

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
имени ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT – инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой PhD, доцент Досжанова А.А
_____ « ____ » _____ 2020 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка автоматизированной системы расчета и калькуляции
стройматериалов

Специальность 5B070400 – Вычислительная техника и программное
обеспечение

Выполнил: Байдәулет О.Қ. Группа ВТ-16-2

Научный руководитель: доцент к.т.н. Балгабаева Л.Ш.

Консультанты:

по экономической части: к.э.н., профессор Габелашвили К.Р
(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

по безопасности жизнедеятельности: ассистент Приходько Н.Г
(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

по программному обеспечению: ст.преп. Майкотов М.Н
(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

Нормоконтролер: ст.преп. Абсатарова Б.Р
(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

Рецензент: _____
(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
имени ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В070400 - «Вычислительная техника и программное обеспечение»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Байдәулет Олжасу Қайратұлы

Тема проекта: Разработка автоматизированной системы расчета и калькуляции стройматериалов

Утверждена приказом по университету № ___ от «___» _____ 2020 г.

Срок сдачи законченного проекта «___» _____ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Руководство разработчика баз данных Delphi, данные преддипломной практики.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) анализ и исследование предметной области;
- б) проектирование приложения;
- в) реализация приложения;
- г) экономическое обоснование эффективности;
- д) вопросы безопасности жизнедеятельности и охраны труда.

Перечень графического материала (с точным указанием обязательных чертежей): имеется 14 таблиц и 32 иллюстраций.

Основная рекомендуемая литература:

1 Климова, Л. М. Delphi 7. Основы программирования. Решение типовых задач. Самоучитель / Л.М. Климова. – М: 2017. – 480 с.

2 Бекаревич, Ю.Б. Создание реляционной базы данных и запросов. MS Access 2013: учебное пособие / Ю.Б. Бекаревич, Н.В. Пушкина. – Санкт-Петербург: СПбГУЭФ, 2012. – 87 с.

3 Боканова Методические указания по выполнению экономической части дипломных работ Алматы, АУЭС, 2020 – 35с.

4 Санатова Т.С. Безопасность жизнедеятельности. – Учебное пособие для всех специальностей, 2017. – 407 с.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант Г	Сроки	Подпись
Экономическая часть	Габелашвили К.Р	21.04.2020	
Безопасности жизнедеятельности	Тыщенко Е.М	30.04.2020	
Программная часть	Майкотов М.Н	14.05.2020	
Нормконтролер	Абсатарова Б.Р	13.05.2020 – 18.05.2020	

ГРАФИК

подготовки дипломной работы (проекта)

Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Анализ и исследование предметной области	17.02.2020 – 26.02.2020	
Проектирование приложения	27.02.2020 – 23.03.2020	
Программная реализация	24.03.2020 – 30.04.2020	

Дата выдачи задания « ____ » _____ 2020 г.

Заведующий кафедрой _____ А.А.Досжанова
(подпись)

Научный руководитель проекта _____ Л.Ш.Балгабаева
(подпись)

Задание принял к исполнению студент _____ О.Қ.Байдәулет
(подпись)

АҢДАТПА

Дипломдық жобаның тақырыбы: «Құрылыс материалдарын есептеу және санаудың автоматтандырылған жүйесін құру».

Дипломдық жоба құрылыс материалдарын есептеу және есептеу үшін автоматтандырылған ақпараттық жүйені жасауға арналған. Өзірленген бағдарлама құрылыс материалдарының есебі мен есебін жақсартады. Бағдарлама Delphi интеграцияланған даму ортасында әртүрлі кітапханаларды қолдана отырып, мәліметтер базасының бағдарламалау тілінде жүзеге асырылды.

Бұдан басқа, қорытынды жұмыста қолданыстағы бағдарламаларға талдау жасалады, сондай-ақ өмір сүру қауіпсіздігі және жүйені өндіріске енгізуден экономикалық тиімділік мәселесі қарастырылады.

АННОТАЦИЯ

Тема дипломного проекта: «Разработка автоматизированной системы расчета и калькуляции стройматериалов».

Дипломный проект посвящен разработке информационной автоматизированной системы для расчета и калькуляции стройматериалов. Разработанная программа улучшает расчет и калькуляции стройматериалов. Программа была реализована на языке программирования базы данных в интегрированной среде разработки Delphi с использованием различных библиотек.

Кроме того, в выпускной работе проводится анализ существующих программ, а также рассматривается вопрос безопасности жизнедеятельности и экономической эффективности от внедрения системы в производство.

ANNOTATION

The theme of the graduation project: «Development of an automated system for building materials calculating».

The graduation project is dedicated to the development of an automated information system for the calculation and calculation of building materials. The developed program improves the calculation and calculation of building materials. The program was implemented in a database programming language in an integrated development environment Delphi using various libraries.

In addition, in the final work, an analysis of existing programs is carried out, as well as considers the issues of life safety and economic efficiency from introducing the system into production.

Содержание

Введение	9
1 Аналитическая часть	10
1.1 Общая характеристика предмета исследования	10
1.2 Исследование предметной области	11
1.2.1 Деловые процессы планово-экономического отдела	11
1.2.2 Деловые процессы договорного отдела	12
1.2.3 Деловые процессы сметного отдела	13
1.2.4 Деловые процессы производственно технического отдела	14
1.2.5 Деловые процессы бухгалтерии	15
1.3 Особенности выбора среды программирования и СУБД	15
1.4 Обзор существующих аналогичных программных продуктов	23
1.5 Постановка цели и задачи	29
2 Проектная часть	30
2.1 Информационная модель	30
2.2.1 Функциональная модель системы	30
2.2.2 Инфологическая модель системы	36
2.3 Алгоритмы расчётов решаемых задач	41
2.4 Описание структуры и функциональных алгоритмов программы	44
3 Экспериментальная часть	47
3.1 Инструкция пользователя	47
3.2 Пример расчёта программы	55
4 Экономический расчет	57
4.1 Расчет трудоемкости разработки ПО	58
4.2 Расчет затрат на разработку ПО	58
4.3 Расчет затрат на электроэнергию	60
4.4 Затраты на оплату труда	61
4.5 Расчет затрат по социальному налогу	62
4.6 Амортизация основных фондов и прочие затраты	63
4.7 Определение возможной (договорной) цены ПО	65
4.8 Расчет эксплуатационных затрат	66
4.9 Сравнительный анализ эксплуатационных затрат до и после внедрения	69
4.10 Расчет основных показателей экономической эффективности	71
4.11 Вывод по экономической части	72
5 Безопасность жизнедеятельности	74
5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал.	74
5.2 Параметры микроклимата	75
5.3 Эргономические требования к рабочему месту	77
5.4 Расчет естественного освещения	78
5.5 Расчет искусственного освещения	80
5.6 Расчет системы кондиционирования помещения	82

Заключение	85
Список литературы.....	86
Приложение А	88
Приложения Б. АКТ Внедрения.....	111

Введение

В сегодняшнее время в мире автоматизированные системы расчета достигли высокого уровня развития. В связи с этим большинство развивающихся компаний используют автоматизированные средства, позволяющие эффективно хранить, обрабатывать и рассчитывать накопленные данные. Исходя из современных требований, предъявляемых к качеству работы финансового звена крупного предприятия, нельзя не отметить, что эффективная работа полностью зависит от уровня оснащения компании автоматизированными средствами на базе компьютерных систем.

Программное обеспечение для работы с базами данных используется на персональных компьютерах уже достаточно давно. К сожалению, эти программы либо были элементарными руководителями хранения данных и не имели путей разработки приложений, либо были настолько сложны и трудны, что даже неплохо разбирающиеся в компьютерах люди избегали работать с ними до тех пор, пока не получали полных, ориентированных на пользователя приложений.

В условиях жесткой конкуренции одним из способов противодействия неблагоприятным условиям является построение эффективной автоматизированной системы, поэтому тема дипломного проекта является актуальной.

Основное преимущество автоматизации - это сокращение избытка хранимых данных, а следовательно, экономия объема используемой памяти, уменьшение затрат на множественные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте, увеличение степени достоверности информации и увеличение скорости обработки информации; излишнее количество внутренних промежуточных документов, различных журналов, папок, заявок и т.д., повторное внедрение одной и той же информации в разного рода промежуточные документы.

В связи с постоянно растущими объемами строительства возникла необходимость в автоматизации работы сметного отдела, так как сотрудники, выполняющие типовые операции не могли уже быстро и качественно справляться со своими задачами при нарастающем темпе строительства.

Целью данной дипломной работы является разработка автоматизированной системы определения стоимости проектов. Основной сферой деятельности предприятия является оказание строительных услуг сторонним организациям и частным лицам. Клиентами фирмы являются как крупные компании, так и физические лица.

В итоге проведенных исследований было принято решение разработать собственную автоматизированную систему учета определения стоимости проектов.

1 Аналитическая часть

1.1 Общая характеристика предмета исследования

Строительство – одна из важнейших и крупных отраслей народного хозяйства. Продуктом функционирования строительной отрасли является возведение гражданских, промышленных, жилых и др. зданий и сооружений.

В связи с научно-техническим прогрессом процесс строительства также развивается и совершенствуется. В нынешнее время вместе с развитием рыночных отношений и возникновением конкурентной среды все больше внимания уделяется экономической эффективности производства. Внедрение новых методов строительства (таких как новые способы монтажа конструкций, повышение технического уровня, применение поточного метода введение работ и др.) позволяет значительно повысить эффективность технологии строительного производства.

Одной из задач управления является правильная организация и обеспечение слаженных действий коллективов трудящихся. С этой целью наряду с экономическими и воспитательными широко используются организационные и административно-распорядительные методы управления.

Базой организационных методов управления являются правовые законы, нормативные акты, инструкции, положения, решения министерств и ведомств. Организационная и административно-распорядительная деятельность управления представляет реализацию директивных методов руководства, централизованного воздействия на управляемый эффект. Это направление работы ставит 3 задачи — обеспечить согласованность и сопоставимость функций отдельных исполнителей и коллективов, четкую определенность прав и обязанностей каждого звена и работника, административное воздействие на виновников отклонений от нормального хода производства и административное поощрение способных, передовых работников.

По оценкам исследований, рынок строительного программного обеспечения в Казахстане сформирован. Для обеспечения конкурентоспособности строительных программных продуктов, им должны быть приданы такие свойства, которые в максимальной степени соответствуют требованиям потребителей. Для того чтобы организовать работу в данном направлении, контролировать и оценивать результаты, необходимо уметь оценивать уровень потребности в программном обеспечении субъектов строительной отрасли.

В средних и крупных строительных компаниях существует множество отделов, которые используют различное программное обеспечение для выпуска рабочей документации (например, сметный отдел использует сметное программное обеспечение, договорной отдел – программные средства по документообороту, бухгалтерия - «1С» и т.д.). Все отделы, разумеется, связаны друг с другом логическим путем, но обмен документов (электронных)

между ними не происходит вследствие различных форматов хранения. Таким образом, отсутствует интеграция одних программных продуктов с другими установленными программами и как следствие – не осуществляется электронный документооборот.

1.2 Исследование предметной области

1.2.1 Деловые процессы планово-экономического отдела

Планово-экономический отдел в любой строительной организации отвечает за анализ финансового состояния предприятия и движение денежных средств в обороте.

Плановый отдел составляет проекты перспективных, годовых и оперативных планов подрядных и субподрядных работ, доводит плановые задания до исполнителей, организует учет их выполнения и составляет отчет по работе; осуществляет экономический анализ производственно-хозяйственной деятельности и разрабатывает предложения по выполнению повышения эффективности работы строительной организации.

Надлежащее планирование поставок материалов и техники приводит к экономии оборотных средств. К сожалению, на практике план снабжения зачастую не согласуется с планом поступления средств от заказчиков, планом платежей подрядчикам и, тем более, планом платежей налогов, несмотря на то, что один-единственный не пришедший платеж от заказчика может совсем изменить картину планирования. Приходится постоянно корректировать планы, заново их сводить и согласовывать со всеми службами. Принимая во внимание, что каждый план – эта сложная система связанных расчетов, требующих многодневных трудозатрат, планирование деятельности из инструмента управления производственным процессом обычно превращается в формальную никому не нужную работу.

В итоге, финансовые службы сталкиваются с постоянной нехваткой денежных средств на расчетных счетах, или с их избытком. Материалы и оборудование закупается заранее, пока есть свободные средства, а не тогда, когда они реально нужны. А возрастающая конкуренция среди строительных компаний уже не позволяет вести конкурентную борьбу за рынок.

Основным решением вышеуказанных задач ведения плановой деятельности – разработать и внедрить автоматизированную систему управления планово-экономической деятельности предприятия.

Стоит отметить, что большая часть информации, необходимая для планово-экономического отдела, приходит из других служб. Информация о договорах с заказчиками и подрядчиками прибывает из договорного отдела, об актах выполненных работ – из производственно-технического отдела, сметы – из сметного отдела, накладные – из бухгалтерии или со склада, информация о поступлении или переводе денег от подрядчиков или

заказчиков – из бухгалтерии. Бухгалтерия и склад на многих предприятиях уже автоматизированы.

1.2.2 Деловые процессы договорного отдела

Субъектами строительной деятельности являются инвесторы, заказчики, подрядчики, пользователи объектов строительства и другие юридические или физические лица. Согласно закону, отношения между субъектами строительной деятельности осуществляются на основе договоров, заключаемых между ними в соответствии с Гражданским кодексом Республики Казахстан (УК РК).

При строительстве любого объекта недвижимости необходимы договорные отношения:

- между инвесторами и службой заказчика (договор об оказании возмездных услуг);
- между службой заказчика и проектной организацией на выполнение изыскательских работ, разработку проектно-сметной документации, организация авторского надзора за строительством (договор подряда);
- между службой заказчика и подрядчиком (генеральным подрядчиком) на строительство определенного объекта (договор строительного подряда);
- между генеральным подрядчиком (службой заказчика) и субподрядчиком на выполнение отдельных видов строительного-монтажных и пусконаладочных работ (договор субподряда);
- между заказчиком (службой заказчика, генеральным подрядчиком, субподрядчиком) и подрядчиком (производителем или посредником) на производство или поставку оборудования, инвентаря, строительных материалов и конструкций (договор купли-продажи или договор поставки).

А также, вполне допустимы договоры, имеющие опосредованное отношение к строительству, а именно договор со страховой компанией на страхование строительных рисков, договор поручительства, договор лизинга, договор аренды и т.д.

Договорной отдел анализирует, согласовывает и оформляет договора подряда и субподряда, ведет контроль их выполнения и правильности расчетов с заказчиками и субподрядчиками за выполнение строительного-монтажных работ.

Любой договор в соответствии с действующим законодательством заключается по усмотрению сторон, что позволяет каждой из сторон выбрать для каждого конкретного случая наиболее подходящие условия и меры стимулирования для надлежащего выполнения обязательств и меры ответственности за нарушение договорных обязательств.

По договору на строительство подрядчик обязан построить конкретный объект по указанию заказчика в сроки, установленные договором, или выполнить отдельные строительные, монтажные или пусконаладочные работы, а заказчик обязан создать необходимые условия подрядчику для

выполнения работ, принятия их результата и уплаты оговоренной цены. В отдельных случаях, предусмотренных договором, подрядчик может принять на себя обязательство по эксплуатации объекта после его принятия заказчиком в течение срока, указанного в договоре.

Повышения качества договорных отношений возможно за счет процесса ведения работы в программе «1С:Смета», реализованной на платформе «1С:Предприятие 8.0.», разработчиком фирмой. В результате время, затрачиваемое на формирование и мониторинг договоров, полностью исключается путем настройки параметров взаиморасчетов, кредитования, составления и сроков предоставления технической документации и также возможна корректировка цены контракта.

1.2.3 Деловые процессы сметного отдела

Сметный отдел рассматривает и утверждает сметную документацию, контролирует ее получение, а также их выполнение; разрабатывает ориентировочные цены на материалы и услуги для трудовых единиц.

Ценообразование в строительстве - сложный и многофакторный процесс. Это связано с тем, что изделия строительного комплекса отличаются необычайно высоким расходом материалов (как по объему, так и по номенклатуре), высокой трудоемкостью изготовления, длительным временем производства и привязанностью к методам строительства. Строительство многих объектов длится годами. В условиях инфляции необходимо регулярно пересчитывать расчетную стоимость с учетом коэффициентов и индексов инфляции.

Стоимость строительства индивидуальна для каждого объекта. Даже при использовании стандартных проектов стоимость строительства всегда различна и рассчитывается отдельно для каждого объекта. В связи со сложностью работы было разработано программное обеспечение для специалистов для составления, корректировки и проверки строительных смет.

Сегодня на рынке программного обеспечения в области оценки цен представлено более 20 программ различных разработчиков, сертифицированных Госстроем. Причем все эти программы находятся примерно на одном уровне. Программы предназначены для составления, расчета, хранения и печати строительных смет: проектные, исполнительные, ресурсные и сводные сметы, ведомости по списанию ресурсов, накопительные ведомости, ведомости объемов работ и т. д.

В последние годы рынок разработчиков оценочного программного обеспечения очень динамично развивается. В связи с этим оценщик должен выбирать между многочисленными оценочными программами, адаптироваться не только к новым программным проектам, но также активно изучать нормативные акты и буквы, указывающие новые методы расчета и использующие новые нормативные базы.

В условиях перехода на новую сметно-нормативную базу были выпущены новые «Методические указания по определению стоимости строительной продукции на территории Республики Казахстан». В течение всего времени, когда велась активная разработка новых стандартов, разработчики предполагаемого программного обеспечения обсуждали процедуру распространения новых стандартов в электронной форме и заботились о функциональном содержании программного обеспечения. Сегодня оценочные программы, сертифицированные в Госстрой, соответствуют необходимой полноте функциональности. Например, такие как объем нормативно-правовой базы, наличие территориальных или ведомственных стандартов, дополнения, коэффициенты пересчета к текущему уровню цен, поправочные коэффициенты, технические части и рекомендации.

В результате, пользователь начал задумываться об интеллектуальности и автоматизации программного обеспечения. Некоторые разработчики начали задумываться о комплексном подходе к автоматизации управления предприятием в единой программной среде. Так, фирма «1С» выпустила новое программное обеспечение на платформе «1С: Предприятие 8.0.» конфигурацию «1С:Смета» для автоматизации процесса составления сметной документации. Типичная конфигурация представлена основными типичными функциями и выходными документами, она открыта для изменений и может быть разработана и пополнена.

1.2.4 Деловые процессы производственно технического отдела

Производственно-технический отдел принимает от заказчиков проектно-сметную документацию, контролируя своевременное поступление, комплектность и качество, обеспечивает технологической документацией строительно-монтажные управления и субподрядные организации; разрабатывает с участием других подразделений треста планы технического развития и организационно-хозяйственных мероприятий; проводит мероприятия по применению перспективных конструкций и эффективных материалов.

При использовании программы «1С:Смета», специалисты производственно-технического отдела значительно сокращают сроки подготовки сметной и учетной документации, расчета объемов работ и ввода данных о выполненных работах. Затраты времени на формирование актов выполненных работ и экспертизу сметной документации сокращаются и оперативно получают полную информацию об объемах и стоимости выполненных работ в разрезестроек, договоров, контрагентов, исполнителей за любой период времени, а также по состоянию на текущую дату.

Также на платформе «1С: Предприятие 8.0» «1С: Строительный подрядчик 3.0». Управление строительством », которая предназначена для автоматизации производственного планирования (календарного

планирования), ведения производственного учета и анализа хода строительно-монтажных работ.

В результате использования этого программного продукта пользователь исключает ручной труд из практики при составлении календарных планов и планировании потребностей в ресурсах. Автоматизация учета и анализа выполнения работ по календарному плану и автоматическая обработка значительных объемов учетной информации с большим количеством работ и объектов строительства могут быть использованы как у классических подрядчиков, так и в работе других участников инвестиционно-строительного процесса.

1.2.5 Деловые процессы бухгалтерии

Трудно найти современное предприятие, которое не автоматизировало бы свой учет. Но, вложив средства в автоматизацию бухгалтерского учета, топ-менеджеры не получали финансовой прибыли, только быструю и качественную отчетность. Позднее возникло желание использовать и анализировать накопленные учетные данные для целей управления предприятием. Некоторые менеджеры даже автоматизировали получение небольшой части управленческой отчетности, которую можно было бы получить из систем бухгалтерского учета. Но руководитель должен сравнить фактические данные с запланированными; вам нужно уметь оставлять сложные многоступенчатые планы, играть в сценарии. В результате вводятся фактические данные для работы с бухгалтерией.

Основные функции бухгалтерии:

- планирование финансовых потоков, сравнение с фактическими данными, сравнительный анализ объектов, динамика показателей по объекту, отчеты об оплате за работу;
- получение точной и своевременной информации о наличии и использовании материалов и денежных средств;
- получение информации о расчетах с подрядчиками, поставщиками, покупателями и покупателями услуг.

1.3 Особенности выбора среды программирования и СУБД

В качестве среды разработки пользовательского приложения была выбрана среда визуального проектирования и разработки приложений Borland Delphi 7. В основе этого выбора лежат следующие соображения.

Общепризнанным является тот факт, что Delphi является универсальным средством для решения самых различных проблем программирования. Причем нельзя не отметить, что эта система сделала колоссальный шаг вперед по сравнению с остальными средствами визуального проектирования приложений. Delphi была первой системой ускоренной разработки приложений (RAD — Rapid Application Development),

в которой удачно соединились средства визуального проектирования и оптимизирующий компилятор. Плюс к этому обширная библиотека компонентов, полная поддержка Internet, многоуровневая архитектура баз данных. Вот почему многие считают Delphi самой мощной системой разработки среди всех известных систем аналогичного назначения для Windows. Если Delphi и не является панацеей, то она очень сильно приближается к тому, чтобы именоваться таким образом.

На сегодняшний день на рынке имеется немало хороших средств разработки, позволяющих создавать приложения для архитектуры клиент/сервер. PowerBuilder, Visual Basic и другие подобные системы охватывают достаточно большие секторы рынка средств разработки баз данных. Однако, Delphi имеет ряд преимуществ.

Как уже говорилось выше, в Delphi удачно сочетаются средства визуального проектирования приложений и оптимизирующий компилятор, чего, к сожалению, нельзя сказать о других системах RAD. Наличие в системе компилятора или генерирование ею выполняемого машинного кода еще не означает, что получаемый код является оптимальным. Такие системы, как PowerBuilder и Visual Basic, изначально создавались на основе концепции генерации псевдокода. При выполнении приложений, созданных с помощью этих систем, полученный псевдокод интерпретировался. Если сравнивать вышеупомянутые языки с языком Object Pascal, используемом в системе Delphi, то различие видно сразу. Object Pascal всегда был компилируемым языком, и при его разработке были соблюдены все требования, выполнение которых обязательно при компиляции и оптимизации. В итоге, Delphi является единственным полноценным средством промышленной разработки систем клиент/сервер. Сравнение Delphi с Visual Basic или PowerBuilder подобно сравнению современного компилятора языка C++ с компилятором Clipper времен господства DOS. Только успех Delphi подвиг многих производителей средств разработки приложений клиент/сервер добавить в свои продукты технологию генерации машинного кода. Однако, пока они только начинают двигаться в этом направлении, развитие системы Delphi идет семимильными шагами.

В случае Delphi слова «оптимизирующий компилятор» не означают «медленный компилятор». В последней версии продукта представлен лучший компилятор языка Pascal компании Borland, которая уже на протяжении многих лет удерживает пальму первенства в этой области. Компиляторы Pascal этой компании снискали заслуженную славу за генерацию выполняемого кода, который экономно использует ресурсы компьютера и одновременно обладает высокой производительностью. Компилятор Object Pascal, используемый в Delphi, не является исключением. Более того, генератор кода Delphi — это тот самый генератор кода, который применяется компанией Borland в ее компиляторах с языка C++. Таким образом, используя Delphi, можно достичь скорости программ, написанных на языке C++, избежав при этом сложностей этого языка программирования.

Помимо эффективности выполняемого кода, каждому разработчику необходимо средство, которое было бы, с одной стороны, достаточно мощным и гибким, чтобы выполнить любую стоящую перед ними задачу, а, с другой стороны — было достаточно простым и удобным в работе. Такая система должна быть построена на принципах объектно-ориентированного программирования, и вместе с тем, должна позволять применять, в случае необходимости, ассемблер. Она должна генерировать не только исполняемые файлы в формате EXE, но и создавать файлы DLL и драйверов устройств. Наконец, она совмещать в себе функции среды для быстрой разработки баз данных, которая не вынуждала бы их каждый раз при программировании спускаться до ядра СУБД.

Все это (и даже больше) предоставляет Delphi. Это стало возможным благодаря подходу, который применила компания Borland при создании этой системы, - собрать все лучшее, что есть в средствах разработки для Windows, и объединить в одном продукте. В состав Delphi входит обширная библиотека компонентов (не говоря еще об огромном количестве библиотек компонент, разработанных третьими фирмами и отдельными разработчиками), с помощью которой можно избежать ручного написания программ, что широко распространено в других средствах разработки. Они достаточно широко представлены в Internet. С другой стороны, имеется возможность в любой момент прибегнуть к низкоуровневым ассемблерным процедурам. Можно создавать приложения в визуальном режиме, просто помещая нужные компоненты на форму и, вместе с тем, сохраняя доступ ко всем функциям программного интерфейса Windows, системным сообщениям и процессам. Наконец, несмотря на то, что Delphi является системой разработки, ориентированной на создание приложений для работы с базами данных, с ее помощью можно разрабатывать любые приложения Windows, начиная от редакторов и заканчивая хранителями экранов. Даже сама система Delphi разрабатывается с использованием системы Delphi[6].

