в Excel", MessageBoxButtons.YesNoCancel);

if (res == DialogResult.Yes)

{ exApp.Visible = true; }

if (res == DialogResult.No)

{

string fileName = String.Empty;

SaveFileDialog saveFileDialog1 = new SaveFileDialog();

saveFileDialog1.Filter = "xls files (\*.xlsx)|\*.xlsx";

saveFileDialog1.FilterIndex = 1;

saveFileDialog1.RestoreDirectory = true;

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

fileName = saveFileDialog1.FileName;

}

else

return;

//сохраняем Workbook

Wb.SaveAs(fileName, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

saveFileDialog1.Dispose();

exApp.Quit();

}

if (res == DialogResult.Cancel)

{

MessageBox.Show("Экспорт данных отменен.");

Wb.Close(false, \_missingObj, \_missingObj);

exApp.Quit();

System.Runtime.InteropServices.Marshal.ReleaseComObject(exApp);

exApp = null;

Wb = null;

workSheet = null;

System.GC.Collect();

}

}

else

{

MessageBox.Show("Втаблиценетзаписей!");

}

}

}

private Word.\_Document GetDoc(string path)

{

Word.\_Document oDoc = oWord.Documents.Add(path);

SetTemplate(oDoc);

return oDoc;

}

private void SetTemplate(Word.\_Document oDoc)

{

if (flag == 0)

{

oDoc.Bookmarks["fio"].Range.Text = "Руководитель практики: " + fio;

oDoc.Bookmarks["work"].Range.Text = "Должность: " + work;

}

else

{

oDoc.Bookmarks["fio"].Range.Text = "Организация: " + fio;

oDoc.Bookmarks["work"].Range.Text = "Адрес: " + work;

}

oDoc.Bookmarks["tel"].Range.Text = "Телефон: " + tel;

oDoc.Bookmarks["email"].Range.Text = "E-mail: " + email;

}

}

}

**ПРИЛОЖЕНИЕБ**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | |
|  | АВТОМАТИЗИРОВАННАЯ СИСТеМА КОНТРОЛЯ ПРОХОЖДЕНИЯ ПРАКТИКИ СТУДЕНТАМИ В КОМПАНИИ «JAZZTEAM» | | | | | | |
|  | Ф.И.О. | Подпись | Дата | Титульный лист | Лист | Листов |
| Дипломник | *Суряпина А.В.* |  |  | 1 | 21 |
| Руководитель | *Шах А.В.* |  |  | ДП 2ИСТ31-ДО-2017 | БарГУ, ИСТ,  г. Барановичи | |
| Нормоконтроль | *Бовкунович М.В.* |  |  |
| Зав. кафедрой | *Шах А.В.* |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
|  |  | ФИО | | Подпись | | Дата | | Цель и задачи дипломного проекта | Лист | Листов |
| Дипломник | *Суряпина А.В.* | |  | |  | |
| Руководитель | *Шах А.В.* | |  | |  | | 2 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Требования к ПП | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 3 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Прикладные программы | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 4 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Концептуальная диаграмма данных | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 5 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Физическая модель данных | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 6 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Диаграмма вариантов использования | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 7 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Диаграмма классов | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 8 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Диаграмма классов (продолжение) | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 9 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Запуск программы | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 10 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Главное меню программы | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 11 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Работа с формой «Руководители» | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 12 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Работа с формой «Организации» | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 13 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Работа с формой «Практиканты» | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 14 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Работа с формой «Авторизация» | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 15 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Работа с формой «Дневник практиканта» | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 16 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Работа с формой «Задания» | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 17 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Работа с формой «Выполнение заданий» | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 18 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Работа с формой «Практика» | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 19 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Эффективность использования ПП | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 20 | 21 |
|  | | | | | | | | | | |
|  |  | | ФИО | | Подпись | | Дата | Заключение | Лист | Листов |
| Дипломник | | *Суряпина А.В.* | |  | |  |
| Руководитель | | *Шах А.В.* | |  | |  | 21 | 21 |