**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИ\_Е ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ФИНАНСОВО­ ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»**

Факультет электронного обvчения

|  |  |  |  |
| --- | --- | --- | --- |
| **Направление** | ***09.03.03*** | **Кафедра** | эо |
|  | *(код)* |  | *(аббревиатура)* |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**на тему: АВТОМАТИЗАЦИЯ ПРОЦЕССА НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ, ИСЧИСЛЕНИЮ ИНДИВИДУАЛЬНОГО ПОДОХОДНОГО НАЛОГА И**

**ИСЧЛСЛЕНИЯ СОЦИАЛЬНОГО НАЛОГА В ТОО «ВЕРШИНА**

Обучающийся

*(Ф.И.*О. *полностью) подпись*

Руководитель

*(Ф.И.*О. *полностью) подпись*

Декан ФЭО

*(Ф.И.О. полностью) подпись*

**МОСКВА2017 г.**

**СОДЕРЖАНИЕ**

[**1** АНАЛИТИЧЕСКАЯ ЧАСТЬ 7](#_bookmark0)

* 1. [Технико-экономическая характеристика предметной области и](#_bookmark1) предприятия. Анализ деятельности ТОО «Вершина» 7
     1. [Характеристика предприятия и его деятельности 7](#_bookmark2)
     2. [Организационная структура управления предприятием 9](#_bookmark3)
     3. [Программная и техническая архитектура ИС предприятия **1**О](#_bookmark4)
  2. [Характеристика комплекса задач, задачи и обоснование необходимости](#_bookmark5) автоматизации 13
     1. [Выбор комплекса задач автоматизации и характеристика](#_bookmark5) существующих бизнес процессов 13
     2. [Определение места проектируемой задачи в комплексе задач и ее](#_bookmark6) описание 19
     3. [Обоснование необходимости использования вычислительной](#_bookmark7) техники для решения задачи 25
     4. [Анализ системы обеспечения информационной безопасности и](#_bookmark8) защиты информации 26
  3. [Анализ существующих разработок и выбор стратегии автоматизации](#_bookmark9)

[«процесса начисления заработной платы, исчислению индивидуального](#_bookmark9) подоходного налога и исчислению социального налога ТОО «Вершина» 30

* + 1. Анализ существующих разработок для автоматизации задачи 30
    2. [Выбор и обоснование стратегии автоматизации задачи](#_bookmark10) 31
    3. [Выбор и обоснование способа приобретения ИС для автоматизации](#_bookmark11) задачи 32
  1. [Обоснование проектных решений 34](#_bookmark12)
     1. [Обоснование проектных решений по информационному](#_bookmark13) обеспечению 34
     2. [Обоснование проектных решений по программному обеспечению .40](#_bookmark14)
     3. [Обоснование проектных решений по техническому обеспечению 47](#_bookmark15)

[11 ПРОЕКТНАЯ ЧАСТЬ 50](#_bookmark16)

* 1. [Разработка проекта автоматизации 50](#_bookmark17)
     1. [Этапы жизненного цикла проекта автоматизации 50](#_bookmark18)
     2. [Ожидаемые риски на этапах жизненного цикла и их описание](#_bookmark19) 55
     3. [Организационно-правовые и программно-аппаратные средства](#_bookmark20) [обеспечения информационной безопасности и защиты информации 57](#_bookmark8)
  2. [Информационное обеспечение задачи 59](#_bookmark21)
     1. [Информационная модель и её описание 59](#_bookmark22)
     2. [Характеристика нормативно-справочной, входной и оперативной](#_bookmark23) информации 62
     3. [Характеристика результатной информации 66](#_bookmark24)
  3. [Программное обеспечение задачи 66](#_bookmark25)
     1. [Общие положения (дерево функций и сценарий диалога) 66](#_bookmark26)
     2. [Характеристика базы данных 68](#_bookmark27)
     3. [Структурная схема пакета (дерево вызова программных модулей) .68](#_bookmark28) [2.3.4. Описание программных модулей 70](#_bookmark29)
  4. Контрольный пример реализации проекта и его описание 72

[111 ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА 79](#_bookmark30)

* 1. [Выбор и обоснование методики расчёта экономической эффективности79](#_bookmark31)
  2. [Расчёт показателей экономической эффективности проекта 83](#_bookmark32)

[ЗАКЛЮЧЕНИЕ 89](#_bookmark33)

[Литература 93](#_bookmark34)

[ПРИЛОЖЕНИЕ 97](#_bookmark35)

ВВЕДЕНИЕ

Во все времена, заработная плата служила и продолжает служить основ­ ным источником дохода для сотрудников предприятий и важнейшим экономи­ ческим рычагом управления экономикой. Она позволяет осушествлять контроль над мерой труда и потребления.