Добавив ко всему этому мощные средства отладки становится понятно, почему Delphi является системой, равных которой на сегодняшний день практически нет.

Выбор Delphi в качестве среды разработки пользовательских приложений основывался еще и на таком простом факте, что программирование на Delphi включено в программу подготовки IT-специалистов в большинстве ВУЗов. А значит, имеется гарантированная возможность коллективной разработки системы группой взаимно понимающих друг друга профессионалов, а также преемственность в случае, если коллектив разработчиков меняется.

Таким образом, Delphi можно выделить из ряда других средств разработки благодаря наличию:

- обширной библиотеки классов;
- быстрого оптимизирующего компилятора, генерирующего машинный код;

- встроенного отладчика, равных которому нет;
- простого в освоении механизма доступа к базам данных;
- мощной и удобной в работе среды разработки.

Другие средства в каждой из перечисленных областей остались далеко позади. В отличие от других средств RAD, система Delphi полнофункциональна – в распоряжении программиста имеются разнообразные возможности наращивания ее мощи, он практически никогда не столкнется с задачей, не имеющей решения в среде Delphi.

Вот лишь краткий список возможностей Delphi, которые делают ее такой гибкой:

- прямой доступ к программному интерфейсу Windows;
- встроенный ассемблер и поддержка программирования в машинных кодах;
- возможность создания пользовательских компонентов VCL и ActiveX;
- поддержка формата DLL и других выполняемых файлов Windows;
- возможность многоуровневой разработки приложений;
- полная объектная ориентированность — в программах можно создавать объекты, берущие начало как от библиотечных классов, так и от созданных программистом.

Нужно отметить, что далеко не все инструментальные средства, обладающие поддержкой объектно-ориентированного программирования (ООП), имеют такие возможности. Многие из систем разработки лишь имитируют объектную ориентацию, но сами по своей природе не являются продуктами, построенными на принципах ООП. С другой стороны, если какой-то продукт действительно реализован на принципах ООП, но при этом чудовищно медлителен, то его вряд ли можно считать полноценным инструментальным средством.

Система разработки, претендующая на название истинно объектно-ориентированной системы, должна соответствовать четырем основным критериям: Наследование (inheritance), Полиморфизм (polymorphism), Инкапсуляция (encapsulation), ООП — Преобладающая методология. Последний — самый важный принцип. Истинно объектно-ориентированное средство разработки должно иметь ООП в качестве преобладающей методологии создания программного кода, а не в качестве дополнительной возможности, добавленной задним числом. Когда ООП является преобладающей методологией, работа средства разработки должна быть высокопродуктивной. Если же производительность системы оставляет желать лучшего, возникает подозрение относительно принципов, положенных в ее основу.

Delphi полностью соответствует этим критериям, как и все традиционные средства ООП, такие как системы программирования на C++. Достаточно вспомнить тот малоизвестный факт, что компилятор с языка Pascal компании Borland был полностью объектно-ориентированным задолго до появления объектно-ориентированных компиляторов с языка C как

компания Microsoft, так и компания Borland. Технология ООП в Delphi — это не «довесок», а краеугольный камень, на котором зиждется вся среда разработки. Добавив средства визуального проектирования приложений, компания Borland лишь избавила пользователей от утомительной работы, которая часто ассоциируется с ООП.

Одной из важнейших характеристик средства разработки систем клиент/сервер является его масштабируемость (scalability), т.е. возможность работы как с простейшими, так и сложными базами данных. Многоуровневая архитектура приложений для работы с базами данных, созданных на Delphi делает данную систему масштабируемой в такой степени, что с ней не может потягаться никакое другое средство из имеющихся на рынке. Среди возможностей, благодаря которым обеспечивается такая масштабируемость, можно выделить следующие:

- поддержка как таблиц, размещенных на локальном компьютере, так и хранящихся на удаленном сервере баз данных;
- поддержка гетерогенных запросов и доступа к различным СУБД из одного приложения;
- не зависящий от платформы доступ к базам данных с помощью использования механизма Borland Database Engine (BDE) или ADO (Microsoft ActiveX Data Objects), позволяющего приложениям переключаться с одной СУБД на другую;
- быстрые и эффективные механизмы работы с БД для всех основных платформ клиент/сервер;
- компоненты DataSet с виртуальными методами, позволяющие разработчикам создавать собственные драйверы баз данных, независящие от встроенных;
- поддержка компактных и не требующих настройки клиентских приложений;
- полная поддержка ODBC.

В этом плане Delphi среди средств разработки систем клиент/сервер является универсальной. При использовании самых разнообразных таблиц, хранящихся на рабочей станции или расположенных на SQL-сервере, Delphi предоставляет программисту все необходимые возможности для создания базы данных, причем кратчайшим путем.

И в заключении хочется отметить. Выбор Delphi из всего многообразия средств визуального проектирования приложений основывается на следующих, выгодно отличающих ее от других систем свойствах:

- обширная библиотека объектов;
- истинная поддержка принципов ООП;
- компиляция в машинный код;
- встроенные средства отладки;
- абстракция доступа к базам данных;
- полный набор средств для разработки компонентов VCL и ActiveX;
- прямой доступ к программному интерфейсу Windows;

- встроенная поддержка ассемблера;
- создание выполняемых файлов в формате DLL и других системных форматах Windows;
- развитая среда разработки с исчерпывающим набором средств.

Delphi гораздо более эффективна в качестве инструментальной среды разработки систем клиент/сервер, по сравнению ее с такими средствами RAD первого поколения, как PowerBuilder и Visual Basic. Даже при сравнении со средствами второго поколения, такими как Optima++, Delphi выгодно отличается мощностью и простотой использования. И, повторим, она вошла в «базовый образовательный набор», т.е. знание Delphi сегодня является не достоинством, а обязанностью.

Выбор этого программного продукта обусловлен также следующим:

а) в Delphi имеется возможность визуального конструирования форм, что избавляет при создании программы от многих аспектов разработки интерфейса программы, так как Delphi автоматически готовит необходимые программные заготовки и соответствующий файл ресурсов;

б) библиотека визуальных компонентов предоставляет огромное разнообразие созданных разработчиками Delphi программных заготовок, которые после несложной настройки готовы к работе в рамках создаваемого приложения. Использование компонентов не только во много раз уменьшает время создания программы, но и существенно снижает вероятность случайных программных ошибок;

в) мощность и гибкость языка программирования Object Pascal – достоинство Delphi, выгодно отличающее эту среду от других инструментов RAD. От языка Visual Basic язык Pascal отличает строгая типизированность, позволяющая ещё на этапе компиляции обнаружить многие ошибки;

г) среди прочих преимуществ Delphi можно выделить самую эффективную работу с базами данных. Это обуславливается тремя обстоятельствами: высоко производительной машиной для доступа к данным разного формата (ADO, ODBC), наличием многочисленных компонентов, ориентированных на работу в этой сфере;

д) Доступ программы к базе данных реализуется с помощью технологии ADO (Microsoft ActiveX Data Objects), которая обеспечивает универсальный доступ к источникам данных из приложений БД. Технология ADO завоевала популярность у разработчиков, благодаря уникальности – базовый набор интерфейсов OLE DB имеется в каждой операционной системе Microsoft.

Несмотря на то, что технология ADO достаточно стара, и существуют более совершенные методы доступа к базам данных, выбор технологии ADO обусловлен тем, что она проста в реализации и не требует больших вычислительных ресурсов. Кроме того, разрабатываемая автоматизированная система не предполагает больших объемов вычислений, что также учитывалось при выборе технологии доступа к базе данных.

Таким образом, можно сделать вывод о том, что Delphi – один из самых мощных инструментов разработки программных продуктов любой сложности и направленности, – наибольшим образом подходит для разработки данной системы.

Для создания базы данных, разрабатываемого программного средства, было решено использовать СУБД “Microsoft Access 2013”.

Как реляционная СУБД “Microsoft Access 2013” обеспечивает доступ ко всем типам данных и позволяет одновременно использовать несколько таблиц базы данных. “Microsoft Access 2013” специально спроектирован для создания многопользовательских приложений, где файлы базы данных являются разделяемыми ресурсами в сети. В “Microsoft Access 2013” реализована надёжная система защиты от несанкционированного доступа к файлам. На основе проектов “Microsoft Access 2013” можно создавать клиентские приложения, которые подключаются к базам данных “MS SQL Server” версии 7.0 или выше. При этом проект “Microsoft Access 2013” содержит только вторичные объекты (формы, отчеты, модули и т.д.). Таблицы, а также специфические аналоги запросов хранятся на стороне сервера.

Основным преимуществом “Microsoft Access 2013” по сравнению с такими СУБД, как “Microsoft SQL Server” и “Oracle” является гораздо меньший объем файла базы данных и меньшие системные требования к установке и работе СУБД. Для разрабатываемой базы данных такие преимущества являются определяющими, именно поэтому для разработки серверной части приложения и был выбран “Microsoft Access 2013”.

Существует мнение о том, что Access – наиболее простая в освоении и использовании СУБД. Отчасти, это – результат маркетинговых усилий компании Microsoft. Те, кто начинал работу с такой СУБД, как dBase III (IV или V), скорее всего с этим не согласятся. Но, тем не менее, для новичка эта система имеет довольно дружественные средства, которые дают возможность без глубоких знаний теории баз данных создавать несложные объекты для хранения и простой обработки данных. С другой стороны, специалист по Access может использовать эту систему для работы в сети и доступа к более эффективным системам баз данных, подобным SQL Server, что я и попытаюсь реализовать в своём приложении.

Access – это набор инструментов конечного пользователя для управления базами данных. В её состав входят конструкторы таблиц, форм, запросов, отчётов и страниц доступа к данным. Эту систему можно рассматривать и как среду разработки приложений баз данных. Используя макросы или модули для автоматизации решения задач, можно создавать ориентированные на пользователя приложения такими же мощными, как и приложения, написанные на непосредственно языках программирования. При этом они будут включать кнопки, меню и диалоговые окна. Мощностъ и практичность Access делают эту систему лучшей СУБД из представленных сегодня на рынке.

СУБД Access одна из самых мощных, гибких и простых в использовании СУБД, ориентированная в первую очередь на пользователей Microsoft Office, в том числе и не знакомых с программированием.

MS Access является СУБД реляционного типа, в которой разумно сбалансированы все средства и возможности, типичных для современных СУБД. Реляционная база позволяет проводить всесторонний анализ содержащейся в ней информации, упрощает поиск, поддержку и защиту данных, поскольку таблицы, индексы, правила ссылочной целостности, бизнес-правила, список пользователей, а также формы и отчеты хранятся в одном файле.

В состав Access 2013 входят:

- средства манипуляции данными Access и данными, доступными через ODBC (последние могут быть "присоединены" к базе данных Access);
- средства создания форм, отчетов и приложений; при этом отчеты могут быть экспортированы в формат Microsoft Word или Microsoft Excel;
- средства публикации отчетов в Internet;
- средства создания интерактивных Web-приложений для работы с данными (Data Access Pages);
- средства доступа к данным серверных СУБД через OLE DB;
- средства создания клиентских приложений для Microsoft SQL Server;
- средства администрирования Microsoft SQL Server.

СУБД Access для работы с данными использует процессор баз данных Microsoft Jet 3.0, объекты доступа к данным и средства быстрого построения интерфейса – Конструктор форм. Для получения распечаток используются Конструкторы отчетов. Автоматизация рутинных операций может быть выполнена с помощью макрокоманд. На тот случай, когда не хватает функциональности визуальных средств пользователи Access могут обратиться к созданию процедур и функций. При этом как в макрокомандах можно использовать вызовы функций, так и из кода процедур и функций можно выполнять макрокоманды.

Несмотря на свою ориентированность на конечного пользователя в Access присутствует язык программирования Visual Basic for Application, который позволяет создавать массивы, свои типы данных, вызывать DLL-функции, с помощью OLE Automation контролировать работу приложений, которые могут функционировать как OLE -серверы. Можно целиком создавать базы данных с помощью кодирования, когда в этом появляется необходимость.

MS Access из всех рассматриваемых средств разработки имеет, пожалуй, самый богатый набор визуальных средств. Тем не менее кодировать в Access приходится. Так как ни одно приложение, не предназначенное для себя лично, создать хотя бы без одной строчки кода невозможно. Для коммерческого распространения приложений, разработанных на Access, как мы уже писали, предназначен пакет Access

Developer Toolkit, вместе с которым поставляются и несколько дополнительных объектов ActiveX.

Главное качество Access, которое привлекает к нему многих пользователей – тесная интеграция с Microsoft Office. К примеру, скопировав в буфер графический образ таблицы, открыв Microsoft Word и применив вставку из буфера, мы тут же получим в документе готовую таблицу с данными из БД.

СУБД “Microsoft Access 2013” удовлетворяет всем требованиям надежности, понятности пользователю, защиты информации, модифицируемости, масштабируемости, минимизации затрат на сопровождение и поддержку, эргономичности, описанным выше.

1.4 Обзор существующих а налогичных программных продуктов

Программное обеспечение, используемое строительными компаниями при подготовке отчетов, достаточно широкое и разнообразное.

Все программные продукты можно разделить на несколько групп – как по назначению, так и по цене. По назначению в учетном процессе программы можно разделить на:

- универсальные электронные табличные процессоры;
- электронные базы данных;
- специализированные бухгалтерские программы;
- системы планирования и бюджетирования;
- системы логистики;
- системы разработки смет;
- нормативно-справочные и поисковые системы, включая базы данных по СНиПам (строительные нормы и правила) и СанПиНам (Санитарные правила и нормы);
- системы оценки и управления рисками;
- генераторы отчетных форм;
- системы электронного документооборота;

Ценовой диапазон программ каждой группы очень большой: от нескольких долларов до десятков сотен тысяч долларов. Рассмотрим их.

В первой группе присутствуют универсальные, электронные, табличные процессоры. Самыми простыми в использовании, дешевыми и наиболее распространенными программными продуктами, используемыми в отчетности, являются табличные процессоры. К ним относятся Excel (широко известный, благодаря тотальному применению в Казахстане продуктов Microsoft), Lotus или Quattro Pro.

Безусловное достоинство этих программ – простота в использовании.

Недостаток — невысокая надежность результатов.

Как показало специальное исследование, проведенное крупнейшей аудиторской фирмой в 2016 году, примерно 80% всех ошибок, обнаруженных в финансовой отчетности компаний, связаны с

использованием электронных таблиц при составлении финансовой отчетности. Действительно, достаточно внести изменения в одну из взаимосвязанных таблиц, введя новые строки или столбцы, так как все отношения между таблицами могут быть нарушены и, как следствие, в окончательных результатах появляются ошибки.

Эту коварную особенность обработчиков таблиц следует помнить в том случае, если финансовая и управленческая отчетность составляется с их помощью.

Достаточно плохо защищены таблицы от:

- циклических ошибок;
- ошибок, связанных с меж файловыми связями;
- несанкционированного доступа к информации и т. д.

Одним из решений этой проблемы является обязательное использование контрольных точек в таблицах (ячейки, которые содержат некоторые промежуточные вычисления). Например, любой бухгалтер знает уравнение: Активы = Обязательства + Капитал. Поэтому при подготовке отчетов для проверки правильности отчета достаточно выбрать ячейку, в которой будут суммироваться все активы, и из этой суммы будут вычтены обязательства и капитал. Эта ячейка для верно составленного балансового отчета всегда равна нулю, поскольку активы – обязательства – капитал = 0. Ну, а ненулевая сумма будет являться признаком ошибки.

Во второй группе можно отнести электронные базы данных. Гораздо более надежными продуктами, используемыми в отчетности, являются электронные базы данных, такие как MS Access. Однако, в отличие от настольных процессоров, эти программы уже требуют бухгалтера или экономиста с минимальными знаниями в области программирования.

Для представителей третьей группе характерны специализированные бухгалтерские программы. Как показывает опыт, бухгалтеру проще использовать специализированное программное обеспечение, которое уже настроено специалистами, для ведения учета и подготовки отчетов.

Такие программы выпускаются как небольшими отечественными компаниями, так и трансконтинентальными холдингами в довольно широком диапазоне расходов (от сотен долларов до нескольких миллионов).

Среди казахстанских производителей программного обеспечения, позволяющего формировать отчетность по нескольким стандартам, можно отметить:

- 1С: Бухгалтерия 8 для Казахстана;
- 1С: Предприятие 8 комплексная поставка для Казахстана;
- К2.Бюджет.

Из западных программ, представленных на международном рынке, стоит отметить:

- Scala;
- Sun Systems;

- Microsoft Navision;
- Microsoft Axapta;
- SAP R/3;
- PeopleSoft;
- Oracle Financials.

Также нужно помнить о том, что при выборе программы непосредственно каждый продукт имеет свои особенности и что идеальных программ не существуют. Поскольку международная отчетность содержит большое количество аналитической информации, при выборе программы необходимо обеспечить получение необходимой информации с помощью программы. Так как компания заинтересована в ведении нескольких видов бухгалтерского учета, необходимо убедиться, что программа может по-разному отражать одну и ту же операцию для разных типов отчетности.

Кроме того, наиболее важным моментом для строительного бизнеса при выборе специализированных программ является наличие высокоэффективного подразделения учета затрат. Для этих целей в учетных системах разрабатываются взаимосвязанные модули Job Estimating («Сметы») и Job Cost («Затраты»). В первом модуле вводятся все сметы по всем проектам, которые выполняются в данный момент. Во втором проводится по объектный учет затрат.

Большинство программ, предлагаемых рынком в той или иной форме, предоставляют возможность для объектно-ориентированного учета.

В качестве хорошего дополнения к вышеуказанному программному обеспечению, непосредственно связанного с вводом, хранением и обработкой учетной информации, наиболее продвинутые компании используют другие программы.

Логистические системы позволяют оптимизировать товарные потоки.

Программы оценки рисков позволяют объективно оценивать предпринимательские, кредитные, рыночные и другие виды рисков, чтобы отразить их как в управленческой, так и в финансовой отчетности.

Генераторы форм отчетности позволяют на основе всей информации, хранящейся в различных базах данных, быстро генерировать необходимые отчеты, делая их простыми и интуитивно понятными.

В конечном счете программы электронного документооборота позволяют быстро доставлять сгенерированные отчеты заинтересованным пользователям. Естественно, этот набор не ограничен кругом программ, которые участвуют в том или ином качестве в процессе учета.

Многие строительные компании, в том числе крупные застройщики ведут расчеты в Excel и пользуются не всегда качественными «самописными» программами. Некоторые предприятия пытаются внедрить узкопрофильные системы автоматизации, созданные сторонними разработчиками: отечественными или западными. Следует отметить, что не все эти системы подходят для автоматизации строительного бизнеса.

Кроме того, процесс их реализации может затянуться на месяцы и даже годы. Информация о договорах с заказчиками и подрядчиками и актах закрытия работ – от контрактного отдела, сметы – от расчетного отдела, счета-фактуры - либо со склада или из бухгалтерии, поступления денег от клиентов и выплаты подрядчикам из бухгалтерии отдел. Поэтому созданная система управления должна автоматически получать необходимые данные из бухгалтерских и складских программ. Данные о сроках и сметной стоимости выполненных работ также могут быть автоматически получены из оценочных программ. Гораздо перспективнее использовать комплексный подход в автоматизации управления предприятием.

Дело также в нежелании некоторых лидеров изменить существующий порядок вещей. Во-первых, вопросы повышения конкурентоспособности или оптимизации затрат в ситуации строительного бума практически не поднимаются. Во-вторых, многие владельцы предприятий строительной отрасли удовлетворены так называемой «лоскутной» автоматизацией: программные решения используются для автоматизации определенных областей деятельности компании (производственные процессы, бухгалтерский учет, бюджетирование, складской учет) и интеграции все операции с использованием одной системы - это даже вопрос речи. Но у такого подхода немало «минусов», в частности:

- неполная информация о текущем состоянии дел (что чревато принятием неверных решений);
- отсутствие контроля над работой сотрудников (а это влечет за собой срывы сроков по проектам);
- невозможность отследить все финансовые потоки компании (следовательно, потеря денег).

Все это рано или поздно приводит менеджера к мысли о внедрении системы управления бизнесом. Компьютеризация строительства в последние десятилетия потребовала огромных материальных затрат и усилий большого количества специалистов. Была сформирована принципиально новая компьютерная методология проектирования и управления в строительстве, основанная на многокритериальных оценках, исследованиях различных вариантов, системном подходе к принимаемым решениям и современных информационных технологиях.

Интересен опыт разработки комплексного программного обеспечения на платформе «1С». Партнерами ЗАО «1С» созданы сотни отраслевых решений, в том числе и для предприятий строительного комплекса, примеры которых были приведены выше. Все эти системы реализованы на основе одной платформы «1С:Предприятие 8.0». Платформа позволяет организовать единую информационную систему для управления различными аспектами строительного предприятия – строительством, снабжением, учетом, бюджетированием, персоналом и т. д.

В то же время в системе могут работать руководители нескольких отделов. Существует алгоритм разделения прав доступа для редактирования

и утверждения различных областей программы, с назначением личной ответственности за заполнение любого документа. Каждая новая информация распространяется среди пользователей, заинтересованных в ней, в той форме, в которой они нуждаются. Система позволяет хранить и анализировать огромное количество информации, находить различные взаимосвязи и взаимодействия и может быть успешно адаптирована к потребностям конкретного предприятия.

Большой популярностью пользуется программный продукт «1С: Предприятие» на Платформе v.7.7, такой как конфигурация «1С: Подрядчик строительства 2.3», разработанная фирмой «ИМПУЛЬС-ИВЦ». Данная конфигурация предназначена для автоматизации деятельности предприятий и организаций, выполняющих функции строительных подрядчиков. Конфигурация позволяет автоматизировать основные разделы управления строительством заказчика: планирование и учет капитальных вложений, учет источников финансирования и взаиморасчетов, оборудования и материалов, учет деятельности предприятия. Учет выполнения строительно-монтажных работ в разрезе объектов строительства, статей затрат на строительство, технологической структуры затрат подрядчиков. Формирование справки о стоимости выполненных работ по форме. Сокращение времени, затрачиваемого на обмен документами и информацией, достигается за счет обмена данными с оценочными программами «1С: Оценка».

В результате использования функционала «1С: Подрядчик строительства 2.3» возможно:

- вести учет расчетов с покупателями, поставщиками, проектными организациями в разрезе объектов;
- вести учет материалов в разрезе материально-ответственных лиц, отклонений в приобретении;
- вести учет работы транспортных средств, от выписки, от выписки первичных документов до аналитических отчетов о работе оборудования;
- вести учет основных средств по местам хранения, вне балансовый учет малоценных основных средств;
- вести учет внутрихозяйственных поселений треста с его структурными подразделениями (филиалами);
- вести учет операций по передаче заказчикам законченных объектов с оформлением соответствующих актов КС-11 и КС-14;
- вести учет получения и использования денежных средств и контролировать использование целевого финансирования;
- вести учет оборудования, требующего монтажа, и строительных материалов в разрезе строительных объектов. Учет затрат на заготовку и хранение материалов и т.д.;
- вести учет денежных документов;
- вести учет вексельных операций.

Общим недостатком вышеуказанных программ является высокая стоимость, а также сложность восприятия для конечного пользователя. Поэтому было принято решение разработать собственную программу.

1.5 Постановка цели и задачи

Задача дипломного проекта заключается в разработке автоматизированной системы для расчета стройматериалов.

Данная программа будет реализовываться на языке программирования базы данных в интегрированной среде разработки Delphi с использованием различных библиотек.

Для создания базы данных, разрабатываемого программного средства, было решено использовать СУБД “Microsoft Access 2013”.

В данной программе анализируется разработка автоматизированной системы учета определения стоимости проектов. Основной сферой деятельности предприятия является оказание строительных услуг сторонним организациям и частным лицам.

Также в программе будет иметься функция, которая будет получать данные из введенного поля, сохранять его, архивировать в буфер, и в конце диспетчеру будет показана таблица с учетом данных о расходах по объектам.

Для реализации данного проекта необходимо осуществить следующие требования:

- разработка информационно-логической модели данных;
- анализ процесса передачи данных в формате СУБД Access;
- выбор оптимальных инструментов для разработки;
- разработка формы документа и отчета документации для оперативного мониторинга;
- пользовательский интерфейс ИС, включающий экранные формы для работы со справочными, оперативными данными и отчетными данными;
- разработка запросов, обеспечивающие формирование данных для первичных и отчетных документов;
- руководство пользователя.

2 Проектная часть

2.1 Информационная модель

2.2.1 Функциональная модель системы

Технология создания информационных систем предъявляет особые требования к методам реализации и программным средствам. Реализация проектов по созданию информационных систем обычно разбивается на стадии анализа (перед созданием информационных систем необходимо разобраться и описать бизнес-логику предметной области), проектирования (необходимо определить модули и архитектура будущей системы), прямое кодирование, тестирование и сопровождение.

Суть структурного подхода к разработке информационных систем заключается в его декомпозиции (разбиении) на автоматизированные функции: система делится на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, подразделяются на задачи и т. д.

Процесс расщепления продолжается вплоть до конкретных процедур. В то же время автоматизированная система поддерживает целостное представление, в котором все составляющие компоненты взаимосвязаны. Основные этапы, на которые разбивается процесс проектирования информационной системы, следующие:

- концептуальное проектирование – сбор, анализ и редактирование требований к данным (обследование предметной области, изучение ее информационной структуры, выявление всех фрагментов, каждый из которых характеризуется пользовательским представлением, информационными объектами и связями между ними, процессами над информационными объектами, моделирование и интеграция всех представлений);

- логическое проектирование – изменение требований к данным в структуры данных. На выходе получаем СУБД-ориентированную структуру базы данных и спецификации прикладных программ;

- физическое проектирование – определение особенностей хранения данных, методов доступа и т.д.