Оплата труда является важнейшим участком бухгалтерского учета и тре­ бует от бухгалтера высокой профессиональной компетентности. Заработная пла­ та позволяет мотивировать работников на профессиональный труд. Качествен­ ное выполнение должностных обязанностей зависит от заинтересованности ра­ ботников, которая в свою очередь обеспечивается правильной организацией уче­ та труда, установлением эффективных систем оплаты труда и поощрения со­ трудников. В свою очередь, соблюдение действующего законодательства по труду и заработной плате защищает организации от налоговых санкций.

Приоритетным направлением развития информатизации в даном конртек­ сте является автоматизация расчета начисления заработной платы и удержаний из нее. Уровень развития информационных технологий на сегодняшний день позволяет сократить объем рутинной работы с документами, усовершенствовать процессы сбора, хранения и обработки данных, получить полную картину по учету расчетов с персоналом по оплате труда, а также позволяет избежать меха­ нических ошибок при подсчетах.

Учет оплаты труда - это важнейший раздел бухгалтерского учета, в этой связи тема автоматизации бухгалтерского учета стала особенно актуальна в наши дни. Расходы на оплату труда являются важнейшим элементом издержек производства и обращения. Для облегчения работы бухгалтера требуется опти­ мизировать процесс по начислению заработной платы и расчету удержаний. В настоящее время существует множество профессиональных бухгалтерских си­ стем (ПБС), призванных облегчить работу бухгалтера за счет автоматизации многих операций бухгалтерского и налогового учета. Тем не менее, все они имеют определенные недостатки, сложны в освоении и достаточно дорого стоят.

Таким образом, целью данного проекта является разработка системы ав­ томатизации процесса начисления заработной платы, расчета индивидуального налога на доходы физических лиц и исчисления социального налога.

Задачи дипломного проекта:

изучить предметную области и выявить недостатки существующей систе­ мы начисления заработной платы, расчета индивидуального налога на до­ ходы физических лиц и исчисления социального налога в ТОО «Верши­ на»;

разработать постановку задачи;

обосновать выбор основных проектных решений;

разработать обеспечивающие подсистемы для начисления заработной платы, расчета индивидуального налога на доходы физических лиц и ис­ числения социального налога;

обосновать экономическую эффективность проекта.

Объектом исследования в данной работе является ТОО «Вершина», зани­ мающееся производством строительных материалов.

Предметом исследования является автоматизация процесса начисления заработной платы, расчета индивидуального налога на доходы физических лиц и исчисление социального налога.

В процессе написания работы использованы следующие методы исследо-

вания:

теоретический анализ литературных источников,

анализ современных фирм-производителей автоматизированных инфор­ мационных систем на российском рынке;

структурный анализ - один из формализованных методов анализа требо­ ваний к программному обеспечению. В этом методе программное изделие рассматривается как преобразователь информационного потока данных; метод факторного анализа;

метод экспертных оценок,

методики оценки финансовой эффективности

Разработка системы автоматизации процесса начисления заработной пла­ ты, исчисления индивидуального налога на доходы физических лиц и социаль­ ного налога основана на следующих принципах: принцип единой БД; принцип одноразового ввода и накопления информации; принцип минимизации ввод и вывода; принцип централизации обработки информации; принцип гибкости ин­ формационной базы.

Данная работа состоит из трех частей.

В первой, аналитической, части представлена аналитическая часть проек­ та, в которой показаны технико-экономическая характеристика ТОО «Вершина» и существующие на предприятии методики расчета заработной платы, исчисле­ ния индивидуального налога на доходы физических лиц и социального налога; характеристика комплекса задач, требующих решения, и обоснование необхо­ димости автоматизации описанного бизнес-процесса; анализ существующих разработок, обоснование проектных решений и выбор стратегии автоматизации процесса расчета заработной платы, исчисления налога на доходы физических лиц и социального налога.

Вторая часть работы охватываетсобственно проектную часть, включает разработку проекта автоматизации процесса расчета заработной платы, индиви­ дуального налога на доходы физических лиц и социального налога, информаци­ онного и программного обеспечения автоматизации описанного процесса, а так­ же описания контрольного примера реализации проекта.

В третьей части работы представлено обоснование экономической эффек­ тивности проекта. В ней описывается выбор методики расчёта, а также пред­ ставлен математический расчёт показателей экономической эффективности ав­ томатизации процесса расчета заработной платы, исчисления индивидуального налога на доходы физических лиц и социального налога.

В заключении приведены основные выводы трех глав.

## 1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Технико-экономическая характеристика предметной области и предприятия. Анализ деятельности ТОО «Вершина»

## Характеристика предприятия и его деятельности

ТОО «Вершина» - известный производитель строительных материалов. Основной продукцией, которую производит ТОО «Вершина» является силикат­ ный кирпич, а также объемно-окрашенный кирпич Боровичи, который известен далеко за пределами региона.

Первая очередь ТОО «Вершина» была запущена в 1974 году. В те годы объем производства составлял порядка 50 миллионов штук в год. Сейчас произ­ водственные мощности ТОО «Вершина» значительно выросли, и за прошедшие годы предприятие выпустило в общей сложности почти полтора миллиарда штук кирпича.

За последние годы в ТОО «Вершина» было введено много инновацион­ ных технологий. В ихчисле можно отметить полную модернизацию оборудова­ ния, установку четырех автоклавов с автоматическим управлением, замену ста­ рых прессов. Кроме этого, можно отметить значительное расширение автопарка, который в настоящее время целиком обеспечивает собственные нужды предпри­ ятия. Сегодня в ТОО «Вершина» работают более 450 человек, для которых, в рамках социальной программы, построены спортивный комплекс, хорошая дет­ ская площадка и школа.

В 1990 году ТОО «Вершина», так же, как и многие промышленные пред­ приятия страны, пережил кризис. После достигнутого высокого объема выпуска продукции, превышающего запланированную мощность, ТОО «Вершина» сни­ зил выпуск продукции со 100 млн. штук в год до 24 млн. штук. Спад продолжал­ ся вплоть до 1996 года, а потом объемы производства стали расти. В настоящее время ТОО «Вершина» работает стабильно, а реализация его продукции ежегод­ но растет на 10-15%. На сегодняшний день ТОО «Вершина» является одним из крупнейших производителей силикатного кирпича. Ассортимент продукции включает изделия практически любых возможных расцветок.

Основные виды деятельности компании:

* проектно-изыскательные работы;
* строительно- монтажные работы;
* работы по производству строительных материалов;
* строительство объектов жилищного, культурного и социально- культурного назначения, промышленных объектов.

До настоящего времени хорошо освоен рынок строительно-монтажных работ и рынок по производству пластиковых окон.

В начале 2007 г. установлена автоматическая линия по сборке стеклопаке­ тов. ТОО «Вершина» располагает цехами по выпуску брусчатки, тротуарной плитки и цокольного камня.

ТОО «Вершина» имеет хорошую материально-техническую базу.Она включает три промышленные площадки:

На первойплощадке, общей площадью 90000 кв.м. находится цех по изго­ товлению силикатного кирпича, цех по изготовлению тротуарной плитки, склад­ ские помещения и гаражи.

На второйплощадке, общей площадью 54 300 кв. м. находится база УМТС, два железнодорожных тупика с козловыми кранами грузоподъемностью 30 т. и др. техника. Также на территории расположен мини-завод по сборке го­ товых металлопластиковых изделий (дверные и оконные блоки, витражи). Уста­ новлены 2 линии турецкой компании «Kaban» и «Murat», фурнитура - герман­ ская.

На третьей площадке, площадью 28 ООО кв. метров, находится гараж, теп­ лые боксы. На территории этой базы расположен столярный цех, работающий на оборудовании, приобретенном в Турции.