Современные объектно-ориентированные CASE-средства позволяют эффективно решать задачи проектирования приложений. Среди таких пакетов – Rational Rose, Together Control Center, BPWin, ERWin, Model Mart, Silverrun Business Process Modeller, Process Analyst.

Для разработки функциональной модели использовалось CASE-средство Computer Associates BPwin 4.0. BPwin является непосредственно инструментом для создания различных моделей, которые позволяют анализировать, документировать и планировать изменения сложных бизнес-процессов. BPwin предлагает средство для сбора необходимой нами информации о работе предприятия и графического изображения этой

информации в виде целостной и непротиворечивой модели. BPwin поддерживает три методологии: IDEF0, DFD и IDEF3, позволяющие анализировать ваш бизнес с трех ключевых точек зрения:

- с точки зрения функциональности системы. В рамках методологии IDEF0 бизнес-процесс представляется в виде набора элементов-работ, которые взаимодействуют между собой, а также выводятся информационные, людские и производственные ресурсы, потребляемые каждой работой;

- с точки зрения потоков информации (документооборота) в системе. Диаграммы DFD могут дополнить то, что уже отражено в модели IDEF3, поскольку они описывают потоки данных, позволяя проследить, каким образом происходит обмен информацией между бизнес-функциями внутри системы. В тоже время диаграммы DFD оставляют без внимания взаимодействие между бизнес-функциями;

- с точки зрения последовательности выполняемых работ: еще более точную картину можно получить, дополнив модель диаграммами IDEF3. Этот метод привлекает внимание к последовательности выполнения событий.

Функциональная модель предназначена для описания имеющих бизнес-процессов на предприятии. Она предписывает разработку иерархической системы диаграмм – единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция – система делится на подсистемы, и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности. Такая технология создания модели позволяет построить необходимую модель, адекватную предметной области на всех уровнях абстрагирования.

Разработка функциональной модели производилась с помощью CASE - средства BPWin 4.0.

BPWin 4.0 поддерживает три методологии: IDEF0, IDEF3, DFD.

Методология IDEF0 является наиболее удобным языком моделирования бизнес-процессов. В IDEF0 система представляется как совокупность взаимодействующих работ или функций. Такая чисто функциональная ориентация является принципиальной - функции системы анализируются независимо от объектов, которыми они оперируют. Это позволяет более четко смоделировать логику и взаимодействие процессов организации. Находясь под управлением, система преобразует входы в выходы, используя механизмы.

Диаграммы потоков данных (DFD) используются для описания рабочего процесса и обработки информации. Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели IDEF0 для более наглядного

отображения текущих операций рабочего процесса в корпоративных системах обработки информации. В отличие от IDEF0, DFD рассматривает систему как целостность предметов. Контекстная диаграмма часто включает работы и внешние ссылки.

IDEF3 - это метод, основной целью которой является дать возможность аналитикам описать нынешнюю ситуацию, когда процессы выполняются в определенной очередности, а также непосредственно описать объекты, участвующие совместно в одном процессе. IDEF3 дополняет IDEF0 и содержит все необходимое для построения моделей, которые в дальнейшем могут быть использованы для имитационного анализа. Каждая работа в IDEF3 выводит какой-либо сценарий бизнес-процесса и может являться составляющей другой работы.

В конечном итоге проведенного анализа предметной области была сформулирована функциональная модель системы по ведению количественного учета в организации, занимающейся ремонтом и обслуживанием компьютерной техники.

Поскольку в разрабатываемой системе автоматизируется документооборот, то удобно использовать нотацию DFD. Нотация DFD включает такие понятия, как внешняя ссылка и хранилище данных, что делает ее более удобной (по сравнению с IDEF0) для моделирования документооборота.

Хранилище данных позволяет сформулировать данные, которые необходимо сохранить в памяти прежде, чем использовать в работах. В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в состоянии покоя. В материальных системах хранилища данных изображаются там, где объекты ожидают обработки, например, в очереди. В системах обработки информации хранилища данных представляют собой механизм, который позволяет сохранять данные для последующих процессов.

Рассматриваемая система – система автоматизации работы производственного предприятия. Целью моделирования является описание процесса преобразования исходной информации о стоимости продукции в информацию о конечной стоимости продукции, а также отчетов и диаграмм. В результате анализа предметной области была разработана функциональная модель АИС учета затрат на производство. Проектирование проводилось на основе методологий IDEF0 и DFD.

Контекстная диаграмма, приведенная на рисунке 2.1, реализована с помощью методологии IDEF0.



Рисунок 2.0.1 – Контекстная диаграмма системы

На данном этапе выполняются следующие функции:

- обработка первичной документации;
- составление сметы;
- формирование отчетов.

Диаграмм декомпозиции представлена на рисунке 2.2



Рисунок 2.2 – Диаграмма декомпозиции первого уровня

Дальнейшая декомпозиция системы производилась на основе методологии DFD. Декомпозиции процессов «Обработка первичной документации» и «Составление сметы» представлены на рисунках 2.3 и 2.4 соответственно.



Рисунок 2.3 – Декомпозиция процесса «Обработка первичной документации»



Рисунок 2.4 – Декомпозиция процесса «Составление сметы»

2.2.2 Инфологическая модель системы

Цель инфологического проектирования – обеспечение предельно естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Следовательно, модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка). Основными конструктивными элементами моделей являются сущности, связи между ними и их свойства (атрибуты).

Разбиение на таблицы осуществляется в согласно с семантическим анализом предметной области, вместе с тем, обычно каждому объекту (сущности) предметной области ставится в соответствие таблица, атрибутам объекта соответствуют атрибуты таблицы, а идентификатору объекта соответствуют ключ таблицы.

Схема БД может быть незадавшейся, т.е. могут возникать избыточность и аномалии (аномалия обновления, аномалия включения, аномалия удаления). Нормализация данных представляет собой процедуру, обеспечивающую соответствие информационной модели некоторым стандартам. Это означает минимизацию дублирования, обеспечение гибкости, необходимой для поддержки различных функциональных требований, и создание условий для целесообразного отображения модели на разнообразные проекты БД.

Процесс нормализации, идущий параллельно с проектированием, включает в себя:

- обнаружение существенных объектов, информация о которых подлежит выяснению или запоминанию. Этим сущностям необходимо взаимно исключать друг друга;
- добавление связей, представляющих поименованные отношения между сущностями;
- для каждой сущности составляется список сведений (атрибутов), которые нужно знать о ней;
- определить, как именно каждое вхождение сущности можно уникально идентифицировать.

Для построения инфологической модели данных используется удобный инструмент – ERWin (средство разработки структуры базы данных). Обычно разработка модели базы данных состоит из двух этапов: составление логической модели и создание на ее основе физической модели. ERwin целиком поддерживает такой процесс, он имеет два представления модели: логическое и физическое. Из чего следует, что разработчик может строить логическую модель базы данных, не задумываясь над деталями физической реализации, т. е. обращая основное внимание на требования к информации и бизнес-процессам, которые будет поддерживать будущая база данных.

При планировании структуры новой базы данных определяют сущности (объекты, явления) предметной области, которые должны найти свое отражение в базе данных. Объект – это такая абстракция множества предметов реального мира, что все экземпляры этого объекта имеют одни и те же характеристики и подчиняются одним и тем же правилам поведения. Объекты обладают определенными свойствами – атрибутами. Атрибут – это абстракция одной характеристики объекта. Каждый атрибут имеет имя и может получать значения из некоторого множества допустимых значений. Как правило, каждому объекту в базе данных соответствует таблица, а его атрибутам – поля этой таблицы.

В результате анализа были выведены 9 объектов, которые характеризуют данную предметную область.

В проектируемой модели была использована логико-физическая модель.

Между объектами предметной области существуют связи, которые должны быть отражены в виде связей между объектами инфологической модели. Связь является логическим соотношением между сущностями. Графически связь обозначается линией, соединяющей связываемые объекты. В каждом направлении связи можно выделить главный объект, от которого идет связь, и подчиненный.

Различают идентифицирующую связь и неидентифицирующую связь. При установлении неидентифицирующей связи дочерняя сущность остается независимой. Экземпляр сущности родителя может существовать безотносительно к какому-либо экземпляру дочерней сущности. При идентифицирующей связи экземпляр подчиненной сущности зависит от родительской сущности и не может существовать без экземпляра родительской сущности.

Мощность связи задает максимальное число экземпляров одной сущности, которые могут быть связаны с экземплярами другой сущности. В отношении «один-к-одному» один экземпляр сущности связан с одним экземпляром другой сущности. В отношении «один-ко-многим» один экземпляр родительской сущности связан со многими экземплярами дочерней сущности. Отношения «многие-ко-многим» появляются там, где один экземпляр одной сущности связан с несколькими экземплярами другой, и один экземпляр этой другой сущности также связан с несколькими экземплярами первой сущности.

Исходя из особенностей предметной области, выделенных объектов и их атрибутов, разработанной структуры связей была построена ER-диаграмма на логическом уровне (Рисунок 2.5).

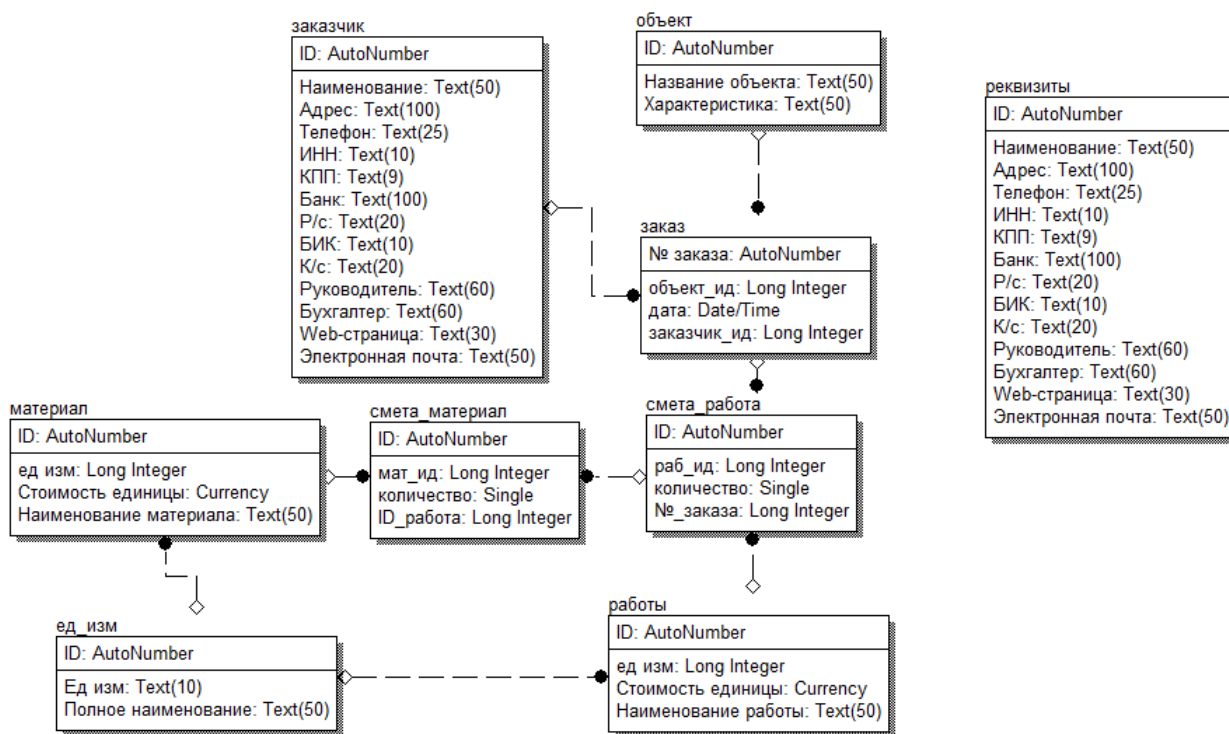


Рисунок 2.5 – ER-диаграмма на логическом уровне

Схема базы данных может быть несостоявшейся: возникают избыточность и аномалии. Нормализация – процесс проверки и реорганизации сущностей и атрибутов с целью удовлетворения требований к реляционной модели данных. Процесс нормализации имеет своей целью предотвращение избыточности данных и аномалий.

В рамках реляционной модели данных существует несколько нормальных форм отношений (нормальные формы ограничивают определенный тип функциональной зависимости и предотвращают аномалии при выполнении операций над отношениями):

- первая нормальная форма если все атрибуты отношения являются атомарными (неделимыми). Понятие атомарности является условным. Важно отметить, что атрибут является атомарным, если его значение не используется по частям;

- вторая нормальная форма если отношения находятся в первой нормальной форме и каждый не ключевой атрибут функционально полно зависит от составного ключа;

- третья нормальная форма если отношения находятся во второй нормальной форме и в них нет транзитивных зависимостей не ключевых атрибутов от ключа.

Имеются и другие виды нормальных форм отношений, но, в большинстве случаев, для сохранения целостности данных хватит и третьей нормальной формы.

Произведенный анализ схемы отношений, показавший отсутствие

многозначных зависимостей в существующей инфологической модели, позволяет прийти к выводу о том, что отношения находятся в третьей нормальной форме.

Физическая модель имеет всю информацию, необходимую для реализации конкретной базы данных. Была произведена генерация схемы физической базы данных для модели предметной области в среде ERWin путем выбора целевой СУБД MS Access 2013. В период конструирования физической модели данных была осуществлена корректировка некоторых типов данных полей в соответствии с реализацией базы данных. Физическая модель данных показана на рисунке 2.6.

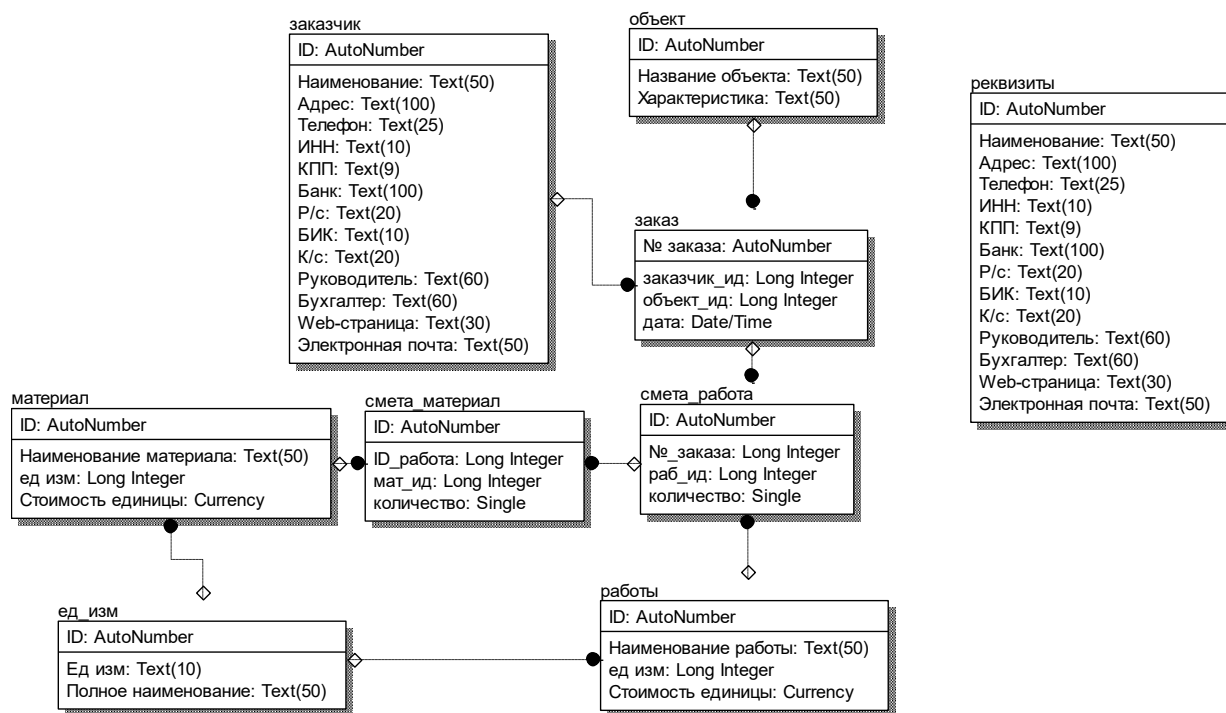


Рисунок 2.6 – ER-диаграмма на физическом уровне

Физическое описание модели удобнее всего сформулировать в виде таблиц. База данных проекта содержит таблицы, названия которых соответствуют именам сущностей инфологической модели. Структура БД описана в таблице 2.1.

Таблица 2.1 Структура полей базы данных

ед_изм		
Name	Datatype	Is PK
Ед_изм	Text(10)	No
Полное наименование	Text(50)	No
ID	AutoNumber	Yes
заказ		

Продолжение таблицы 2.1

Name	Datatype	Is PK
объект_ид	Long Integer	No
дата	Date/Time	No
заказчик_ид	Long Integer	No
№ заказа	AutoNumber	Yes
заказчик		
Name	Datatype	Is PK
К/с	Text(20)	No
БИК	Text(10)	No
Р/с	Text(20)	No
Web-страница	Text(30)	No
Руководитель	Text(60)	No
Бухгалтер	Text(60)	No
Электронная почта	Text(50)	No
Наименование	Text(50)	No
Адрес	Text(100)	No
ID	AutoNumber	Yes
Банк	Text(100)	No
КПП	Text(9)	No
Телефон	Text(25)	No
ИНН	Text(10)	No
материал		
Name	Datatype	Is PK
ед изм	Long Integer	No
Стоимость единицы	Currency	No
Наименование материала	Text(50)	No
ID	AutoNumber	Yes
объект		
Name	Datatype	Is PK
Название объекта	Text(50)	No
Характеристика	Text(50)	No
ID	AutoNumber	Yes
Name	Datatype	Is PK
ед изм	Long Integer	No
Стоимость единицы	Currency	No
Наименование работы	Text(50)	No
ID	AutoNumber	Yes
реквизиты		

Продолжение таблицы 2.1

Name	Datatype	Is PK
К/с	Text(20)	No
БИК	Text(10)	No
Р/с	Text(20)	No
Web-страница	Text(30)	No
Руководитель	Text(60)	No
Бухгалтер	Text(60)	No
Электронная почта	Text(50)	No
Наименование	Text(50)	No
Адрес	Text(100)	No
ID	AutoNumber	Yes
Банк	Text(100)	No
КПП	Text(9)	No
Телефон	Text(25)	No
ИНН	Text(10)	No
смета_материал		
Name	Datatype	Is PK
мат_ид	Long Integer	No
количество	Single	No
ID_работа	Long Integer	No
ID	AutoNumber	Yes
смета_работа		
Name	Datatype	Is PK
раб_ид	Long Integer	No
количество	Single	No
№ заказа	Long Integer	No
ID	AutoNumber	Yes

2.3 Алгоритмы расчётов решаемых задач

Технология внутри машинной организации задается последовательностью реализуемых процедур схем взаимосвязи программных модулей и информационных массивов. Такая схема представляет собой декомпозицию общего процесса решения задачи на отдельные процедуры преобразования массивов, именуемыми модулями (это ввод, контроль, перезапись информации с одного носителя на другой, сортировка, уплотнение данных, редактирование, накопление, вывод на печать и т.п.). Структуру программы можно описать основными блоками, представленными на рисунке 2.7.

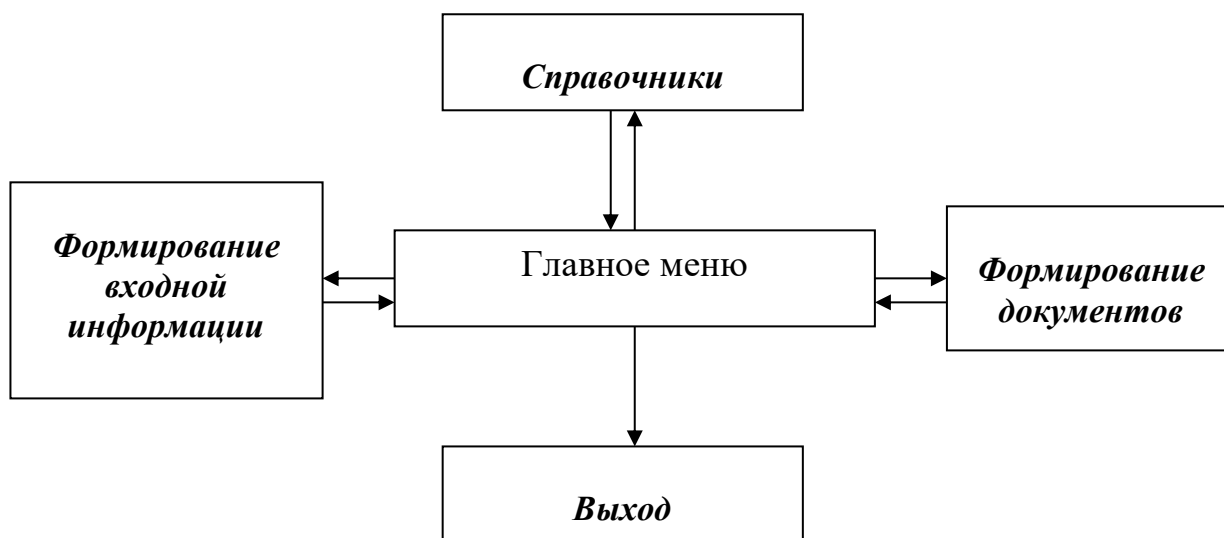


Рисунок 2.7 – Схема взаимодействия основных модулей программы

Приблизительный алгоритм работы с базой данных представлен на рисунке 2.8. Алгоритмы по редактированию данных и занесению их в базу, а также алгоритмы, осуществляющие удаление информации из базы данных, также являются стандартными.

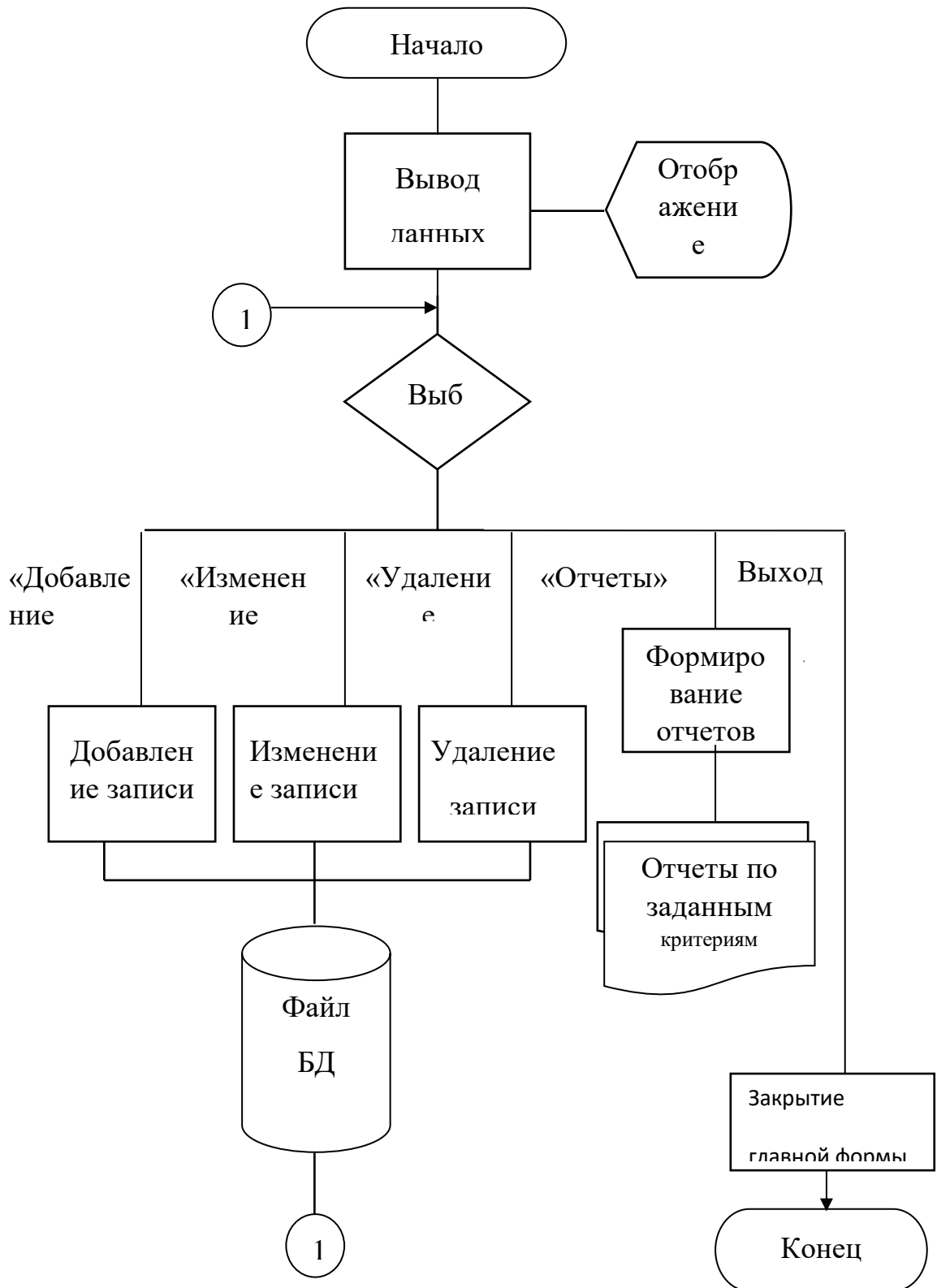


Рисунок 2.8 – Схема работы программы

Система состоит из клиентской (приложение, разработанное с помощью “Borland Delphi 7.0”) и серверной (база данных, разработанная с помощью “Microsoft Access 2013”) частей. Клиентская и серверная часть

обменивается данными следующим образом: при вводе данных пользователем в клиентской части эта информация, после проверки на целостность, попадает в серверную часть, где осуществляется хранение данных; при выводе информации пользователю на экран данные берутся из серверной части программы и отображаются в клиентской. На рисунке 2.9 представлена модель архитектуры системы.

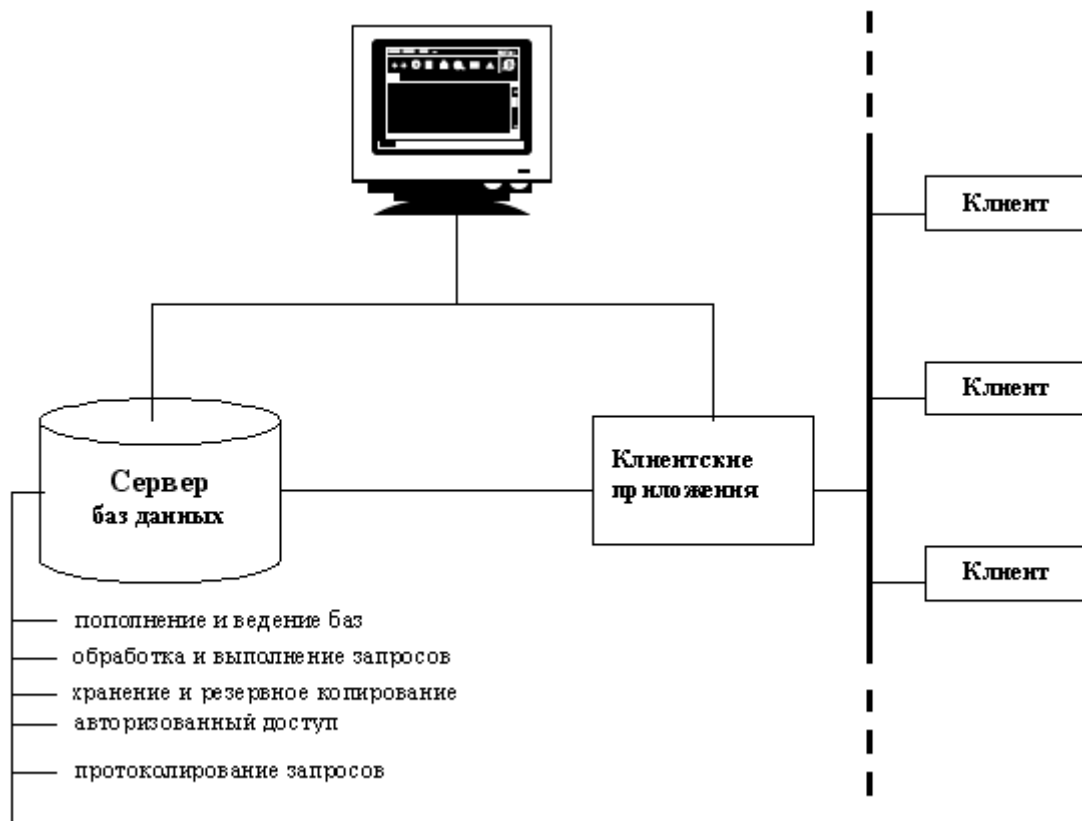


Рисунок 2.9 – Архитектура разработанной системы

2.4 Описание структуры и функциональных алгоритмов программы

Типовая технология формирования массивов информации на машинном носителе можно представить в виде схемы, представленной на рисунке 2.10

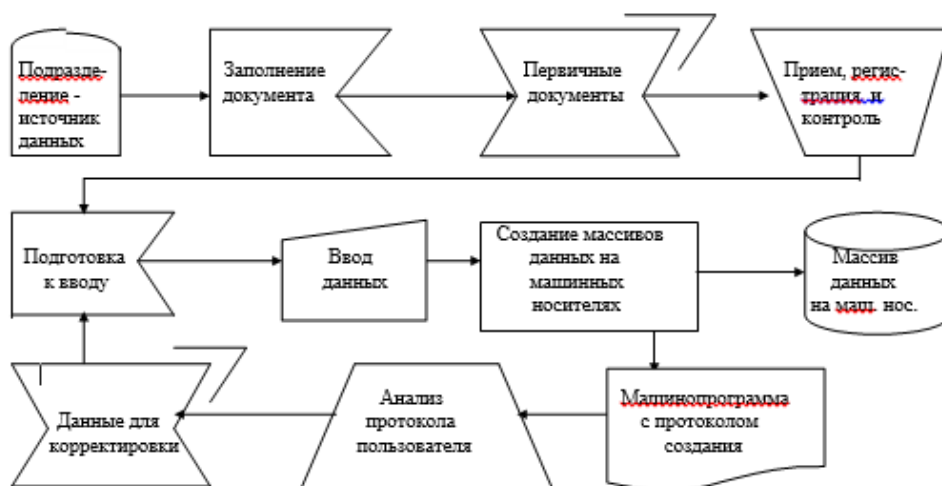


Рисунок 2.10 – Технология формирования массивов информации

На рисунке 2.11 представлен алгоритм формирования отчетов по данным, содержащимся в базе данных.

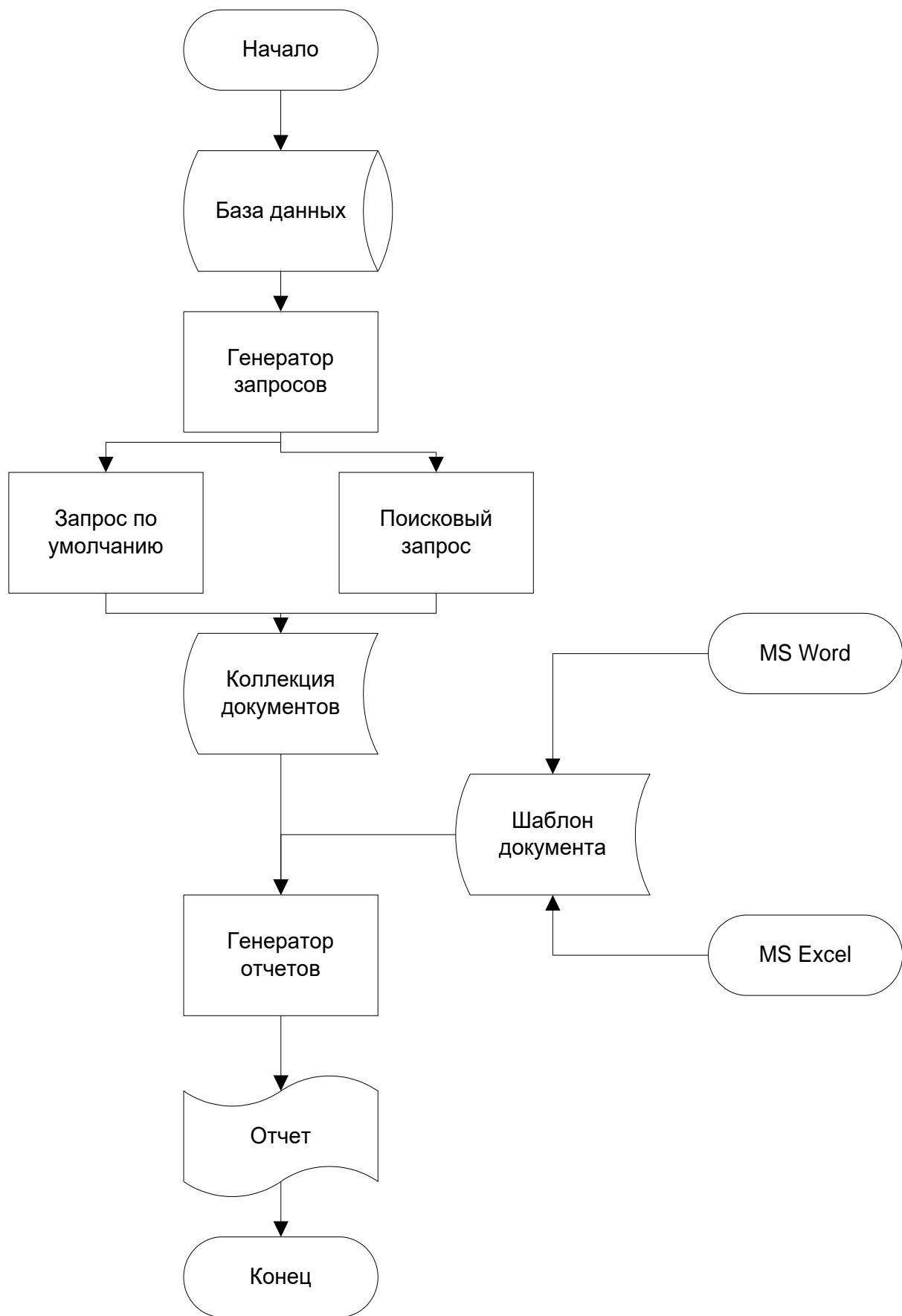


Рисунок 2.11 – Алгоритм построения отчетов

3 Экспериментальная часть

3.1 Инструкция пользователя

Для работы с программой необходимо запустить исполняемый файл программы SMETA.EXE.

При запуске программы на экране появляется форма «ЭИС «Внутрифирменное определение стоимости проекта». Данная форма является связующей со всеми остальными. На ней присутствует элемент меню, которое представлено следующими категориями: «Файл», «Справочники», «Планирование» и «Отчеты». В каждой из категории располагаются элементы меню, обеспечивающие переход к той или иной форме. Данная форма представлена на рисунке 3.1.

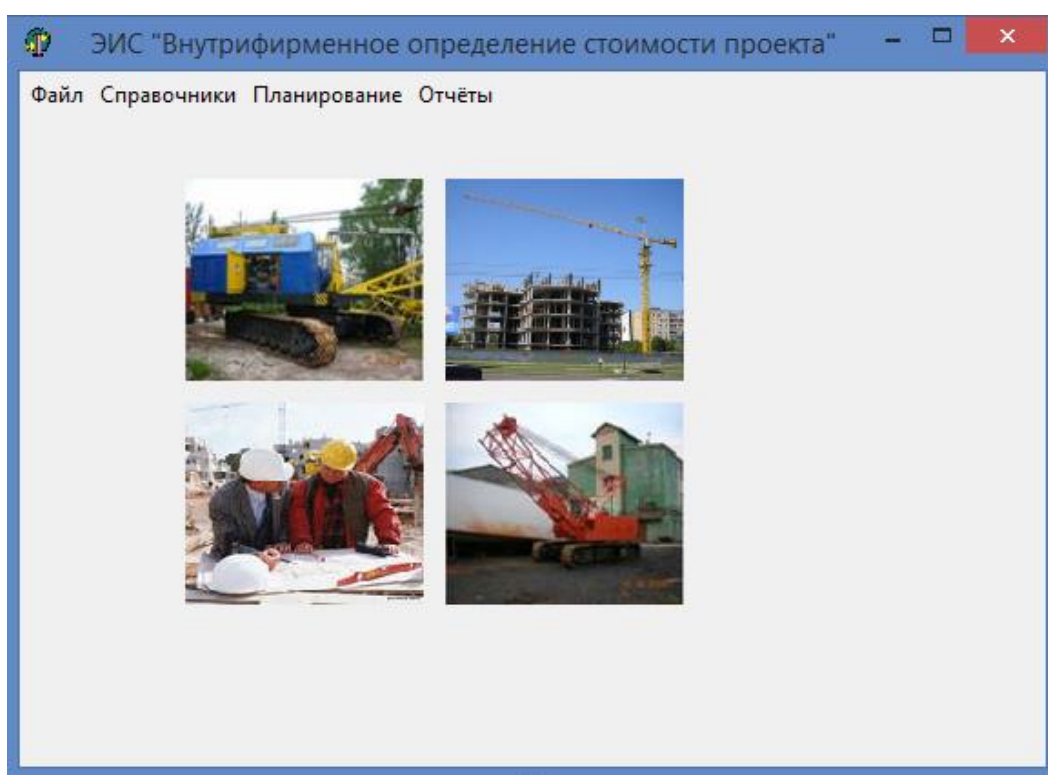


Рисунок 3.1 – Главная форма приложения

При выборе пункта меню «Справочники»-> «Реквизиты» появляется форма с реквизитами. На рисунке 3.2 представлена форма с реквизитами организации.

Рисунок 3.2 – Реквизиты организации

Чтобы вывести на печать реквизиты организации, следует нажать кнопку «Вывести на печать». При этом на экране отобразится отчет, представленный на рисунке 3.3

	A	B	C	D	E	F	G
1		Реквизиты организации					
2							
3	Название организации	TOO "Аль Хади Адилет"	Банк	АО "Банк ЦентрКредит"			
4							
5	ИНН	6004243018	БИК	0904400041			
6							
7	КПП	50639629	к/с	40702810600502000398			
8							
9	Адрес	г.Алматы, бульв. Бухар жырау,	р/с	30103810500504535459			
10							
11	Телефон	+7 (707) 737-87-25	ФИО руководителя	Даулбаев К.Б.			
12							
13	Сайт	ww.pk.uchet.kz/	ФИО гл. бухгалтера	Мулдашева Г.К.			
14							
15	Электронная почта	d.kairat@bk.ru					
16							
17							
18							

Рисунок 3.3 – Отчет с реквизитами

Работа со справочной информацией происходит следующим образом: выбирается необходимый справочник, после этого в таблице данных предоставляется возможность добавления, удаления и редактирования. Для перехода по записям предусмотрены навигационные кнопки, при нажатии на которые указатель текущей записи переходит либо на предыдущую запись, либо на следующую.

При выборе пункта «Заказчики» на экране отображается окно, представленное на рисунке 3.4. На данной форме можно добавлять, удалять, редактировать и искать информацию о клиентах организации при помощи навигационных интерфейсных элементов. Предусмотрен поиск по частичному совпадению по наименованию, ИНН, КПП.

Наименование	ИНН	КПП
ТОО "almaz-group"	2345234534	457654

Наименование: ТОО "almaz-group"

ИНН: 2345234534 КПП: 457654654

Адрес: г. Алматы, ул. Токтогула, 75.

Тел.: +7 (727) 345-45-34 Электронная почта: alim.rah@gmail.com

ФИО руководителя: Алимбаев Р. Д.

ФИО гл. бухгалтера: Абдулаева Ж. П.

Наименование банка: АО "ТемірБанк" БИК: 345345345

к/с: 301038105005045354 р/с: 4070281060080200089

Web-сайт: отсутствует

Закреть окно

Рисунок 3.4 – Форма «Заказчики»

При выборе пункта «Работы» на экране отображается окно, представленное на рисунке 3.5. На данной форме можно добавлять, удалять, редактировать и искать информацию о работах, выполняемых организацией, при помощи навигационных интерфейсных элементов. Предусмотрен поиск по частичному совпадению по наименованию работы.

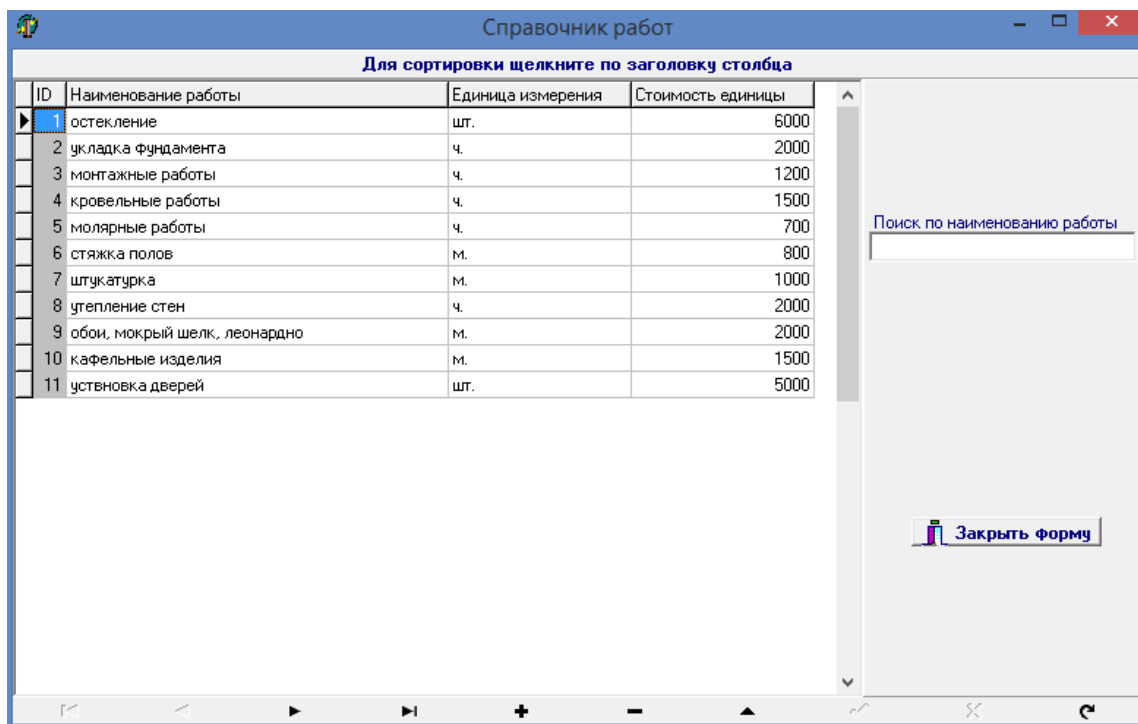


Рисунок 3.5 – Форма «Справочник работ»

При выборе пункта «Объекты» на экране отображается окно, представленное на рисунке 3.6. На данной форме можно добавлять, удалять, редактировать и искать информацию о объектах, на которых организацией выполняются работы, при помощи навигационных интерфейсных элементов. Предусмотрен поиск по частичному совпадению по названию объекта.

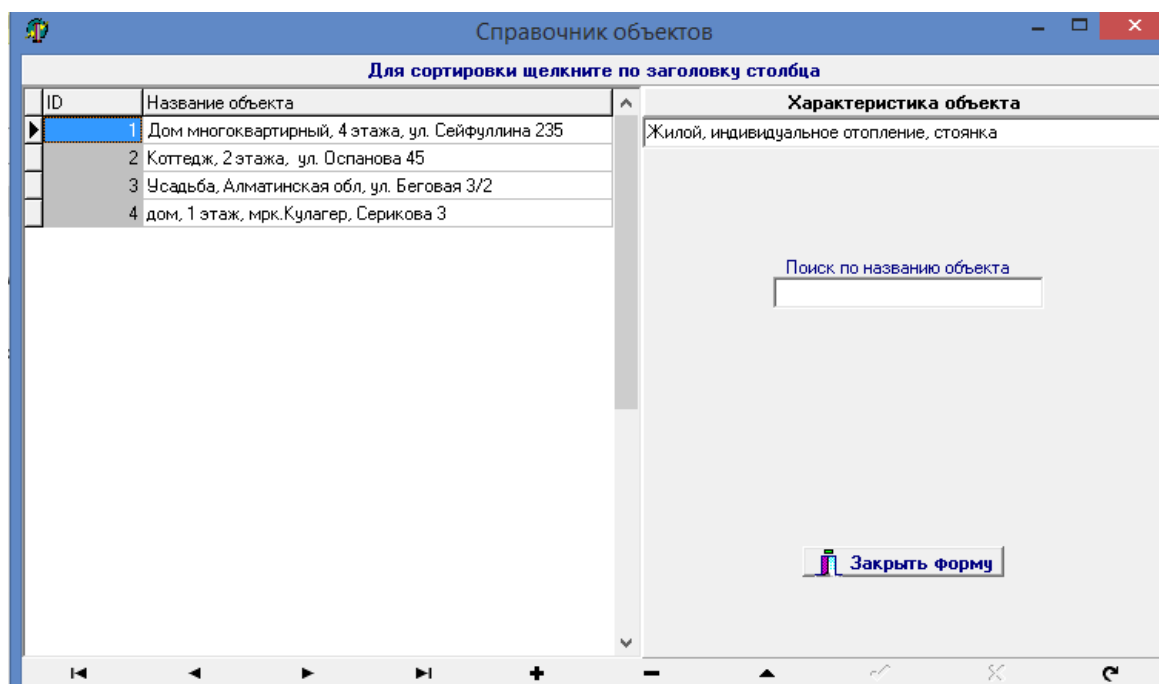


Рисунок 3.6 – Форма «Справочник объектов»

При выборе пункта «Материалы» на экране отображается окно, представленное на рисунке 3.7. На данной форме можно добавлять, удалять, редактировать и искать информацию об используемых в работе материалах, при помощи навигационных интерфейсных элементов. Предусмотрен поиск по частичному совпадению по названию материала.

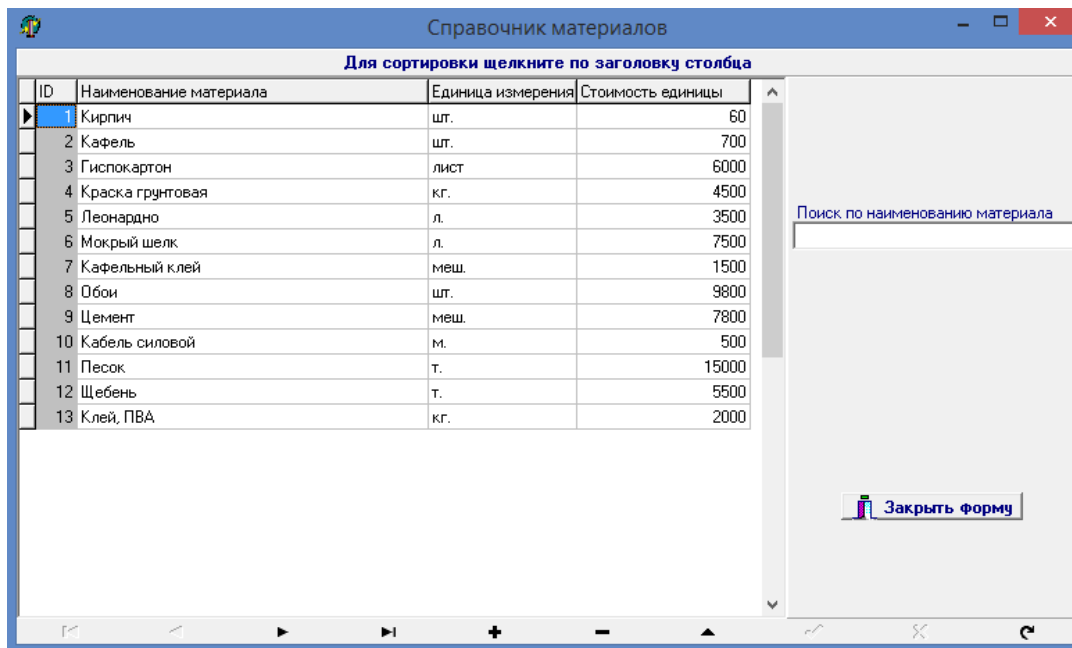


Рисунок 3.7 – Форма «Справочник материалов»

При выборе пункта «Ед. измерения» на экране отображается окно, представленное на рисунке 3.8. На данной форме можно добавлять, удалять, редактировать и искать информацию о единицах измерения материалов, при помощи навигационных интерфейсных элементов. Предусмотрен поиск по частичному совпадению по названию в названии единицы измерения.

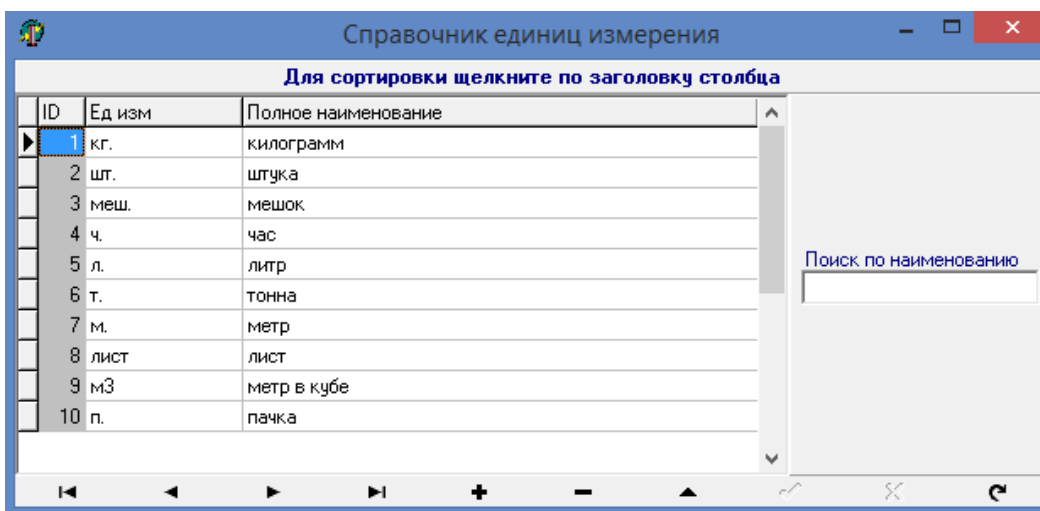


Рисунок 3.8 – Форма «Справочник единиц измерения»

При выборе пункта «Планирование» > «Заказы» на экране отображается окно, представленное на рисунке 3.9. На данной форме можно добавлять, удалять, редактировать и искать информацию о заказах, выполняемых организацией, при помощи навигационных интерфейсных элементов. В верхней части формы сосредоточена информация о заказах (Номер, дата, заказчик, объект), а в нижней – расположены сведения о работах, выполняемых в рамках данного заказа.