Для складирования и хранения материалов, оборудования, запчастей ТОО

«Вершина» имеет большое количество складских помещений и площадок.

Таким образом, необходимость ввода различных АИС продиктована как объемом и разбросанностью производства, так и постоянным расширением спектра работ, а также ростом количества персонала.

Таблица **1**

Объемы выпуска продукции в натуральном выражении

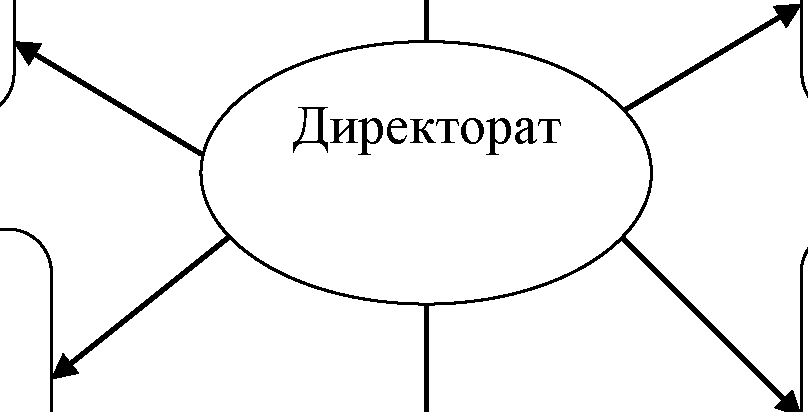
|  |  |  |
| --- | --- | --- |
| №  **п.\п.** | **Наименование характеристики** | **Объем** |
| **1** | Кирпич полнотелый М-150 | 250 ТЫС. ШТ. |
| 2 | Кирпич полнотелый М-200 | 550 ТЫС. ШТ. |
| 3 | Кирпич полнотелый утолщенный М-125 | 30 ТЫС. ШТ. |
| 4 | Кирпич полнотелый утолщенный М-150 | 77 ТЫС. ШТ. |
| 5 | Кирпич полнотелый утолщенный М-200 | 24 ТЫС. ШТ. |
| 6 | Кирпич пустотелый утолщенный М-125 | 60 ТЫС. ШТ. |
| 7 | Кирпич пустотелый утолщенный М-150 | 28 ТЫС. ШТ. |
| 8 | Кирпич пустотелый утолщенный М-150 | 32 ТЫС. ШТ. |
| 9 | Кирпич пустотелый лицевой | 30 ТЫС. ШТ. |
| 10 | Плитка тротуарная, брусчатка | 30 ТЫС. КВ. М. |

## Организационная структура управления предприятием

Главный офис компании расположен по адресу: Новгородская область, г.

Боровичи, ул. Окуловская, 10.

Общее руководство находится в главном офисе. В этой связи организаци­ онная структура управления построена по схеме звезда. Каждое подразделение предприятия ежемесячно представляет отчет о производственных показателях, утверждает планы работ, сметы и др.



Участок троту­

арной плитки

Кирпичный

участок

Участок

ме-

таллопласти­

ковых излелий

Первая база

Песчаный ка-

рьер

Складская база

Рисунок 1. Укрупненная структура управления предприятием Рассматриваемые в работе процессы начисления заработной платы, хра­

нения и обработки данных о ранее произведенных начислениях, вывод отчетов

отслеживались в бухгалтерии головного офиса.

## Программная и техническая архитектура ИС предприятия

Понятие «программная архитектура», в зависимости от контекста, может пониматься по разному. Во-первых, она может трактоваться как архитектура взаимодействия приложений в рамках информационной системы предприятия (архитектура приложений). Во-вторых, она может иметь значение архитектуры программных модулей или архитектуры взаимодействия различных классов в рамках одного приложения. 1 Тем не менее, каждая из перечисленных архитек­ тур, может рассматриваться под определенным углом зрения и с различной сте­ пенью детализации. К примеру, в число традиционных для программной архи­ тектуры можно назвать следующие перспективы или уровни ее описания:

концептуальная архитектура; логическая архитектура; физическая реализация.