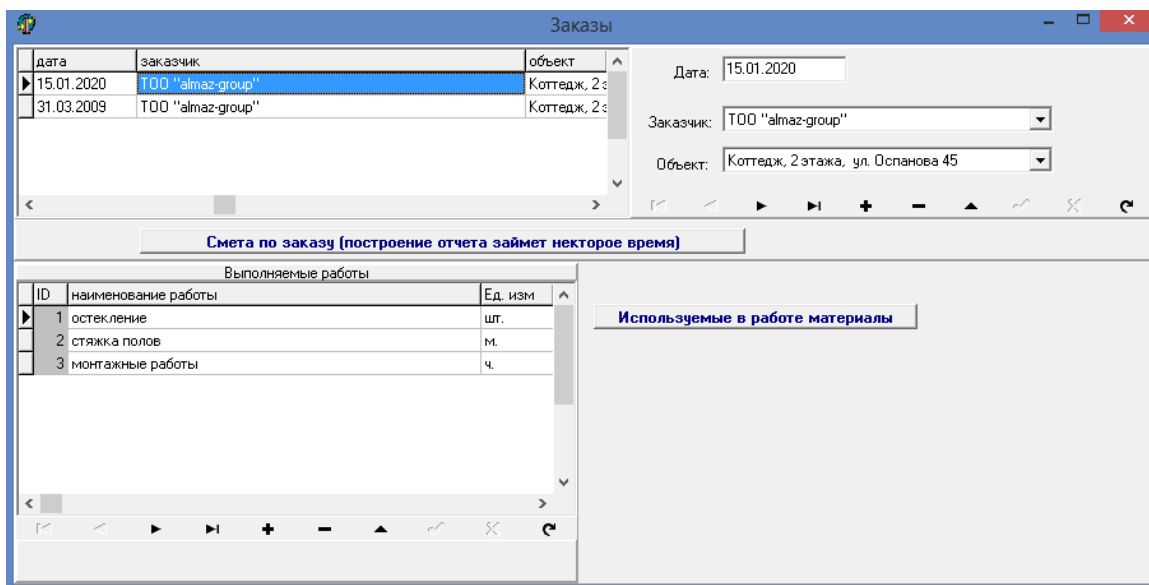


Рисунок 3.9 – Форма «Заказы»

В каждой из работ, задействованы определенные материалы. Для просмотра материалов, использованных при выполнении выбранной работы, следует нажать кнопку «Используемые в работе материалы». При этом на экране отобразится форма, представленная на рисунке 3.10

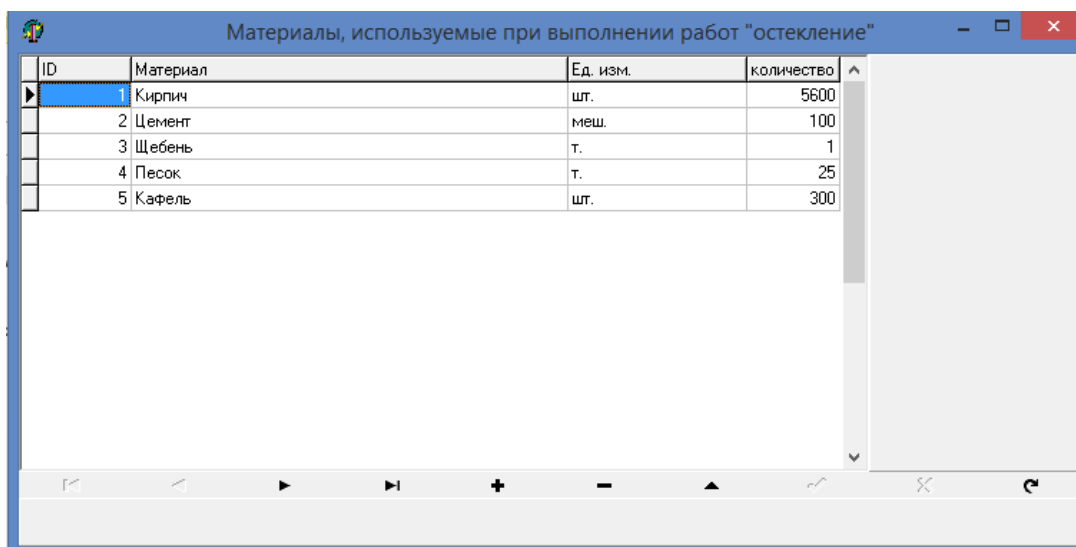


Рисунок 3.10 – Форма «Материалы, используемые в работе»

Для формирования сметы по выбранному заказу следует нажать кнопку «Смета по заказу». Построение данного отчета занимает некоторое время, после чего на экране появляется отчет «Смета». Внешний вид отчета представлен на рисунке 3.11.

Номер по порядку	Номер по смете	Наименование работ	Единица измерения	количество	цена за единицу, тенге	Выполнено работ, тенге	Материалов
23	1	остекление	шт.	20,0	6000,0	120000,0	
24	2	Кирпич	шт.	5600,0	60,0		336000,0
25	3	Цемент	меш	100,0	7800,0		780000,0
26	4	Щебень	т.	1,0	5500,0		5500,0
27	5	Песок	т.	25,0	15000,0		375000,0
28	6	Кафель	шт.	300,0	700,0		210000,0
29	7	стяжка полов	м.	15,0	800,0	12000,0	
30	8	Кафель	шт.	300,0	700,0		210000,0
31	9	Краска грунтовая	кг.	6,0	4500,0		27000,0
32	10	монтажные работы	ч.	40,0	1200,0	48000,0	
33	11	Кирпич	шт.	2000,0	60,0		120000,0
34	12	Гиспокартон	лист	10,0	6000,0		60000,0
Итого стоимость работ :						180000,0	
Стоимость материалов:							2123500,0
Итого (работы + материалы) :						2 303 500,0	
41	Принят					Алимбаев Р.Д. расшифровка подписи	
42							
43	Сдал					Даулбаев К.Б. расшифровка подписи	
44							
45				МП			

Рисунок 3.11 – Отчет «Смета»

При выборе пункта «Отчеты» > «Расход по материалам за период» на экране отображается окно, представленное на рисунке 3.12.

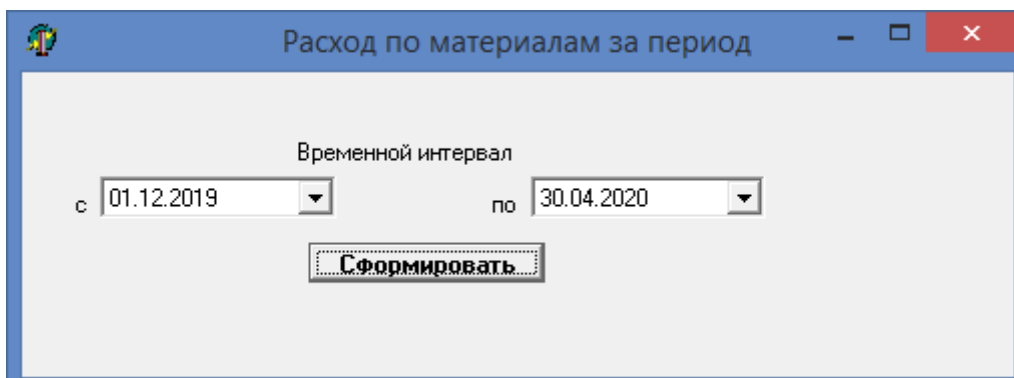


Рисунок 3.12 – Форма расхода по материалам

В данном окне нужно указать интервал дат, за который требуется получить информацию и нажать кнопку «Сформировать». При этом на экране появляется отчет «Расход материалов». Внешний вид отчета представлен на рисунке 3.13

№ п/п	Наименование материала	Ед. изм.	Базовая стоимость, тенге	Суммовой расход, тенге.
1	Краска грунтовая	кг.	4 500,00	27 000,00
2	Гипсокартон	лист	6 000,00	60 000,00
3	Цемент	меш.	7 800,00	780 000,00
4	Песок	т.	15 000,00	375 000,00
5	Щебень	т.	5 500,00	5 500,00
6	Кафель	шт.	700,00	420 000,00
7	Кирпич	шт.	60,00	456 000,00
	Итого			2 123 500,00

Рисунок 3.13 – Отчет «Расход материалов за период»

При выборе пункта «Отчеты» > «Расход о расходах по объектам за период» на экране отображается окно, представленное на рисунке 3.12. В данном окне нужно указать интервал дат, за который требуется получить информацию и нажать кнопку «Сформировать». При этом на экране

появляется отчет «Расходы по объектам». Внешний вид отчета представлен на рисунке 3.14

№ п/п	Дата	Наименование объекта	Заказчик	Затраты по работам, тенге	Затраты по материалам, тенге
1	15.01.2020	ТОО "almaz-group" г. Алматы, ул. Токтогулова 75, +7(727) 745-45-34	Коттедж, 2 этажа, ул. Оспанова 45	180 000,00	2 123 500,00
			Итого	180 000,00	2 123 500,00

Рисунок 3.14 – Отчет «Расходы по объектам»

3.2 Пример расчёта программы

Для проверки правильности работы программы необходимо испытать её функционирование на тестовых примерах. Тестовые примеры должны оптимальный набор входных данных и должны быть разработаны таким образом, чтобы процесс тестирования можно осуществить проведением последовательности определённых операций, охватывающих все функциональные возможности программы.

В качестве примера рассчитаем затраты по объекту «Коттедж, 2 этажа, ул Оспанова 45». Заказчик «ТОО almaz-group», Дата заказа «30.03.2020».

Наименование работ: «утепление стен».

Используемые материалы: «Гипсокартон» в количестве 20 листов.

С учетом стоимости работы «Утепление стен» 2000 тенге и стоимости гипсокартона 6000 тенге за лист, итоговая стоимость работ и материалов по смете должна получиться как $2000 + 6000 * 20$, т.е. 160 000 тенге.

Смета сформированная программой выдает тот же результат. Из чего можно сделать вывод, что программа работает правильно. См. рисунок 3.15

№		А	В	С	Д	Е	Ф	Г	Н	И	Ж
3										Код	
4							Форма по ОКУД			322001	
5	Заказчик	ТОО "almaz-group" г. Алматы, ул. Токтогулова 75 +7 (727) 345-45-34					по ОКПО				
6		<small>организация, адрес, телефон, факс</small>									
7	Генподрядчик	ТОО "Аль Хали Адилет" г. Алматы, бульв. Бухар жырау, д.64, +7 (707) 737-87-25					по ОКПО				
8		<small>организация, адрес, телефон, факс</small>									
9	Субподрядчик	ТОО "Аль Хали Адилет" г. Алматы, бульв. Бухар жырау, д.64, +7 (707) 737-87-25					по ОКПО			54443490	
10		<small>организация, адрес, телефон, факс</small>									
11	Объект	Коттедж, 2 этажа, ул. Оспанова 45									
12		<small>наименование, адрес</small>									
13							Вид деятельности по ОКДП				
14						Договор подряда (контракт)	номер			5	
15							дата			30.03.2020	
16							Вид операции				
17						Номер документа	Дата составления			Отчетный период	
18										с	по
19		Сметная стоимость работ и материалов				5	30.03.2020				
20	№ по порядку	№ по смете	Наименование работ		Единица измерения	количество	цена за единицу, тенге	Выполнено работ, руб.			
21	у							стоимость, тенге		Работ	Материалов
22											
23	1	1	утепление стен		ч.	20,0	2000,0	40000,0			
24	2		Гиспокартон		лист	20,0	6000,0				120000,0
25											
26											
27			Итого стоимость работ :					40000,0			
28			Стоимость материалов:								120000,0
29			Итого (работы + материалы) :					160 000,0			
30											
31	Принял									Алимбаев Ж.П.	
32						подпись				расшифровка подписи	
33	Сдал									Даулбаев К.Б.	
34						подпись				расшифровка подписи	

Рисунок 3.15 – Отчет «Смета»

4 Экономический расчет

Темой дипломного проекта является – «Разработка автоматизированной системы расчета и калькуляции стройматериалов».

Целью данной дипломной работы является разработка автоматизированной системы определения стоимости проектов. Основной сферой деятельности предприятия является оказание строительных услуг сторонним организациям и частным лицам. Клиентами фирмы являются как крупные компании, так и физические лица.

В итоге проведенных исследований было принято решение разработать собственную автоматизированную систему учета определения стоимости проектов.

В разработке программного обеспечения будет участвовать группа специалистов, которая включает в себя: технический руководитель, программист-разработчик. В обязанности технического руководителя входит соблюдение и разработка рабочих графиков, контроль персонала и его полезные действия, в первую очередь за качество будет отвечать именно технический руководитель. Программист-разработчик в свою очередь занят разработкой технической части и его обоснования, разработкой программного обеспечения и его тестирование. Следовательно, весь основной процесс разработки осуществляет программист-разработчик, в то время как технический руководитель занимается организационными вопросами.

Технико-экономическое обоснование имеет следующие пункты:

- расчет трудоемкости разработки ПО;
- расчет затрат на разработку ПО;
- расчет затрат на электроэнергию;
- расчет затраты на оплату труда;
- расчет затрат по социальному налогу;
- амортизация основных фондов и прочие затраты;
- определение возможной (договорной) цены ПО;
- расчет эксплуатационных затрат;
- сравнительный анализ эксплуатационных затрат до и после внедрения;
- расчет основных показателей экономической эффективности;
- вывод по экономической части.

4.1 Расчет трудоемкости разработки ПО

Таблица 4.1 - Распределение работ по этапам и видам и оценка их трудоемкости

Этап разработки	Вид работы	Трудоемкость разработки, чел.×ч.
1	Постановка задачи	10
2	Сбор исходных данных	15
3	Поиск и изучение аналогичных программ	10
4	Оформление теоретической части дипломной работы	10
5	Определение и анализ требований к проектируемой программе	25
6	Техническое проектирование разработки программы	80
7	Программирование модулей в выбранной среде программирования	45
8	Тестирование программных модулей	30
9	Сборка и испытание программы	20
10	Оформление рабочей документации	20
ИТОГО		265

Продолжительность рабочего дня равна 8 часам: $265 / 8 = 33$

В результате для реализации программного обеспечения необходимо 33 рабочих дней.

4.2 Расчет затрат на разработку ПО

Определение затрат на разработку ПО производится путем составления соответствующей сметы, которая включает следующие статьи:

- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов;
- прочие затраты.

Материальные затраты делятся на основные и вспомогательные. Расчет материальных затрат происходит по форме, предоставленной в таблице 4.2.1.

Таблица 4.2.1 - Затраты на материальные ресурсы

Наименование материалов	Единица измерения	Требуемое количество	Цена за единицу, тг	Сумма, тг
Тетрадь 96л.	шт.	1	250	250
Блокнот	шт.	1	350	350
Ручка	шт.	3	130	390
Бумага офисная	пачка	2	1700	3400
Итого				4390

Для разработки программного обеспечения будет использоваться персональный ноутбук находящийся на предприятии и ноутбук ACER Nitro 5 AN515-43, мощности ПК и ноутбука достаточно для выполнения поставленных задач. Ввиду того, что ПК содержит установленную операционную систему и программное обеспечение необходимое для разработки ПО, нет нужды производить дополнительные расходы на новые ОС и ПО.

Общую сумму, необходимую на материальные средства (Z_M) можно рассчитать по следующей формуле:

$$Z_M = \sum_{i=1}^n P_i * C_i, \quad (4.2)$$

где P_i - расход i -го вида материального ресурса, натуральные единицы;

C_i - цена за единицу i -го вида материального ресурса, тг;

i - вид материального ресурса;

n - количество видов материальных ресурсов.

Расчет затрат на необходимое оборудование и программное обеспечение производится по форме, приведенной в таблице 4.3.2.

Таблица 4.2.2 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за единицу, тг	Сумма, тг
Ноутбук	ACER Nitro 5 AN515-43	шт.	1	280000	280000
Принтер	HP Laser Jet Pro M15w	шт.	1	39000	39000
Модем	TP-Link Archer C60	шт.	1	13000	13000
Компьютерная мышь	HP Z3700	шт.	1	5500	5500
Итого					337500

$$Z_M = 4390 + 337500 = 341890 \text{ (тг)}$$

Для реализации программного обеспечения необходимы материалы на сумму 341890 тенге.

4.3 Расчет затрат на электроэнергию

Так как при разработке программного обеспечения не обойтись без потребления электроэнергии, имеет смысл произвести расчет затрат на электроэнергию.

Согласно таблице 4.1 для разработки программного обеспечения необходимо порядка 265 часов, теперь необходимо рассчитать стоимость электроэнергии, которая будет потрачена в течении 265 часов. Для принтера расчет будет проводиться для периода в 24 часа, так как нет необходимости постоянно использовать принтер.

Если для разработки ПП используется электрооборудование, то необходимо рассчитать затраты на электроэнергию по форме, приведенной в таблице 4.3.

Общая сумма затрат на электроэнергию (Z_{Σ}) рассчитывается по формуле:

$$Z_{\Sigma} = \sum_{i=1}^n M_i * K_i * T_i * Ц, \quad (4.3)$$

где, M_i - паспортная мощность i -го электрооборудования, кВт;
 K_i - коэффициент использования мощности i -го электрооборудования

(принимается $K_i=0.9$);

T_i - время работы i -го оборудования за весь период разработки ПП ч;

Ц - цена электроэнергии, тг/кВт×ч;

i - вид электрооборудования;

n - количество электрооборудования.

Затраты на электроэнергию находятся исходя из продолжительности периода разработки ПО, количества кВт/ч, затраченных на проектирование ПО и тарифа за 1 кВт/ч. Тариф по городу Алматы для физических лиц с 1 января 2020 года на электроэнергию по тарифу ТОО «АлматыЭнергоСбыт» составляет 19,17 тенге за 1 кВт/ч с учетом НДС.

$$Z_{\Sigma \text{ ноутбук}} = 0,8 \cdot 0,9 \cdot 265 \cdot 19,17 \approx 3657,64 \text{ тг,}$$

$$Z_0 = 0,3 \cdot 0,7 \cdot 265 \cdot 17,62 \approx 980,55 \text{ тг.}$$

Таблица 4.3 - Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена электроэнергии, тг/кВт·ч	Сумма, тг
Ноутбук ACER Nitro 5 AN515-43	0,8	0,9	278	17,62	3657,64
Освещение	0,3	0,7	278	17,62	980,55
ИТОГО					4638,2

4.4 Затраты на оплату труда

В этом разделе рассчитывается основная и дополнительная заработная плата всех работников, которые были задействованы в разработке программного продукта. В разработке ПО было задействовано два работника:

- технический руководитель – управление рабочим временем, корректировка рабочих процессов, координация, изучение предметной области;

- разработчик – разработка ПО, тестирование и сопровождение.

Зарплата руководителя проекта составляет 300 000 тенге, а зарплата программиста-разработчика составляет 250 000 тенге.

Часовая тарифная ставка будет рассчитываться по формуле:

$$Ч_м = N_м \cdot Ч_{рд}, \quad (4.4.1)$$

где, $Ч_м$ — рабочие часы сотрудника за месяц;

$N_м$ — количество рабочих дней за месяц;

$Ч_{рд}$ — количество рабочих часов в день.

$$Ч_м = 21 \cdot 8 = 168 \text{ ч}$$

Ставка работника за час работы рассчитана по формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i}, \quad (4.4.2)$$

Технический руководитель проекта:

$$ЧС_i = \frac{300000}{168} = 1785,7 \text{ тг}$$

Программист-разработчик:

$$ЧC_i = \frac{250000}{168} = 1488 \text{ тг}$$

где, $ЗП_i$ - заработная плата в месяц i -го работника, тг;

$ФРВ_i$ - фонд рабочего времени в месяц i -го работника, час.

Часовая ставка руководителя составляет 1785,7 (тг/ч), трудоемкость разработки равняется 100 часам. Часовая ставка разработчика составляет 1488 (тг/ч), трудоемкость разработки равняется 265 часам. Согласно формуле (4.5) можно рассчитать сумму расходов на заработную плату работников:

$$З_{mp} = 1785,7 \cdot 100 + 1488 \cdot 265 = 178\,570 + 394\,320 = 572\,890 \text{ тг.}$$

Таблица 4.4 – Затраты на оплату труда

Квалификация	Трудоемкость разработки ПП, чел.×ч	Часовая ставка, тг/ч	Сумма, тг
Руководитель	100	1785,7	178 570
Разработчик	265	1488	394 320
ИТОГО			572 890

Дополнительная заработная плата составляет в среднем 10% от основной заработной платы и рассчитывается по формуле:

$$З_{дi} = З_{oi} \times Нд / 100 = 572\,890 \cdot 0,1 \approx 57\,289 \text{ тенге}$$

где $Нд$ - коэффициент дополнительной заработной платы разработчиков.

$$З_{Фот} = З_{oi} + З_{дi} = 572\,890 + 57\,289 = 630\,179 \text{ тенге}$$

4.5 Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан социальный налог составляет 9,5% от фонда оплаты труда. Социальный налог можно рассчитать по следующей формуле:

Затраты на налоговые отчисления рассчитаны в таблице 4.5

Таблица 4.5.1 – Затраты на налоговые отчисления

Уплаченные налоги юридическим лицом	10,455	ФОТ	630 179
СО (Социальные отчисления)	3,5	(ЗП - ОПВ)*3,5%	19 850,64
ВОСМСЮ (Отчисления на ВОСМСЮ)	2,0	ЗП*2%	12 603,58
СН (Социальный налог)	9,5	(ЗП - ОПВ - ВОСМСФ)*9,5%-СО	33 431,00
Всего уплаченные налоги			65 885,21

4.6 Амортизация основных фондов и прочие затраты

Нормы амортизации ОФ необходимо определить в соответствии с налоговым кодексом РК. Амортизацию ОФ можно определить по следующей формуле:

$$A_z = \frac{C_{об} * H_a}{100} \quad (4.6.1)$$

где, $C_{об}$ – стоимость оборудования;

H_a – норма амортизации (норма амортизация = 25);

Формула (4.8) позволяет рассчитать нужную сумму для амортизационных отчислений за год для ноутбука:

$$Z_{ам} = \frac{C_{обор} * H_a * N}{100 * 12 * t} \quad (4.6.2)$$

где, H_a – норма амортизации (%);

$C_{обор}$ – первоначальная стоимость ПО (тг.);

N – время использования ПО (дн.);

t – количество рабочих дней в месяце.

Теперь необходимо рассчитать норму амортизации за период разработки:

$$Z_{ам} = \frac{280000 * 25 * 33}{100 * 12 * 21} = 9166,7 \text{ тенге}$$

$$Z_{ам} = \frac{39000 * 25 * 33}{100 * 12 * 21} = 1276,8 \text{ тенге}$$

$$Z_{ам} = \frac{13000 * 25 * 33}{100 * 12 * 21} = 425,6 \text{ тенге}$$

$$Z_{\text{ам}} = \frac{5500 * 25 * 33}{100 * 12 * 21} = 180 \text{ тенге}$$

Подобным образом необходимо рассчитать норму амортизации для всего оборудования. Результаты расчетов приведены в таблице (4.6.1).

Таблица 4.6.1 – Амортизация ОФ

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	280000	25	70000	9166,7
Принтер	39000	25	9750	1276,8
Модем	13000	25	3250	425,6
Компьютерная мышь	5500	25	1375	180
Итого:			84375	11049,1

Расходы по статье «Прочие затраты» (P_{zi}) на конкретное ПО включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате:

$$P_{zi} = Z_{oi} \cdot \frac{H_{пз}}{100} \quad (4.6.3)$$

где $H_{пз}$ - норматив прочих затрат в целом по организации в (%), в дипломной работе нужно брать 20%.

$$P_z = 572\,890 \cdot 0,2 = 114\,578 \text{ тенге}$$

Затраты по статье «Накладные расходы» (P_{ni}), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды (P_{ni}), относятся на конкретное ПО по нормативу ($H_{нр}$) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации:

$$P_{ni} = Z_{oi} \cdot \frac{H_{нр}}{100} \quad (4.6.4)$$

где P_{ni} - накладные расходы на конкретную ПО (тыс. тенге);

N_{nr} - норматив накладных расходов в целом по организации в (%), в дипломной работе нужно брать 70%.

$$P_{ni} = 572\,890 \cdot 0,7 = 401\,023 \text{ тенге}$$

Смета расходов на разработку ПО.

На основе всех представленных расчетов необходимо оформить смету расходов на разработку ПО согласно форме, которая приведена в таблице (4.6.2).

Таблица 4.6.2 – Смета затрат на разработку ПО

Статьи затрат	Сумма, тг
Затраты на оборудование	337500
Затраты на материальные ресурсы	4390
Затраты на оплату труда	630179
Социальные отчисления и налоги	65885,21
Затраты на электроэнергию	7504,3
Амортизация основных фондов	11049,1
Прочие расходы	114578
Накладные расходы	401023
Итого по смете:	1572108,61

4.7 Определение возможной (договорной) цены ПО

Стоимость программного обеспечения определяется на основе качества разработанного продукта, сроков его разработки и производительности продукта. Стоимость C_D для программного обеспечения можно рассчитать по следующей формуле:

$$C_D = Z_{nir} \left(1 + \frac{P}{100} \right), \quad (4.7.1)$$

где Z_{nir} – затраты на разработку программного обеспечения, тг;

P – средний уровень рентабельности ПО, (%). Данный параметр принят равным 40%.

$$C_D = 1572108,61 \left(1 + \frac{40}{100} \right) = 2\,200\,952 \text{ тенге}$$

Далее необходимо определить стоимость реализации с учетом НДС. Ставка НДС устанавливается законодательством Республики Казахстан. На

2020 год ставка НДС составляет 12%. Стоимость реализации учитывая НДС можно рассчитать по следующей формуле:

$$C_p = C_{\partial} + C_{\partial} * НДС, \quad (4.7.2)$$

$$C_p = 2\,200\,952 + 2\,200\,952 * 0,12 = 2\,465\,066,3 \text{ тенге}$$

Таким образом, себестоимость программного продукта равна – 2 465 066,3 тенге.

Организация-разработчик участвует в освоении ПО и несет соответствующие затраты, на которые составляется смета, оплачиваемая заказчиком по договору. Затраты на освоение определяются по нормативу ($H_0=10\%$) от себестоимости ПО в расчете на 3 месяца и рассчитываются по формуле:

$$P_{oi} = C_{ni} \cdot \frac{H_0}{100} = 1572108,61 \cdot 0,1 = 157210,9 \text{ тенге}$$

Затраты на сопровождение ПО (P_{ci}). Организация-разработчик осуществляет сопровождение ПО и несет соответствующие расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Затраты на сопровождение определяются по установленному нормативу ($H_c=20\%$) от себестоимости ПО (в расчете на год) и рассчитываются по формуле:

$$P_{ci} = C_{ni} \cdot \frac{H_c}{100} = 1572108,61 \cdot 0,2 = 314421,8 \text{ тенге}$$

Капиталовложения программного обеспечения с учетом затрат на освоение и сопровождение будет:

$$K = 2465066,3 + 157210,9 + 314421,8 = 2\,936\,699 \text{ тенге.}$$

4.8 Расчет эксплуатационных затрат

Годовые эксплуатационные текущие затраты в условиях функционирования информационных технологий (C) рассчитываются по формуле:

$$C = ЗП + ОТ + А + М + НР \quad (4.8.1)$$

где ЗП - годовые затраты на оплату труда специалистов при выполнении ими своих функций в рамках автоматизируемого процесса после внедрения ИТ, тенге;

ОТ - отчисления по социальному налогу, тенге;

А - затраты на амортизацию, тенге;

М - годовые материальные затраты на сопровождение программного продукта, тенге;

НР - накладные расходы, тенге.

Годовые затраты по заработной плате специалистов после внедрения ИТ определяются по формуле:

$$\text{ЗП} = \frac{O_c \cdot Ч_c \cdot 12}{\Phi_{р.в.}} \cdot t_{\text{общ}} \cdot 12 \cdot (1 + K_d) \quad (4.8.2)$$

где O_c - оклад специалиста, тенге/мес.;

$Ч_c$ - численность специалистов, участвующих в процессе, чел.;

$\Phi_{р.в.}$ - годовой фонд рабочего времени, час;

$t_{\text{общ}}$ - трудоемкость решения задач в условиях функционирования ИТ в месяц, час;

K_d - коэффициент дополнительной заработной платы.

Трудоемкость решения задач в условиях функционирования ИТ в месяц вычисляется следующим образом:

$$t_{\text{общ}} = \sum_{\beta=1}^n t_{\beta} \cdot K_{\beta} \quad (4.8.3)$$

где t_{β} - затраты времени на решение β -й задачи, час;

K_{β} - количество решаемых β -х задач в месяц, ед.

В ходе эксплуатации системы предполагается появление серверных ошибок, на обработку одной ошибки в среднем уходит 2 часа, на протяжении месяца эксплуатации может возникать до 60 ошибок, помимо этого ожидается внесение изменений в интерфейс и доработка компонентов по требованию руководства. Всего компонентов в системе 30, на доработку одного уходит в среднем 2 часов.

$$t_{\text{общ}} = 2 \cdot 60 + 2 \cdot 30 = 120 + 60 = 180 \text{ часов}$$

$$\text{ЗП} = \frac{240\,000 \cdot 1 \cdot 12}{1920} \cdot 180 \cdot 12 \cdot 1 = 3\,240\,000 \text{ тенге}$$

Материальные затраты определяются по формуле:

$$M = \sum_{\mu=1}^n C_{\mu} \cdot N_{\mu} \quad (4.8.4)$$

где C_{μ} - цена μ -го вида единицы материальных затрат в условиях функционирования ИТ, тенге.

N_{μ} - используемое количество μ -го вида материальных затрат в месяц, ед.

В условиях функционирования разрабатываемой ИС, не предполагается материальных затрат. Амортизационные отчисления производятся по установленным нормам амортизации, выражаются в процентах к балансовой стоимости оборудования и рассчитываются по формуле:

$$A = C_{\text{обор}} \cdot H_a \quad (4.8.5)$$

где H_a — норма амортизации (25 %);

$C_{\text{обор}}$ — первоначальная стоимость оборудования;

$$A = (337500 \cdot 0,25) = 84375 \text{ тенге.}$$

Отчисления по социальному налогу и накладные расходы рассчитываются так же, как и при разработке информационных технологий.

$$CH = 3\,240\,000 \cdot 10,455\% = 338\,904 \text{ тенге.}$$

$$HP = ЗП \cdot \frac{H_{HP}}{100} = 3\,240\,000 \cdot 0,7 = 2\,268\,000 \text{ тенге}$$

Таблица 4.8 – Эксплуатационные затраты

Затраты на эксплуатацию	Условное обозначение	Значение, тенге	В процентах от общей суммы
Заработная плата	ЗП	3 240 000	54,87%
Социальные отчисления и налоги	ОТ	338 742	5,74%,
Материальные затраты	М	0	0%
Амортизационные отчисления	А	84 375	0,97%
Накладные расходы	НР	2 268 000	38,42%
Итого		5 931 117	

C = 5 931 117 тенге

4.9 Сравнительный анализ эксплуатационных затрат до и после внедрения

Разработка автоматизированной системы расчета и калькуляции стройматериалов предназначена в главную очередь для упрощения и ускорения работы компании. Она упрощает процесс принятия решений по управлению, а также способна повысить эффективность и скорость работы сотрудников, за счет автоматизированной систем и учета определения стоимости проектов.

Для оценки экономии от использования разрабатываемой ИС необходимо сравнить эксплуатационные расходы с ее применением и без.

Список расходов после внедрения:

- основная заработная плата сотрудник – программист, сопровождающий систему, включая отчисляемыми налогами;
- материальные ресурсы;
- амортизация основных фондов;
- накладные расходы.

Список расходов до внедрения:

- основная заработная плата 2 сотрудников по работе с клиентами для определения стоимости проектов;
- материальные ресурсы;
- амортизация основных фондов;
- накладные расходы.

В организации имеется два сотрудника по работе с клиентами (месячная заработная плата – 210 000 тенге), основной задачей которых является определения стоимости проектов, управление проектированием.

ЗП = 5 040 000тенге

СО = 526 932

В их распоряжении находятся два персональных компьютера (стоимость одного ~ 300 000 тенге), принтер (150 000 тенге) и прочая компьютерная периферия (~50 000 тенге); итого затрат на оборудование – 500 000 тенге.

Износ оборудования рассчитывается исходя из 25 % амортизационных отчислений за год.

$$A = (300\,000 \cdot 0,25) + (150\,000 \cdot 0,25) + (50\,000 \cdot 0,25) = 125\,000 \text{ тенге}$$

На заправку картриджей, бумагу прочую канцелярию в месяц уходит порядка 30 000 тенге.

Общие накладные расходы составят:

$$HP = 3П \cdot \frac{H_{HP}}{100} = 5\,040\,000 \cdot 0,7 = 3\,528\,000 \text{ тенге}$$

Статьи затрат с применением ИС включают в себя:

- заработная плата программиста, сопровождающий систему (месячная заработная плата – 270 000 тенге), который осуществляет автоматизированную обработку информации и управление проектами;
- износ используемого оборудования;
- расход материалов не предусмотрено;
- накладные расходы.

Таблица 4.9 – Годовые эксплуатационные затраты

Статьи	Без применения ИС (2 сотрудника по работе с клиентами)	С применением ИС (1 сотрудник – программист, сопровождающий систему)
Годовая заработная плата	5040000	3240000
Социальные отчисления и налоги	526932	338742
Расходуемые материалы	500000	0
Амортизационные отчисления	125000	84375
Накладные расходы	3528000	2268000
Всего	9719932	5931117

Ожидаемая условно-годовая экономия определяется по формуле:

$$\mathcal{E}_{yg} = C_1 - C_2 + \sum \mathcal{E}_i \quad (4.9)$$

где \mathcal{E}_{yg} - величина экономии, тенге;

C_1 и C_2 – показатели текущих затрат по базовому и внедряемому вариантам, тенге;

$\sum \mathcal{E}_i$ – ожидаемый дополнительный эффект от различных факторов, тенге.

$$\mathcal{E}_{yg} = C_1 - C_2 = 9719932 - 5931117 = 3788815 \text{ тенге}$$

4.10 Расчет основных показателей экономической эффективности

Так как разработанная информационная система несет более социальный эффект, чем экономический, целесообразно оценивать его эффективность за счет экономии в сравнении с предыдущим периодом работы без использования ИС.

Величина ожидаемого годового экономического эффекта от внедрения ИС рассчитывается по формуле:

$$\mathcal{E}_r = \mathcal{E}_{yr} - K \cdot E_H \quad (4.10.1)$$

где \mathcal{E}_r - ожидаемый годовой экономический эффект, тенге;

\mathcal{E}_{yr} — ожидаемая условно-годовая экономия, тенге;

K — капитальные вложения, тенге;

E_H - нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле:

$$E_H = \frac{1}{T_H} \quad (4.10.2)$$

где T_H — нормативный срок окупаемости капитальных вложений, лет.

Нормативный срок окупаемости капитальных вложений принимается исходя из срока морального старения -технических средств и проектных решений ИС ($T_H=1,2,3...n$), для программных продуктов срок окупаемости принимаем равным 4 года.

$$E_H = \frac{1}{4} = 0,25$$

$$\mathcal{E}_r = 3\,788\,815 - 2\,936\,699 \cdot 0,25 = 3\,054\,640 \text{ тенге}$$

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{\mathcal{E}_{yr}}{K} \quad (4.10.3)$$

где E_p - расчетный коэффициент экономической эффективности капитальных вложений;

\mathcal{E}_{yr} — ожидаемая условно-годовая экономия, тенге;

K — капитальные вложения на создание системы, тенге.

$$E_p = \frac{3\,788\,815}{2\,936\,699} = 1,29$$

Расчетный срок окупаемости капитальных вложений составляет:

$$T_p = \frac{1}{E_p} \quad (4.10.4)$$

где E_p - коэффициент экономической эффективности капитальных вложений.

$$T_p = \frac{1}{1,29} = 0,78 \text{ года} \approx 9,3 \text{ месяцев}$$

Таблица 4.10 – Показатели сравнительной экономической эффективности от внедрения программного продукта

Наименование показателей	Значение
Условная годовая экономия затрат, тенге	3 054 640
Коэффициент экономической эффективности капитальных вложений (E_p)	1,29
Срок окупаемости капитальных вложений (T_p)	0,78

4.11 Вывод по экономической части

В данной главе дипломной работы содержатся экономические расчеты, которые позволяют определить затраты необходимые для разработки программного продукта. Расчеты включают в себя:

- расчет трудоемкости разработки программного продукта;
- расчет затрат на разработку программного продукта;
- расчет затрат на электроэнергию;
- расчет затрат на оплату труда;
- расчет затрат по социальному налогу;
- амортизация основных фондов, накладные и прочие затраты;

Для покупателя основным показателем будет считаться оптимальная цена программного продукта и его производительность. Цена и полезность программного продукта должна обладать равновесием, чтобы покупатель был заинтересован в приобретении разработки. Качественным результатом для покупателя считается, что приобретенное программное обеспечение полностью покрывает все необходимые задачи, которые встают перед покупателем. Так же в последней главе был произведен расчет договорной цены программного продукта, который равняется 2 936 699 тенге, данное значение является рациональным с точки зрения экономической эффективности.

Коэффициент эффективности указывает на соотношение сэкономленных тенге и потраченных. В заключении можно с уверенностью заявить о том, что на 1 затраченный тенге приходится 1,29 тенге экономии.

Внедрение данного программного продукта в рабочий процесс предприятия позволит значительно снизить стоимость приобретения дорогостоящих аналогов на рынке. Также установка данного программного комплекса позволяет сократить штат сотрудников, что в следствии сокращает затраты на выплату налогов, зарплат и прочих денежных расходов. Соответственно, уменьшится объем технического оснащения, амортизационные отчисления и затраты на электроэнергию.

Таким образом, разработанная информационная система позволила не только упростить процесс управления работой организации и процесс принятия решений руководством, но и значительно сэкономить на различных факторах. Во-первых, автоматизация процесса управления позволит больше не использовать услуги дополнительных сотрудников, так как теперь отсутствует необходимость осуществлять контроль, вся работа организации сосредоточена в ИС. Во-вторых, значительно снизится расход материалов, за счет того, что назначение на задания и все изменения по заданиям можно сделать прямым в системе, не прибегая к формированию документов по назначению и их распечатке.

Ожидаемый годовой экономический эффект составил 3 054 640 тенге. Приложение окупится в первые 10 месяцев использования.

5 Безопасность жизнедеятельности

В данной дипломной работе анализируется разработка автоматизированной системы учета определения стоимости проектов. Основной сферой деятельности предприятия является оказание строительных услуг сторонним организациям и частным лицам. Чтобы обеспечить безопасность сотрудников был проведен анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал, а также выполнены расчеты обеспечивающие комфортные условия труда.

5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал.

В наше время люди все чаще переходят от компьютеров, к более удобным ноутбукам. В данной главе описаны основные опасные и вредные факторы при работе с ноутбуком. Ноутбук оказывает вредное воздействие на организм работающего человека. Профессия программиста связана прежде всего с работой на ноутбуке, следовательно, все свои услуги программист выполняет только на ноутбуке. Программист и его руководитель должны знать о опасном и вредном воздействии факторов, и об эффективных способах защиты от них, что уменьшает вероятность получения ими различных профессиональных заболеваний, а также снижает количество сбоев и ошибок в работе.

Основные опасные и вредные факторы, плохо воздействующие на здоровье человека при работе с ноутбуком:

- стесненная поза, сидячее положение в течение длительного времени;

- воздействие электромагнитного излучения;
- утомление глаз, нагрузка на зрение;
- перегрузка суставов кистей;
- стресс при потере информации;
- психические расстройства.
- повышенный уровень шума на рабочем месте;
- опасный уровень напряжения;
- недостаточная освещенность рабочей зоны;
- опасность возникновения пожара;

Рабочее помещение:

- рабочее помещение находится на 1 этаже;
- размер помещения 8×5×3 м. (длина -ширина-высота);
- светильник типа ЛД-50;
- одно окно с тройным остеклением, размер которого 2×3 м.;
- в помещении работают 2 служащих;
- рабочее помещение оборудовано техникой: ноутбуком, принтером, телефоном.

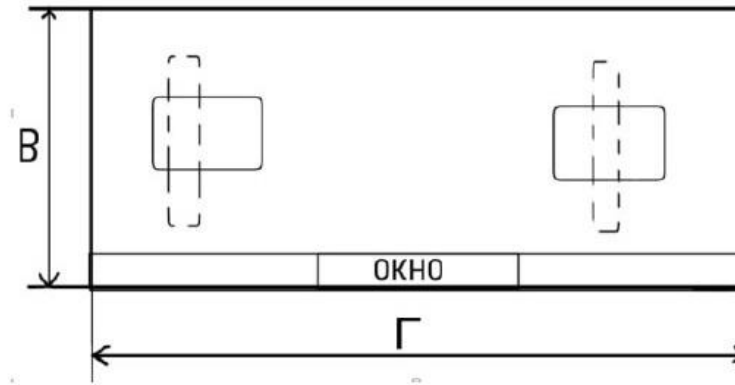


Рисунок 5.1 – План учебного помещения

Поскольку вся работа выполняется в помещении на ноутбуке с хорошей вентиляцией и системой шумоподавления, то дискомфорта с шумом быть не может. Вдобавок в помещении установлена кондиционирующая система, которая обеспечивает приток чистого и фильтрованного воздуха, создавая вентиляцию и нужную комфортную температуру. Из чего следует, что в офисе основная работа над дипломным проектом ведется без проблем с шумом и вентиляцией.

5.2 Параметры микроклимата

Микроклимат в офисном помещении, в основном, влияет на тепловое состояние организма человека и его теплообмен с окружающей средой. Несмотря на то, что параметры микроклимата офисных помещений могут значительно колебаться, температура тела человека остается постоянной (36,5 °С). Нормальное протекание физиологических процессов в организме возможно лишь тогда, когда выделяемое организмом тепло непрерывно отводится в окружающую среду. Отдача теплоты организмом человека во внешнюю среду происходит тремя основными способами: конвекцией, излучением и испарением [1].

Снижение температуры при всех других одинаковых условиях приводит к росту теплоотдачи путем конвекции и излучения и может привести к переохлаждению организма.

При высокой температуре практически все тепло, которое выделяется, отдается в окружающую среду испарением пота. Если микроклимат характеризуется не только высокой температурой, но и значительной влажностью воздуха, то пот не испаряется, а стекает каплями с поверхности кожи.

Недостаточная влажность приводит к интенсивному испарению влаги со слизистых оболочек, их пересыханию и эрозии, загрязнению болезнетворными микробами. Вода и соли, выделяемые из организма потом,

должны замещаться, поскольку их потеря приводит к сгущению крови и нарушению деятельности сердечно-сосудистой системы.

Повышение скорости движения воздуха способствует усилению процесса теплоотдачи конвекцией и испарением пота. Длительное влияние высокой температуры в сочетании со значительной влажностью может привести к накоплению тепла в организме и к гипертермии состоянию, при котором температура тела повышается до 38...40 °С.

При низкой температуре, значительной скорости и влажности воздуха возникает переохлаждение организма (гипотермия). Вследствие воздействия низких температур могут возникнуть холодцовые травмы. Параметры микроклимата оказывают также существенное влияние на производительность труда и травм человека [1].

Микроклимат в производственных помещениях определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха.

Таблица 5.2 – Параметры микроклимата в помещении

Период года	Величина микроклимата	Физическая величина
Холодный	Температура в помещении	22-24 °С
	Относительная влажность воздуха	40-60 %
	Скорость движения воздуха	0,2 м/с
Теплый	Температура в помещении	20-25 °С
	Относительная влажность воздуха	40-60 %
	Скорость движения воздуха	0,1 м/с

Допустимые параметры микроклимата установлены по критериям теплового и функционального состояния человека на период 8 – часовой рабочей смены. Они не наносят ущерб или нарушения здоровья персонала, не могут приводить к появлению общих и локальных ощущений теплового дискомфорта, ухудшению самочувствия и снижению работоспособности.

Для создания и автоматического поддержания в офисе независимо от наружных условий оптимальных значений температуры, влажности, чистоты и скорости движения воздуха, в холодное время года используется водяное отопление, в теплое время года применяется кондиционирование воздуха. Кондиционер представляет собой вентиляционную установку, которая с помощью приборов автоматического регулирования поддерживает в помещении заданные параметры воздушной среды.

5.3 Эргономические требования к рабочему месту

Расположение элементов на рабочих местах относится к числу важных проблем эргономического проектирования в области компьютерных технологий.

Если сотрудникам довольно часто приходится ходить по офису, и к ним часто приходят клиенты, следует организовывать рабочее пространство таким образом, чтобы его центр был свободным. В этом случае следует использовать угловые столы, которые располагаются в четных углах офиса, а также возле двух стен, где два угловых стола ставятся впритык друг к другу.

В офисе не должно быть длинных путей, все необходимые для работы материалы и приборы должны находиться под рукой, так чтобы не приходилось целый день бегать из одного угла в другой.

Слишком тесный офис означает дополнительный шум, то есть дополнительную нагрузку на мозг. Даже один системный блок компьютера создает достаточно много шума, который вроде и не замечаешь, пока работаешь. Но какое облегчение испытываешь, когда его выключаешь! Этот постоянный шумовой фон усиливает психическую усталость, человек устает быстрее, во второй половине дня ему уже становится сложно сосредоточиться.

Слишком мало места в офисе означает, что работнику придется сидеть в неудобном, стесненном положении без возможности движения и не имея достаточно места для работы. Минимум 58 см по длине стола должно отводиться для ног. Минимальный размер рабочей поверхности стола должен быть 1,28 м², только в исключительных случаях можно уменьшить ширину до 120 см.

Для свободного движения на рабочем месте должно быть отведено не менее 1,5 м², от кромки стола это пространство не должна быть менее одного метра. Если какая-то часть стола по ширине меньше чем 80 см, то это не относится к рабочей поверхности. Если стол имеет боковые контейнеры, то его длина должна быть не менее 120 см.

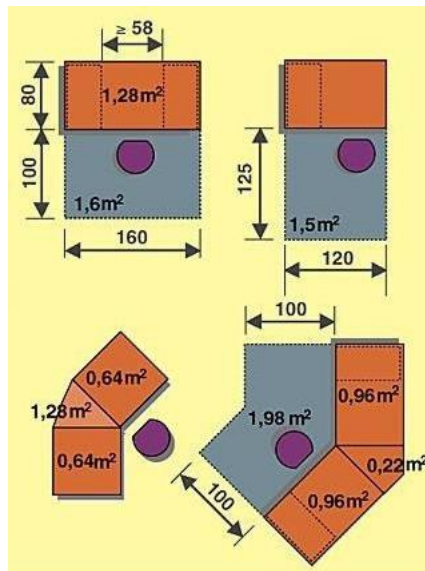


Рисунок 5.3 – Комфортное расположение рабочего места

Если в офисе планируется принимать клиентов с разными вопросами или разного уровня, то целесообразно можно будет разделить рабочее помещение на две зоны. Сделать это можно при помощи объединенных столов возле двух стен помещения. Помещение на передней части офиса может предназначаться для приема посетителей не высокого уровня, а внутренняя часть офиса, более подходящая для длительной работы с посетителями высшего уровня.

5.4 Расчет естественного освещения

В данном случае, под работой над проектом предполагается чтение, письмо, а также непосредственная работа с ноутбуком, которая имеет следующие недостатки:

- вероятность появления прямой блёскости;
- ухудшенная контрастность между изображением и фоном;
- отражение экрана.

Для начала делаем проверку площади боковых световых просветов здания, предназначенных для обеспечения необходимыми показателями освещенности рабочего помещения.

Характеристики помещения: длина $L = 8$ м, ширина $B = 5$ м, высота $H = 3$ м. Высота трудовой поверхности над уровнем пола $h_{рп} = 0,85$ м, окна начинаются с высоты $h_{нок} = 0,9$ м, высота окна $h_0 = 2$ м, ширина 3 м. Рабочее пространство находится в IV часовом поясе – в городе Алматы (пояс светового климата – IV северной широты и южнее).

Рабочее помещение находится на расстоянии $l_{рт} = 1$ м, от наружной стены здания, в которое встроены два оконных проема. Наименьшая освещенность находится на расстоянии 5 метров от оконного.

Общая площадь окон $S_0, м^2$, определяется по формулам 5.4.1 и 5.4.2

$$100 s_0 S_n = e_n \cdot K_3 \cdot \eta_0 \cdot 0r1, \quad (5.4.1)$$

$$S_0 = s_0 \cdot e_n \cdot K_3 \cdot \eta_0 \cdot 100 \cdot 0r1, \quad (5.4.2)$$

где S_n - площадь помещения; $S_n = 40 м^2$;

e_n - нормированное значение КЕО, выбираемое по таблице 1.1. Для высокой точности зрительных работ принимаем $e_n = 1,2$.

m_N - коэффициент светового климата, который находится по таблице 1.1 приложения. Учитывая заданный световой пояс, приняв ориентацию световых проемов на Север, определим по формуле 5.3, при $m_N = 0,75$

$$exIV = e_n \cdot m \cdot c, \quad (5.4.3)$$

где, $m = 0,65$;

$c = 0,75$ (в наружных стенах зданий);

$e_n = 1,2$ для работ высокой точности III разряда зрительной работы;

$$exIV = 0,65 \cdot 0,75 \cdot 1,2 = 0,585$$

Учитываем класс здания и находим коэффициент 5,3 $k_3 = 1,2$ (учебные помещения, лаборатории, конструкторские бюро).

τ_0 - общий коэффициент светопропускания, равный

$$\tau_0 = \tau_1 \times \tau_2 \times \tau_3 \times \tau_4, \quad (5.4.4)$$

где, $\tau_1 = 0,5$ (пустотелые стеклянные блоки);

$\tau_2 = 0,6$ (деревянные двойные отдельные переплеты);

$\tau_3 = 0,8$ (железобетонные фермы и арки);

$\tau_4 = 1$ (регулируемые жалюзи и шторы);

η_0 – световая характеристика окон.

$$\tau_0 = 0,5 \cdot 0,6 \cdot 0,8 \cdot 1 = 0,24$$

Определим по формуле 5.4.5

$$L = B - 1, \quad (5.4.5)$$

$$h_{расч} = h_{но} + h_0 - h_{рт}$$

$$L = 3 - 1 = 2 \text{ м,}$$

$$L1 = LB-1 = 42 = 3 \cdot 14$$

$$h_{\text{расч}} = 0,75 + 2 - 0,75 = 2$$

$$V_{\text{расч}} = 31,575 = 1,9 \cdot 16,625$$

По таблице 5.2 определим $\eta_0 = 9$.

r_1 – коэффициент, учитывающий повышение КЕО при боковом освещении благодаря свету, отраженному от поверхностей помещения и подстилающего слоя.

Средний коэффициент отражения в здании $\rho_{\text{ср}} = 0,5$, принимаем одностороннее боковое освещение. Определяем значение r_1 . Принимаем r_1 равным 1,05.

$k_{\text{зд}}$ – коэффициент, учитывающий затенение окон противостоящими зданиями.

Поскольку затеняющих зданий поблизости нет, $k_{\text{зд}} = 1$. Проведем подсчет общей площади окон:

$$S_0 = 40 \cdot 0,585 \cdot 9 \cdot 1 \cdot 1,20,24 \cdot 1,05 \cdot 100 = 12,34 \text{ м}^2$$

Площадь световых проемов равна $S_{\text{сп}} = 12,34 \text{ м}^2$.

В проделанных расчетах была посчитана площадь боковых световых проемов помещения, необходимой для создания условий нормируемой освещенности на рабочих местах. Для здания размерами 8 x 5 x 3 м³ значение для площади одного светового проема получилось равным 12,34 м², где высота окна $h_{\text{ок}} = 2$ м, а длина окна $l_{\text{ок}} = 3$ м. В конечном расчете, фактический размер светового проема соответствует нормативным значениям.

5.5 Расчет искусственного освещения

В связи с тем, что естественное освещение слабое, на рабочем помещении, то должно применяться также искусственное освещение. В данном разделе будет произведен расчет искусственного освещения.

Определим по формуле 5.5 размещение светильников:

$$h_{\text{п}} = H - h_{\text{с}}, \quad (5.5.1)$$

$$h = h_{\text{п}} - h_{\text{р}}, \quad (5.5.2)$$

где, $H = 3$ м. - высота помещения;

$h_{\text{с}} = 0,25$ м. - расстояние светильников от перекрытия;

$h_{\text{п}}$ - высота светильников над полом;

$h_p = 0,7$ м. - высота расчетной поверхности для помещений, связанной с работой ноутбука;
 h - расчетная высота.

$$h_{\text{п}} = 3 - 0,25 = 2,75 \text{ м.}$$

$$h_{\text{п}} = 2,75 - 0,7 = 2,05 \text{ м.}$$

Светильник типа ЛДР (2x40 Вт). Длина которого 1,24 м, ширина 0,27 м, а высота 0,10 м.

L - расстояние между соседними светильниками (рядами люминесцентных светильников);

L_a (по длине помещения) = 1,76 м;

L_b (по ширине помещения) = 3 м;

l - расстояние от крайних светильников или рядов светильников до стены,

$$l = 0,3 - 0,5L.$$

$$la = 0,5L_a, lb = 0,3L_b$$

$$la = 0,88 \text{ м.}, lb = 0,73 \text{ м.}$$

Светильники с люминесцентными лампами в помещениях для работы рекомендуют устанавливать рядами. Метод коэффициента использования светового потока предназначен для расчета общего равномерного освещения горизонтальных поверхностей при отсутствии крупных затемняющих предметов.

По формуле 5.8 мы определяем потребный поток ламп в каждом светильнике:

$$\Phi = \frac{E \cdot r \cdot S \cdot z}{N \cdot h} \quad (5.5.3)$$

где E - заданная минимальная освещенность = 300 лк., так как. разряд зрительных работ равен 3.

r - коэффициент запаса = 1,3 (для помещений, связанных с работой ПК)

S - освещаемая площадь = 30 м².

z - характеризует неравномерное освещение,

$z = E_{\text{ср}} / E_{\text{min}}$ - зависит от отношения

$$l = L/h,$$

$$la = L_a/h = 0,6,$$

$lb = L_b/h = 1,5$, если l превышают допустимых значений, то $z=1,1$ (для люминесцентных ламп).

N - число светильников. Первоначально намечается число рядов n , которое подставляется вместо N . Тогда Φ - поток ламп одного ряда.

$$N = \frac{\Phi}{\Phi_1} \quad (5.5.4)$$

где Φ_1 - поток ламп в каждом светильнике.

h - коэффициент использования. Для его нахождения выбирают индекс помещения i и оцениваются коэффициенты отражения поверхностей помещения:

$r_{\text{пот.}}$ (потолка) = 70%,

$r_{\text{ст.}}$ (стены) = 50%,

$r_{\text{р.}}$ (пола) = 30%.

$$\Phi = \frac{300 \cdot 1,3 \cdot 25 \cdot 1,1}{2 \cdot 0,3} = 21450 \text{ лм.}$$

На основе расчетов была проведена реконструкция искусственного освещения, так как ранее установленные три лампы ЛДР – 2x40 с световым потоком не обеспечивали достаточно количества света в рабочем помещении с размерами 8x5x3. В результате реконструкции количество ламп было увеличено, что в результате стало выполнять норму освещенности.

5.6 Расчет системы кондиционирования помещения

В данном разделе приведены расчеты системы кондиционирования в офисном помещении. Кондиционирование обеспечивает наилучшее состояние микроклимата в рабочем помещении и условия работы.

Для начала рассчитаем расход воздуха (L , [м³/час]) по формуле [2]:

$$L = \frac{Q_{\text{ИЗБ}} \times 860}{C \times \Delta t \times \gamma}, L = \frac{Q_{\text{ИЗБ}} \times 860}{C \times \Delta t \times \gamma}, \quad (5.6.1)$$

где $C=0,24$ ккал/(кг x °C) – теплоемкость воздуха,

$\gamma=1,206$ кг/м³ – удельная масса приточного воздуха/

$$Q_{\text{ИЗБ}} = Q_{\text{ОБ}} + Q_{\text{Л}} + Q_{\text{ОВ}} + Q_{\text{РАД}} - Q_{\text{ОГР}}. \quad (5.6.2)$$

1) $Q_{\text{ОБ}} = P_{\text{ОБ}} \times \eta \times n_{\text{ОБ}} = 10 \times 0,5 \times 0,95 = 4,75$, кВт – количество тепла от оборудования.

2) $Q_{OV} = \eta \times N_{OV} \times F = 0,92 \times 40 \times 10 \times 4 = 1,47$, кВт – количество тепла от осветительных устройств.

F – площадь помещения (m^2).

3) $Q_L = q_{\text{я}} \times n$ – количество тепла от людей.

$q_{\text{я}}$ – тепловыделение явное (Вт) (таблица 8).

$q_{\text{я}}$ женщин составляет 85% $q_{\text{я}}$ мужчин.

Лето: $Q_L = q_{\text{я}} \times n = 4 \times 63 + 3 \times 63 \times 0.85 = 0.41$, кВт,

Зима: $Q_L = q_{\text{я}} \times n = 4 \times 92 + 3 \times 92 \times 0.85 = 0.6$, кВт.

4) $Q_{РАД}$ – количество тепла от солнечного излучения.

$$Q_{РАД} = m \times F_0 \times (q^I + q^{II}) \times \beta_{СЗ} \times k_1 \times k_2, \quad (5.6.3)$$

где, m – количество окон,

F_0 – площадь окна,

q^I, q^{II} – тепловые потоки от прямой и рассеянной радиации (таблица 5) ($Вт/m^2$),

$\beta_{СЗ}$ – коэффициент теплопропускания (таблица 4),

k_1 – коэффициент затемнения остекления (таблица 6),

k_2 – коэффициент загрязнения стекла (таблица 7).

$Q_{РАД} = 3 \times 3 \times (214 + 79 + 73 + 77) \times 0.15 \times 0.95 \times 1.28 = 0.73$, кВт.

5) $Q_{ОГР}$ – теплопоступления и теплопотери в результате разности температур.

$Q_{ОГР.ЛЕТО} = 0$, кВт.

$Q_{ОГР.ЗИМА} = X_0 \times V_{ПОМ} \times (t_{Нрасч} - t_{Врасч})$,

где, $V_{ПОМ}$ – объем помещения (m^3),

$X_0 = 0.42$ Вт/($m^3 \times ^\circ C$) – удельная тепловая характеристика,

$t_{Нрасч}, t_{Врасч}$ – выбираем из приложения 8 по параметру Б для города

Алматы.

$Q_{ОГР.ЗИМА} = 0.42 \times 128 \times (31.2 - 25) = 0.33$, кВт.

6) Составим тепловой баланс помещения.

Лето: $Q_{ИЗБ} = 4.75 + 0.41 + 1.47 + 0.73 - 0 = 7.36$, кВт.

Зима: $Q_{ИЗБ} = 4.75 + 0.6 + 1.46 + 0.73 - 0.33 = 7.21$, кВт.

Так как тепловой баланс для лета больше зимнего теплового баланса, то рассчитаем тепло напряженность воздуха по формуле [2]:

$$Q_H = \frac{Q_{ИЗБ.ЛЕТО} \times 860}{V_{ПОМ}} = \frac{7,36 \times 860}{128} = 49,5, \text{ ккал}/m^3.$$

При $Q_H > 20 \text{ ккал/м}^3$, $\Delta t = 8 \text{ }^\circ\text{C}$,

при $Q_H < 20 \text{ ккал/м}^3$, $\Delta t = 6 \text{ }^\circ\text{C}$.

8) Определение количества воздуха, необходимое для поступления в помещение:

$$L = \frac{Q_{\text{изб}} \times 860}{C \times \Delta t \times \gamma} = \frac{7.36 \times 860}{0.24 \times 8 \times 1.206} = 2733 \text{ м}^3/\text{час.}$$

9) Определение кратности воздухообмена:

$$n = \frac{L}{V_{\text{пом}}} = \frac{2733}{128} = 21,35 \text{ час}^{-1}.$$

Таблица 5.6 Характеристики кондиционера SUA 0501

Модель с верхней подачей SUA			0501
Электропитание		В/Ф/Гц	400/3/50+N
Расход воздуха	Макс.	м ³ /ч	4720
	Мин.	м ³ /ч	2950
Максимальная холодопроизводительность	(24 °С, 50%)	кВт	16,7
	(26 °С, 50%)	кВт	17,7
Мощность компрессора		кВт	4,5
Электронагреватель		кВт	6,6
Увлажнитель		кВт	2,3
Расход пара		кг/ч	3,0
Выносной стандартный конденсатор малошумящий		CAL	0511
		CAL	0661
Размеры	Высота	мм	1740
	Ширина	мм	1200
	Глубина	мм	450
Масса		кг	260
Соединительные линии	Газовая	мм	16
	Жидкостная	мм	16
	Дренажная	мм	21

С учетом полученных данных был выбран кондиционер SUA 0501 с верхней подачей воздуха.

Заключение

В ходе выполнения дипломного проекта была изучена работа предприятия, оказывающего строительные и подрядные услуги сторонним организациям.

В результате была разработана автоматизированная система учета. Автоматизированная система поддерживает все операции, связанные с учетом, обработкой и предоставлением необходимых данных. Данная система обеспечивает ввод, удаление, хранение и редактирование информации, которая содержится в таблицах данных.

Результатами разработки являются:

- информационное обеспечение ИС в формате СУБД Access;
- пользовательский интерфейс ИС, включающий экранные формы для работы со справочными, оперативными данными и отчетными данными;
- формы документов и отчетной документации для автоматизированной подготовки на рабочем месте требуемой информации;
- запросы, обеспечивающие формирование данных для первичных и отчетных документов;
- руководство пользователя.

Так же была рассмотрена существующая на предприятии нормативно-справочная и результатная информация, разработана информационно-логическая модель данных, обеспечивающая оперативный и количественно-суммовой учет клиентов, материалов, работ и заказов.

Реализация системы проводилась с использованием инструментальных средств Borland Delphi 7.0. в сочетании с СУБД MS ACCESS 2013. При написании программы основное внимание было уделено удобству работы пользователя и построению дружественного интерфейса.

Список литературы

- 1 Климова, Л. М. Delphi 7. Основы программирования. Решение типовых задач. Самоучитель / Л.М. Климова. – М.: 2017. – 480 с.
- 2 Культин, Программирование на Object Pascal / Н. Культин. – М.: БХВ-Петербург, 2012. – 528 с.
- 3 Бекаревич, Ю.Б. Создание реляционной базы данных и запросов. MS Access 2013: учебное пособие / Ю.Б. Бекаревич, Н.В. Пушкина. – Санкт-Петербург: СПбГУЭФ, 2012. – 87 с.
- 4 Осипов, Дмитрий Delphi. Профессиональное программирование / Дмитрий Осипов. – М.: Символ-плюс, 2013. – 820с.
- 5 Ревич, Ю. Нестандартные приемы программирования на Delphi / Ю. Ревич. – М.: БХВ-Петербург, 2016. – 560 с.
- 6 Одиноккина, С.В. Разработка баз данных в Microsoft Access 2013 / Одиноккина С.В. – Санкт-Петербург: НИУ ИТМО, 2012. – 83 с.
- 7 Санников, Е. В. Курс практического программирования в Delphi. Объектно-ориентированное программирование / Е.В. Санников. – М.: Солон-Пресс, 2013. – 188 с.
- 8 Г.Ш. Боканова. Методические указания по выполнению экономической части дипломных работ для студентов специальностей 5В070400 – Вычислительная техника и программное обеспечение, 5В070300 – «Информационные системы», 5В060200 – «Информатика». – Алматы: АУЭС, 2020 – 35 с.
- 9 Г. Боканова Методические указания по выполнению экономической части дипломных работ Алматы, АУЭС, 2020 – 35с.
- 10 Техничко-экономическое обоснование дипломных проектов Брест, БГТу, 2014 – 15с.
- 11 Методические указания к выполнению расчетно-графической работы для студентов специальности 080801 «Прикладная информатика (по областям)» Уфа, 2010 – 12с.
- 12 Методические указания по выполнению экономической части дипломных работ Москва, Московский университет им. С.Ю. Витте, 2016 – 21с.
- 13 Симионов Ю.Ф., Боромотов В.В. Информационный менеджмент. — Ростов н.Д: Феникс, 2013 – 250с.
- 14 Microsoft Excel для чайников, ГрегХарвей. Диалектика / Вильямс 2013 г. – 368 с.
- 15 Гетия И.Г., Шумилин В.К., Леонтьева И.Н. и др. Экология компьютерной техники. – Учебное пособие. – М.: МГУПИ, 2010г. – 69 с.
- 16 Excel 2013.Профессиональное программирование на VBA. Джон Уокенбах 2014 г. – 960 с.
- 17 ГОСТ 12.2.032 – 78. ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.

18 Гетия И.Г. Безопасность при работе на ПЭВМ. – М.: МГАПИ, 2009 – 73с.

19 Санатова Т.С. Безопасность жизнедеятельности. – Учебное пособие для всех специальностей, 2017. – 407 с.

20 Абдимуратов Ж. С., Мананбаева С. Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавр – Алматы: АИЭС, 2009 – 20 с.

21 С.Е. Мананбаева, А.С. Бегимбетова Охрана труда. Методические указания к выполнению расчетно-графических работ для студентов – бакалавров всех специальностей. – Алматы: АУЭС, 2013 – 17 с.

Приложение А

Листинг программы

```
program SMETA;

uses
  Forms,
  UnMain in 'UnMain.pas' {FmMenu},
  UnDM in 'UnDM.pas' {DM: TDataModule},
  UnSWork in 'UnSWork.pas' {FmSWork},
  UnInch in 'UnInch.pas' {FmInch},
  UnSMat in 'UnSMat.pas' {FmSMat},
  UnRekv in 'UnRekv.pas' {FmRekv},
  UnClient in 'UnClient.pas' {FmClient},
  UnObject in 'UnObject.pas' {FmObject},
  UnOrder in 'UnOrder.pas' {FmOrder},
  UnUsedMat in 'UnUsedMat.pas' {FmUsedMat},
  UnBounds in 'UnBounds.pas' {FmBounds};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TFmMenu, FmMenu);
  Application.CreateForm(TDM, DM);
  Application.CreateForm(TFmSWork, FmSWork);
  Application.CreateForm(TFmInch, FmInch);
  Application.CreateForm(TFmSMat, FmSMat);
  Application.CreateForm(TFmRekv, FmRekv);
  Application.CreateForm(TFmClient, FmClient);
  Application.CreateForm(TFmObject, FmObject);
  Application.CreateForm(TFmOrder, FmOrder);
  Application.CreateForm(TFmUsedMat, FmUsedMat);
  Application.CreateForm(TFmBounds, FmBounds);
  Application.Run;
end.

unit UnBounds;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, Buttons, Grids, DBGrids, ComObj;

type
  TFmBounds = class(TForm)
    DateTimePicker1: TDateTimePicker;

    DateTimePicker2: TDateTimePicker;
    Label1: TLabel;
```


Продолжение приложения А

```
Label2: TLabel;  
Label3: TLabel;  
BitBtn1: TBitBtn;  
procedure BitBtn1Click(Sender: TObject);  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;  
  
var  
  FmBounds: TFmBounds;  
  
implementation  
  
uses UnDM;  
  
{ $R *.dfm }  
  
procedure TFmBounds.BitBtn1Click(Sender: TObject);  
var Excel: Variant;  
    WorkbookName : string;  
    PosInc, i : Integer;  
    sum : real;  
begin  
  if FmBounds.Caption = 'Расход по материалам за период' then  
    begin  
  
      DM.QRMat.Active := false;  
      DM.QRMat.Parameters[0].Value := DateTimePicker1.Date;  
      DM.QRMat.Parameters[1].Value := DateTimePicker2.Date;  
      DM.QRMat.Active := true;  
  
      WorkbookName := GetCurrentDir + '\Расход материалов.xlt';  
      Excel := CreateOleObject('Excel.Application');  
  
      try  
        Excel.Workbooks.Open(WorkbookName);  
        //Экспорт данных из таблиц БД:  
        Excel.WorkSheets[3].Select;  
  
        PosInc := 4; //Начальная позиция  
        sum := 0;  
        Excel.Cells[2,3] := DateToStr(DateTimePicker1.Date);  
        Excel.Cells[2,5] := DateToStr(DateTimePicker2.Date);  
        DM.QRmat.First;  
        for i:=0 to DM.QRmat.RecordCount-1 do  
          begin  
            if i <> DM.QRmat.RecordCount-1 then  
              begin
```

Продолжение приложения А

```
Excel.Rows.Item[PosInc+1].Select;
Excel.Selection.Insert;
Excel.Rows.Item[PosInc].Select;
Excel.Selection.Copy;
Excel.Rows.Item[PosInc+1].Select;
Excel.ActiveSheet.Paste;
Excel.Application.CutCopyMode := False;
end;
//Вывод данных в Excel
Excel.Cells[PosInc,1] := IntToStr(DM.QRMAT.RecNo);

Excel.Cells[PosInc,2] := DM.QRMAT.FieldByName('Наименование
материала').AsString;
Excel.Cells[PosInc,3] := DM.QRMAT.FieldByName('Ед изм').AsString;
Excel.Cells[PosInc,4] := FloatToStr(DM.QRMAT.FieldByName('Стоимость
единицы').AsFloat);
Excel.Cells[PosInc,5] := FloatToStr(DM.QRMAT.FieldByName('Расход').AsFloat);

sum := sum + DM.QRMAT.FieldByName('расход').AsFloat;
PosInc := PosInc + 1;

DM.QRMAT.Next;

end; //for i
Excel.Cells[PosInc,4] := 'Итого';
Excel.Cells[PosInc,5] := FloatToStr(sum);

Excel.WorkSheets[3].Select;
Excel.Visible := True;
except
  Excel.Quit;
end;
end;

if FmBounds.Caption = 'Расходы по объектам за период' then
begin

  DM.QObW.Active := false;
  DM.QObW.Parameters[0].Value := DateTimePicker1.Date;
  DM.QObW.Parameters[1].Value := DateTimePicker2.Date;
  DM.QObW.Active := true;

  WorkbookName := GetCurrentDir + '\Расходы по объектам.xlt';
  Excel := CreateOleObject('Excel.Application');
try
  Excel.Workbooks.Open(WorkbookName);
  //Экспорт данных из таблиц БД:
  Excel.WorkSheets[3].Select;

  PosInc := 4; //Начальная позиция
```

Продолжение приложения А

```
sum := 0;
Excel.Cells[2,4] := DateToStr(DateTimePicker1.Date);
Excel.Cells[2,6] := DateToStr(DateTimePicker2.Date);
DM.QObW.First;
for i:=0 to DM.QObW.RecordCount-1 do
begin
  if i <> DM.QObW.RecordCount-1 then
  begin

    Excel.Rows.Item[PosInc+1].Select;
    Excel.Selection.Insert;
    Excel.Rows.Item[PosInc].Select;
    Excel.Selection.Copy;
    Excel.Rows.Item[PosInc+1].Select;
    Excel.ActiveSheet.Paste;
    Excel.Application.CutCopyMode := False;
  end;
  //Вывод данных в Excel
  Excel.Cells[PosInc,1] := IntToStr(DM.QObW.RecNo);

  Excel.Cells[PosInc,2] := DM.QObW.FieldByName('Дата').AsString;
  Excel.Cells[PosInc,3] := DM.QObW.FieldByName('Наименование').AsString + '
'+DM.QObW.FieldByName('Адрес').AsString + '
'+DM.QObW.FieldByName('Телефон').AsString;
  Excel.Cells[PosInc,4] := DM.QObW.FieldByName('Название объекта').AsString + '
'+DM.QObW.FieldByName('Характеристика').AsString;
  Excel.Cells[PosInc,5] :=
FloatToStr(DM.QObW.FieldByName('работа_расход').AsFloat);

  sum := sum + DM.QObW.FieldByName('работа_расход').AsFloat;
  PosInc := PosInc +1;

  DM.QObW.Next;

end; //for i
Excel.Cells[PosInc,4] := 'Итого';
Excel.Cells[PosInc,5] := FloatToStr(sum);

//Формируем расход по материалам

DM.QObM.Active := false;

DM.QObM.Parameters[0].Value := DateTimePicker1.Date;
DM.QObM.Parameters[1].Value := DateTimePicker2.Date;
DM.QObM.Active := true;
PosInc := 4; //Начальная позиция
sum := 0;
DM.QObM.First;
for i:=0 to DM.QObM.RecordCount-1 do
```

Продолжение приложения А

```
begin

    //Вывод данных в Excel

    Excel.Cells[PosInc,6] :=
FloatToStr(DM.QObM.FieldName('материал_расход').AsFloat);

    sum := sum + DM.QObM.FieldName('материал_расход').AsFloat;
    PosInc := PosInc +1;

    DM.QObM.Next;

end; //for i
Excel.Cells[PosInc,6] := FloatToStr(sum);

//Excel.WorkSheets[3].Select;
Excel.Visible := True;
except
    Excel.Quit;
end;
end;
end.

unit UnClient;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, Mask, DBCtrls, Grids, DBGrids, ComCtrls, StdCtrls, Buttons,
    ExtCtrls, DB;

type
TFmClient = class(TForm)
    Panel1: TPanel;
    Panel3: TPanel;
    Label15: TLabel;
    DBNavigator1: TDBNavigator;
    RadioGroup1: TRadioGroup;
    EditFind: TEdit;
    DBGrid1: TDBGrid;
    Panel2: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
```

Продолжение приложения А

```
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label14: TLabel;
DBEdit1: TDBEdit;
DBEdit2: TDBEdit;
DBEdit3: TDBEdit;
DBEdit4: TDBEdit;
DBEdit5: TDBEdit;
DBEdit6: TDBEdit;
DBEdit7: TDBEdit;
DBEdit8: TDBEdit;
DBEdit9: TDBEdit;
DBEdit10: TDBEdit;
DBEdit11: TDBEdit;
DBEdit12: TDBEdit;
DBEdit14: TDBEdit;
BitBtn1: TBitBtn;
procedure EditFindChange(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FmClient: TFmClient;

implementation
uses UnDM;
{$R *.dfm}

procedure TFmClient.EditFindChange(Sender: TObject);
begin
  Case RadioGroup1.ItemIndex of //Выполняем поиск в зависимости от положения
  переключателя
    0 : DM.TbClient.Locate('Наименование',EditFind.Text,[loCaseInsensitive, loPartialKey]
  );
    1 : DM.TbClient.Locate('ИНН',EditFind.Text,[loCaseInsensitive, loPartialKey] );
    2 : DM.TbClient.Locate('КПП',EditFind.Text,[loCaseInsensitive, loPartialKey] );
  end;
end;

procedure TFmClient.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  DM.TbClient.Active := false;
  DM.TbClient.Active := true
```

Продолжение приложения А

```
end;  
end.
```

```
unit UnDM;
```

```
interface
```

```
uses
```

```
  SysUtils, Classes, DB, ADODB;
```

```
type
```

```
  TDM = class(TDataModule)  
    ADOC: TADOConnection;  
    TbRekv: TADOTable;  
    TbRekvID: TAutoIncField;  
    TbRekvDSDesigner: TWideStringField;  
    TbRekvDSDesigner2: TWideStringField;  
    TbRekvDSDesigner3: TWideStringField;  
    TbRekvDSDesigner4: TWideStringField;  
    TbRekvDSDesigner5: TWideStringField;  
    TbRekvDSDesigner6: TWideStringField;  
    TbRekvDSDesigner7: TWideStringField;  
    TbRekvDSDesigner8: TWideStringField;  
    TbRekvDSDesigner9: TWideStringField;  
    TbRekvDSDesigner10: TWideStringField;  
    TbRekvDSDesigner11: TWideStringField;  
    TbRekvWeb: TWideStringField;  
    TbRekvDSDesigner12: TWideStringField;  
    DsRekv: TDataSource;  
    DSObject: TDataSource;  
    DSSMat: TDataSource;  
    DSInch: TDataSource;  
    DSClient: TDataSource;  
    DSSWork: TDataSource;  
    DSOrder: TDataSource;  
    TbSWork: TADOTable;  
    TbInch: TADOTable;  
    TbSMat: TADOTable;  
    TbClient: TADOTable;  
    TbSWorkID: TAutoIncField;  
    TbSWorkDSDesigner: TWideStringField;  
    TbSWorkDSDesigner2: TIntegerField;  
    TbSWorkDSDesigner3: TBCDField;  
    TbInchID: TAutoIncField;  
    TbInchDSDesigner: TWideStringField;  
    TbInchDSDesigner2: TWideStringField;  
    TbSWorkField: TStringField;  
    TbSMatID: TAutoIncField;  
    TbSMatDSDesigner: TWideStringField;  
    TbSMatDSDesigner2: TIntegerField;
```

Продолжение приложения А

TbSMatDSDesigner3: TBCDField;
TbSMatField: TStringField;
TbObject: TADOTable;
TbOrder: TADOTable;
TbClientID: TAutoIncField;
TbClientDSDesigner: TWideStringField;
TbClientDSDesigner2: TWideStringField;
TbClientDSDesigner3: TWideStringField;
TbClientDSDesigner4: TWideStringField;
TbClientDSDesigner5: TWideStringField;
TbClientDSDesigner6: TWideStringField;
TbClientDSDesigner7: TWideStringField;
TbClientDSDesigner8: TWideStringField;
TbClientDSDesigner9: TWideStringField;
TbClientDSDesigner10: TWideStringField;
TbClientDSDesigner11: TWideStringField;
TbClientWeb: TWideStringField;
TbClientDSDesigner12: TWideStringField;
TbObjectID: TAutoIncField;
TbObjectDSDesigner: TWideStringField;
TbObjectDSDesigner2: TWideStringField;
TbWork: TADOTable;
TbOrderDSDesigner: TAutoIncField;
TbOrder_: TIntegerField;
TbOrder_2: TIntegerField;
TbOrderDSDesigner2: TDateTimeField;
TbOrderField: TStringField;
TbOrderField2: TStringField;
TbMat: TADOTable;
TbMatID: TAutoIncField;
TbMat_: TIntegerField;
TbMatDSDesigner2: TFloatField;
TbWorkID: TAutoIncField;
TbWork_: TIntegerField;
TbWorkDSDesigner2: TFloatField;
TbMatField: TStringField;
TbWork_2: TIntegerField;
TbWorkStringField: TStringField;
DSWork: TDataSource;
DSMat: TDataSource;
TbMatID_: TIntegerField;
QGetWork: TADOQuery;
QGetMat: TADOQuery;
QGetClient: TADOQuery;
QGetObject: TADOQuery;
QGetClientID: TAutoIncField;
QGetClientDSDesigner: TWideStringField;
QGetClientDSDesigner2: TWideStringField;
QGetClientDSDesigner3: TWideStringField;
QGetClientDSDesigner4: TWideStringField;

Продолжение приложения А

```
QGetClientDSDesigner6: TWideStringField;  
QGetClientDSDesigner7: TWideStringField;  
QGetClientDSDesigner8: TWideStringField;  
QGetClientDSDesigner9: TWideStringField;  
QGetClientDSDesigner10: TWideStringField;  
QGetClientDSDesigner11: TWideStringField;  
QGetClientWeb: TWideStringField;  
QGetClientDSDesigner12: TWideStringField;  
QGetObjectID: TAutoIncField;  
QGetObjectDSDesigner: TWideStringField;  
QGetObjectDSDesigner2: TWideStringField;  
QGetWorkDSDesigner: TWideStringField;  
QGetWorkDSDesigner2: TWideStringField;  
QGetWorkDSDesigner3: TBCDField;  
QGetMatDSDesigner: TWideStringField;  
QGetMatDSDesigner2: TWideStringField;  
QGetMatDSDesigner3: TBCDField;  
QGetMat_: TIntegerField;  
QRMat: TADOQuery;  
QRMatDSDesigner: TWideStringField;  
QRMatDSDesigner2: TWideStringField;  
QRMatDSDesigner3: TFloatField;  
QRMatDSDesigner4: TBCDField;  
QObW: TADOQuery;  
QObM: TADOQuery;  
QObWDSDesigner: TWideStringField;  
QObWDSDesigner2: TWideStringField;  
QObW_: TFloatField;  
QObWDSDesigner3: TDateTimeField;  
QObWDSDesigner4: TWideStringField;  
QObWDSDesigner5: TWideStringField;  
QObWDSDesigner6: TWideStringField;  
QObMDSDesigner: TWideStringField;  
QObMDSDesigner2: TWideStringField;  
QObM_: TFloatField;  
QObM_2: TFloatField;  
QObMDSDesigner3: TDateTimeField;  
QObMDSDesigner4: TWideStringField;  
QObMDSDesigner5: TWideStringField;  
QObMDSDesigner6: TWideStringField;  
TbWorkField: TStringField;  
TbMatField2: TStringField;  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;  
  
var  
  DM: TDM;
```


Продолжение приложения А

```
implementation
{$R *.dfm}

end.

unit UnInch;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DBCtrls, StdCtrls, Buttons, ExtCtrls, Grids, DBGrids, DB, ADODB;

type
  TFmInch = class(TForm)
    DBGrid1: TDBGrid;
    Panel1: TPanel;
    Label1: TLabel;
    EditFind: TEdit;
    BitBtn1: TBitBtn;
    DBNavigator1: TDBNavigator;
    Panel2: TPanel;
    procedure EditFindChange(Sender: TObject);
    procedure DBGrid1TitleClick(Column: TColumn);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  FmInch: TFmInch;

implementation
uses UnDM;
{$R *.dfm}

procedure TFmInch.EditFindChange(Sender: TObject);
begin
  DM.TbInch.Locate('Полное наименование',
  EditFind.Text,[loCaseInsensitive,loPartialKey]);
  //Поиск по частичному совпадению без учета регистра
end;

procedure TFmInch.DBGrid1TitleClick(Column: TColumn);
begin
  if Column.Grid.DataSource.DataSet is TADOTable then
    //if Column.FieldName <> 'Единица измерения' then
begin
```

Продолжение приложения А

```
if TADODataSet(Column.Grid.DataSource.DataSet).Sort =
  Column.FieldName then
  // Не работает с просмотревыми полями и датой
  TADODataSet(Column.Grid.DataSource.DataSet).Sort :=
  Column.FieldName + ' DESC'
else
  TADODataSet(Column.Grid.DataSource.DataSet).Sort :=
  Column.FieldName;
end;
end;

procedure TFmInch.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  DM.TbInch.Active := false;
  DM.TbInch.Active := true;
end;
end.

unit UnMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ActnList, ImgList, ActnMan, ActnCtrls, ToolWin, ActnMenus,
  StdCtrls, Buttons, ExtCtrls, jpeg;

type
  TFmMenu = class(TForm)
    ActionMainMenuBar1: TActionMainMenuBar;
    ActionToolBar1: TActionToolBar;
    ActionManager1: TActionManager;
    ImageList1: TImageList;
    ActExit: TAction;
    ActClient: TAction;
    ActObject: TAction;
    ActSWork: TAction;
    ActMat: TAction;
    ActRekv: TAction;
    ActInch: TAction;
    ActOrder: TAction;
    ActRMat: TAction;
    ActRObj: TAction;
    Image1: TImage;
    Image2: TImage;
    Image3: TImage;
    Image4: TImage;
    procedure ActExitExecute(Sender: TObject);
    procedure ActClientExecute(Sender: TObject);
    procedure ActObjectExecute(Sender: TObject);
```

Продолжение приложения А

```
procedure ActSWorkExecute(Sender: TObject);
procedure ActMatExecute(Sender: TObject);
procedure ActRekvExecute(Sender: TObject);
procedure ActInchExecute(Sender: TObject);
procedure ActOrderExecute(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure ActRMatExecute(Sender: TObject);
procedure ActRObjExecute(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FmMenu: TFmMenu;

implementation
uses UnDM, UnObject, UnSWork, UnSMat, UnRekv, UnInch, UnClient, UnOrder, UnBounds;
{$R *.dfm}

procedure TFmMenu.ActExitExecute(Sender: TObject);
begin
  close;
end;

procedure TFmMenu.ActClientExecute(Sender: TObject);
begin
  FmClient.ShowModal;
end;

procedure TFmMenu.ActObjectExecute(Sender: TObject);
begin
  FmObject.ShowModal;
end;

procedure TFmMenu.ActSWorkExecute(Sender: TObject);
begin
  FmSWork.ShowModal;
end;

procedure TFmMenu.ActMatExecute(Sender: TObject);
begin
  FmSMat.ShowModal;
end;

procedure TFmMenu.ActRekvExecute(Sender: TObject);
begin
  FmRekv.ShowModal;
end;
```

Продолжение приложения А

```
procedure TFmMenu.ActInchExecute(Sender: TObject);
begin
  FmInch.ShowModal;
end;

procedure TFmMenu.ActOrderExecute(Sender: TObject);
begin
  FmOrder.ShowModal;
end;

procedure TFmMenu.BitBtn1Click(Sender: TObject);
begin
  FmBounds.ShowModal;
end;

procedure TFmMenu.ActRMatExecute(Sender: TObject);
begin
  FmBounds.Caption := 'Расход по материалам за период';
  FmBounds.ShowModal;

end;

procedure TFmMenu.ActRObjExecute(Sender: TObject);
begin
  FmBounds.Caption := 'Расходы по объектам за период';
  FmBounds.ShowModal;
end;
end.

unit UnObject;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DBCtrls, StdCtrls, Buttons, ExtCtrls, Grids, DBGrids, DB, ADODB,
  Mask;

type
  TFmObject = class(TForm)
    DBGrid1: TDBGrid;
    Panel1: TPanel;
    Label1: TLabel;
    EditFind: TEdit;
    BitBtn1: TBitBtn;
    DBNavigator1: TDBNavigator;
    Panel2: TPanel;
    Panel3: TPanel;
    DBEdit1: TDBEdit;
    procedure EditFindChange(Sender: TObject);
```

Продолжение приложения А

```
procedure DBGrid1TitleClick(Column: TColumn);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FmObject: TFmObject;

implementation
uses UnDM;
{$R *.dfm}

procedure TFmObject.EditFindChange(Sender: TObject);
begin
  DM.TbObject.Locate('Название объекта', EditFind.Text,[loCaseInsensitive,loPartialKey]);
  //Поиск по частичному совпадению без учета регистра
end;

procedure TFmObject.DBGrid1TitleClick(Column: TColumn);
begin
  if Column.Grid.DataSource.DataSet is TADOTable then
    //if Column.FieldName <> 'Единица измерения' then
begin
  if TADODataset(Column.Grid.DataSource.DataSet).Sort =
    Column.FieldName then
    // Не работает с просмотровыми полями и датой
    TADODataset(Column.Grid.DataSource.DataSet).Sort :=
    Column.FieldName + ' DESC'
  else
    TADODataset(Column.Grid.DataSource.DataSet).Sort :=
    Column.FieldName;
end;
end;

procedure TFmObject.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  DM.TbObject.Active := false;
  DM.TbObject.Active := true;
end;
end.

unit UnOrder;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```

Продолжение приложения А

Dialogs, ExtCtrls, Grids, DBGrids, DBCtrls, StdCtrls, Mask, Buttons, ComObj;

type

```
TFmOrder = class(TForm)
  Panel1: TPanel;
  Panel2: TPanel;
  Panel3: TPanel;
  Panel4: TPanel;
  Panel5: TPanel;
  DBNavigator1: TDBNavigator;
  DBGrid1: TDBGrid;
  DBLookupComboBox1: TDBLookupComboBox;
  DBEdit1: TDBEdit;
  DBLookupComboBox2: TDBLookupComboBox;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Panel6: TPanel;
  Panel8: TPanel;
  Panel10: TPanel;
  DBNavigator2: TDBNavigator;
  DBGrid2: TDBGrid;
  BitBtn1: TBitBtn;
  BitBtnUsedMat: TBitBtn;
```

```
procedure BitBtnUsedMatClick(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

```
FmOrder: TFmOrder;
```

implementation

```
uses UnDM, UnUsedMat;
{$R *.dfm}
```

```
procedure TFmOrder.BitBtnUsedMatClick(Sender: TObject);
begin
```

```
  FmUsedMat.Caption := 'Материалы, используемые при выполнении работ  
"+DM.TbWork.FieldName('Наименование работы').AsString+";
```

```
  FmUsedMat.ShowModal;
end;
```

```
procedure TFmOrder.BitBtn1Click(Sender: TObject);
```

```
var Excel: Variant;
```

```
  WorkbookName : string;
```

```
  i, j, PosInc, num: Integer;
```

Продолжение приложения А

```
sum : real;
begin
  //Cursor :=
  WorkbookName := GetCurrentDir + '\Смета.xlt';
  Excel := CreateOleObject('Excel.Application');

  try
    Excel.Workbooks.Open(WorkbookName);
    //Экспорт данных из таблиц БД:
    Excel.WorkSheets[1].Select;

    PosInc := 23; //Начальная позиция
    num := 0;

    DM.QGetClient.Active := false;
    DM.QGetClient.Parameters[0].Value :=
DM.TbOrder.FieldByName('заказчик_ид').Value;
    DM.QGetClient.Active := true;

    Excel.Cells[5,3] := DM.QGetClient.FieldByName('Наименование').AsString + '
'+DM.QGetClient.FieldByName('Адрес').AsString + '
'+DM.QGetClient.FieldByName('Телефон').AsString;

    Excel.Cells[7,3] := DM.TbRekv.FieldByName('Наименование').AsString + '
'+DM.TbRekv.FieldByName('Адрес').AsString + '
'+DM.TbRekv.FieldByName('Телефон').AsString;
    Excel.Cells[9,3] := DM.TbRekv.FieldByName('Наименование').AsString + '
'+DM.TbRekv.FieldByName('Адрес').AsString + '
'+DM.TbRekv.FieldByName('Телефон').AsString;

    DM.QGetObject.Active := false;
    DM.QGetObject.Parameters[0].Value := DM.TbOrder.FieldByName('объект_ид').Value;
    DM.QGetObject.Active := true;

    Excel.Cells[11,3] := DM.QGetObject.FieldByName('Название объекта').AsString + '
'+DM.QGetObject.FieldByName('Характеристика').AsString;
    Excel.Cells[14,7] := DM.TbOrder.FieldByName('№ заказа').AsString;
    Excel.Cells[15,7] := DM.TbOrder.FieldByName('Дата').AsString;

    Excel.Cells[19,5] := DM.TbOrder.FieldByName('№ заказа').AsString;
    Excel.Cells[19,6] := DM.TbOrder.FieldByName('Дата').AsString;

    DM.TbWork.First;
    for i:=0 to DM.TbWork.RecordCount-1 do
      begin
        if i <> DM.TbWork.RecordCount-1 then
          begin
            Excel.Rows.Item[PosInc+1].Select;
            Excel.Selection.Insert;
            Excel.Rows.Item[PosInc].Select;
```

Продолжение приложения А

```
Excel.Selection.Copy;
Excel.Rows.Item[PosInc+1].Select;
Excel.ActiveSheet.Paste;
Excel.Application.CutCopyMode := False;
end;
//Вывод данных в Excel
Excel.Cells[PosInc,1] := IntToStr(num+1);
Excel.Cells[PosInc,2] := IntToStr(i+1);

DM.QGetWork.Active := false;
DM.QGetWork.Parameters[0].Value := DM.TbWork.FieldByName('раб_ид').Value;
DM.QGetWork.Active := true;
//Получаем сведения о работе

Excel.Cells[PosInc,3] := DM.QGetWork.FieldByName('Наименование
работы').AsString;
Excel.Cells[PosInc,4] := DM.QGetWork.FieldByName('Ед изм').AsString;
Excel.Cells[PosInc,5] := FloatToStr(DM.TbWork.FieldByName('количество').AsFloat);
Excel.Cells[PosInc,6] := FloatToStr(DM.QGetWork.FieldByName('Стоимость
единицы').AsFloat);

//Excel.Cells[PosInc,5] := FormatFloat('0.00',
DM.TbWork.FieldByName('количество').AsFloat);
```

Продолжение приложения Б

```
//Excel.Cells[PosInc,6] := FormatFloat('0.00', DM.QGetWork.FieldByName('Стоимость
единицы').AsFloat);
sum := DM.QGetWork.FieldByName('Стоимость
единицы').AsFloat*DM.TbWork.FieldByName('количество').AsFloat;
//Excel.Cells[PosInc,7] := FormatFloat('0.00', sum);
Excel.Cells[PosInc,7] := FloatToStr(sum);
PosInc := PosInc + 1;

num := num+1;

DM.TbMat.First;
for j:=0 to DM.TbMat.RecordCount-1 do
begin
//if j <> DM.TbMat.RecordCount-1 then
begin

Excel.Rows.Item[PosInc+1].Select;
Excel.Selection.Insert;
Excel.Rows.Item[PosInc].Select;
Excel.Selection.Copy;
Excel.Rows.Item[PosInc+1].Select;
Excel.ActiveSheet.Paste;
Excel.Application.CutCopyMode := False;
end;
Excel.Cells[PosInc,1] := IntToStr(num+1);
```


Продолжение приложения А

```
DM.QGetMat.Active := false;
DM.QGetMat.Parameters[0].Value := DM.TbMat.FieldByName('мат_ид').Value;
DM.QGetMat.Active := true;
//Получаем сведения о работе

Excel.Cells[PosInc,3] := DM.QGetMat.FieldByName('Наименование
материала').AsString;
Excel.Cells[PosInc,4] := DM.QGetMat.FieldByName('Ед изм').AsString;
Excel.Cells[PosInc,5] := FloatToStr(DM.TbMat.FieldByName('количество').AsFloat);
Excel.Cells[PosInc,6] := FloatToStr(DM.QGetMat.FieldByName('Стоимость
единицы').AsFloat);
sum := DM.QGetMat.FieldByName('Стоимость
единицы').AsFloat*DM.TbMat.FieldByName('количество').AsFloat;
Excel.Cells[PosInc,9] := FloatToStr(sum);

PosInc := PosInc +1;
num := num+1;

DM.TbMat.Next;
end; //for j

DM.TbWork.Next;

end; //for i

Excel.Cells[PosInc+6,7] := DM.QGetClient.FieldByName('Руководитель').AsString;
Excel.Cells[PosInc+8,7] := DM.TbRekv.FieldByName('руководитель').AsString;

Excel.WorkSheets[1].Select;
Excel.Visible := True;
except
Excel.Quit;
end;
end;
end.

unit UnRekv;

interface

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ComObj, StdCtrls, Buttons, Mask, DBCtrls;

type
TFmRekv = class(TForm)
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
```

Продолжение приложения А

```
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
DBE dit1: TDBE dit;
DBE dit2: TDBE dit;
DBE dit3: TDBE dit;
DBE dit4: TDBE dit;
DBE dit5: TDBE dit;
DBE dit6: TDBE dit;
DBE dit7: TDBE dit;
DBE dit9: TDBE dit;
DBE dit10: TDBE dit;
DBE dit11: TDBE dit;
DBE dit12: TDBE dit;
DBE dit13: TDBE dit;
DBE dit14: TDBE dit;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn3: TBitBtn;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TClose Action);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FmRekv: TFmRekv;

implementation
uses UnDM;
{$R *.dfm}

procedure TFmRekv.BitBtn1Click(Sender: TObject);
begin
  If DM.TbRekv.Modified then DM.TbRekv.Post;
end;
procedure TFmRekv.BitBtn2Click(Sender: TObject);
begin
  close;
end;
```

Продолжение приложения А

```
procedure TFmRekv.BitBtn3Click(Sender: TObject);
```

```
var Excel: Variant;
```

```
    WorkbookName: string;
```

```
begin
```

```
    WorkbookName := GetCurrentDir + '\Реквизиты.xls';
```

```
    Excel := CreateOleObject('Excel.Application');
```

```
try
```

```
Excel.Workbooks.Open(WorkbookName);
```

```
    //Экспорт данных из таблиц БД:
```

```
Excel.WorkSheets[1].Select;
```

```
    Excel.Cells[3,2] := DM.TbRekv.FieldByName('Наименование').AsString;
```

```
    Excel.Cells[5,2] := DM.TbRekv.FieldByName('ИНН').AsString;
```

```
    Excel.Cells[7,2] := DM.TbRekv.FieldByName('КПП').AsString;
```

Продолжение приложения Б

```
Excel.Cells[9,2] := DM.TbRekv.FieldByName('Адрес').AsString;
```

```
    Excel.Cells[11,2] := DM.TbRekv.FieldByName('Телефон').AsString;
```

```
    Excel.Cells[13,2] := DM.TbRekv.FieldByName('Web-страница').AsString;
```

```
    Excel.Cells[15,2] := DM.TbRekv.FieldByName('Электронная почта').AsString;
```

```
    Excel.Cells[3,4] := DM.TbRekv.FieldByName('Банк').AsString;
```

```
    Excel.Cells[5,4] := DM.TbRekv.FieldByName('БИК').AsString;
```

```
    Excel.Cells[7,4] := DM.TbRekv.FieldByName('P/c').AsString;
```

```
    Excel.Cells[9,4] := DM.TbRekv.FieldByName('K/c').AsString;
```

```
    Excel.Cells[11,4] := DM.TbRekv.FieldByName('Руководитель').AsString;
```

```
    Excel.Cells[13,4] := DM.TbRekv.FieldByName('Бухгалтер').AsString;
```

```
Excel.WorkSheets[1].Select;
```

```
Excel.Visible := True;
```

```
except
```

```
    Excel.Quit;
```

```
end;
```

```
end;
```

```
procedure TFmRekv.FormClose(Sender: TObject; var Action: TCloseAction);
```

```
begin
```

```
    DM.TbRekv.Active := false;
```

```
    DM.TbRekv.Active := true;
```

```
end;
```

```
end.
```

```
unit UnSMat;
```

```
interface
```

```
uses
```

```
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
    Dialogs, DBCtrls, StdCtrls, Buttons, ExtCtrls, Grids, DBGrids, DB, ADODB;
```

```
type
```

Продолжение приложения А

```
TFmSMat = class(TForm)
DBGrid1: TDBGrid;
Panel1: TPanel;
Label1: TLabel;
EditFind: TEdit;
BitBtn1: TBitBtn;
DBNavigator1: TDBNavigator;
Panel2: TPanel;
procedure EditFindChange(Sender: TObject);
procedure DBGrid1TitleClick(Column: TColumn);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FmSMat: TFmSMat;

implementation
uses UnDM;
{$R *.dfm}

procedure TFmSMat.EditFindChange(Sender: TObject);
begin
  DM.TbSMat.Locate('Наименование материала',
EditFind.Text,[loCaseInsensitive,loPartialKey]);
  //Поиск по частичному совпадению без учета регистра
end;

procedure TFmSMat.DBGrid1TitleClick(Column: TColumn);
begin

  if Column.Grid.DataSource.DataSet is TADOTable then
  if Column.FieldName <> 'Единица измерения' then
begin
  if TADODataSet(Column.Grid.DataSource.DataSet).Sort =
Column.FieldName then
  // Не работает с просмотровыми полями и датой
  TADODataSet(Column.Grid.DataSource.DataSet).Sort :=
Column.FieldName + ' DESC'
  else
  TADODataSet(Column.Grid.DataSource.DataSet).Sort :=
Column.FieldName;
end;
end;

procedure TFmSMat.FormClose(Sender: TObject; var Action: TCloseAction);
begin
```

Продолжение приложения А

```
DM.TbSMat.Active := false;
DM.TbSMat.Active := true;
end;
end.

unit UnSWork;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DBCtrls, ExtCtrls, Grids, DBGrids, StdCtrls, DB, ADODB, Buttons;

type
  TFmSWork = class(TForm)
    DBGrid1: TDBGrid;
    Panel1: TPanel;
    DBNavigator1: TDBNavigator;
    Panel2: TPanel;
    EditFind: TEdit;
    Label1: TLabel;
    BitBtn1: TBitBtn;
    procedure DBGrid1TitleClick(Column: TColumn);
    procedure EditFindChange(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FmSWork: TFmSWork;

implementation
uses UnDM;
{$R *.dfm}

procedure TFmSWork.DBGrid1TitleClick(Column: TColumn);
begin

  if Column.Grid.DataSource.DataSet is TADOTable then
    if Column.FieldName <> 'Единица измерения' then
begin
if TADODataset(Column.Grid.DataSource.DataSet).Sort =
Column.FieldName then
  // Не работает с просмотровыми полями и датой
  TADODataset(Column.Grid.DataSource.DataSet).Sort :=
  Column.FieldName + ' DESC'
else
```

Продолжение приложения А

```
TADODataset(Column.Grid.DataSource.Dataset).Sort :=
Column.FieldName;
end;
end;

procedure TFmSWork.EditFindChange(Sender: TObject);
begin
  DM.TbSWork.Locate('Наименование работы',
EditFind.Text,[loCaseInsensitive,loPartialKey]);
  //Поиск по частичному совпадению без учета регистра
end;

procedure TFmSWork.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  DM.TbSWork.Active := false;
  DM.TbSWork.Active := true;
end;
end.

unit UnUsedMat;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, DBCtrls, ExtCtrls;
type
  TFmUsedMat = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    DBNavigator1: TDBNavigator;
    DBGrid1: TDBGrid;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FmUsedMat: TFmUsedMat;

implementation
uses UnDM;
{$R *.dfm}
end.
```

Приложения Б АКТ Внедрения


ТОО «АБК Аль-Хади Әділет»
БИН 090440004131
Республика Казахстан,
г. Алматы, Бухар жырау 64
e-mail: d.kairat@bk.ru
тел. +7(701)788-44-65

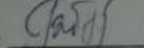


Утверждаю
Директор ТОО
«АБК Аль-Хади Әділет»
Даулбаев К.Б.
«20» мая 2020г.

АКТ ВНЕДРЕНИЯ

Настоящий акт свидетельствует о том, что результаты дипломной работы студента НАО АУЭС им. Г.Даукеева, гр. ВТ-16-2, очной формы обучения Байдаулет О.К. на тему «Разработка автоматизированной системы расчета и калькуляции стройматериалов» в ТОО «АБК Аль-Хади Әділет» и используются для работы предприятия. Использование результата выпускной работы Байдаулет О.К. обеспечивает доступ к детализации по отдельным типам товаров, организация поставки, складом, актуальности товаров в требуемый отрезок даты.

Директор ТОО «АБК Аль-Хади Әділет»  Даулбаев К.Б.

Исполнитель  Байдаулет О.К.