На уровне концептуальной архитектуры определяют, что система должна уметь делать с точки зрения конечного пользователя, т.е. компоненты системы и их назначения в неформальном виде. Это представление обычно используется для обсуждения деталей с нетехническими специалистами, такими как админи­ страторы и конечные пользователи функциональных характеристик системы.

Логическая архитектура должна уже определить взаимосвязи между ком­ понентами системы, описывать интерфейсы, протоколы взаимодействия. Этот уровень представление обеспечивает эффективную организацию параллельной разработки.

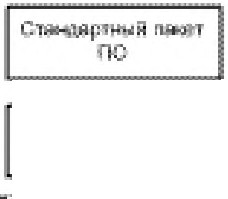
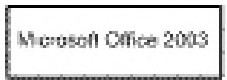
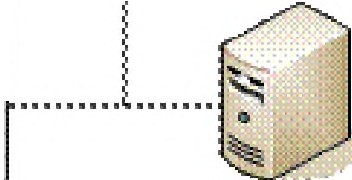
На уровне физической реализации рассматривается привязка к конкрет­ ным узлам размещения, типам оборудования, характеристикам окружения, та­ ким как, например, используемые операционные системы.

1 Трутнев Д. Р. Архитектуры информационных систем. Основы проектирования: Учебное пособие. - СПб.: НИУ ИТМО, 2012. - 66 с. (стр. 21)

Рассмотренные выше положения определены ANSI \ IEEE Std 1471 - 2000\_ Они задают лишь общую модель разработки архитектуры, но являются полез­ ными для реализации архитектурного подхода разработки программного обес­ печения\_

Программная архитектура предприятия включает в себя лицензированные операционные системы и офисные пакеты фирмы Майкрософт, бухгалтерскую систему lC, бухгалтерию 8\_ 1

Программная архитектура является совокупностью программных средств организации диалога, операционных систем, программ и программных средств администрирования баз данных, которые расширяют возможности операцион­ ных систем, специального программного обеспечения (уникальные программы пользователя, функциональные пакеты прикладных программ) [6]\_



**йi:,,Atll.i'A**

**....,.,-;ро**

**C-,1:нtwti.mн:: l"в 'i'**

**l1'J**

...........

**[1-f[E-p**

-

**i- O'.ыtA 1.1:D\'Ц.fR**

<f,.,r.urmpa

[ **"4k:rooot101tD!!! •** 1

........,., 1

!Ю

..J

1 r.кrosofl oltc,o *>Ш* 1

'**j** 1◊ "'°1"

1.........................

...., - ... - ....... **(-,:,;-.'1,·**

'}

**..;;... •1 -,J\.Vli'L"**

... - ·i·,. ***r"*** .;\_;,- -..- :-::

- п:. -r. **l.1:v:• - 1**

**Cфi.llr,llill...lalUIM:**

**;!<j!)o.** "Ч .....

,..,,(>О<,.O!i *;;,,п*

**ilflCПl!tUМ.П**

* **ooтp'!дJl.lt.iilDТд,сп,.1**

с.б!,1'1

1

1

**6(tttNJ"Jt,;tп'**

1 t;лм..,- ,с r P""'"" I

|  |  |
| --- | --- |
|  | **P RCП!tnJltA**    1 |
| ""11ч,wт-<;<> namr ПD 1 |
| llr.и111Y Н: Прод1"f1П110 1 |
| 1&;.,t,: ! |

|  |
| --- |
| **Р езс:т М**  **tо'J'И'д•и,;,ж.ос** |
| **1 t.4 -**ОК 11:ОЭ **1** |
| lс,-щ н•..-т **1** |

|  |
| --- |
| **1,1t1,:-'f;(,нu\_н.t**  **::DТJГ Ю:Ю ""f!t.!llllloд,:ПUШIIO."'D**  **O<Til.l)М** |
| **1** м,n>dun м, **1** |
| **1** G,\_ao • т **1** |
|  |





Рисунок 2\_ Программная архитектура ИС

На компьютерах анализируемой организации установлен стандартный па­ кет программного обеспечения:

Операционная система Windows 10; Пакет MS Office 2013;

Антивирус Касперского; Браузер Mozilla Firefox; Ридер Adobe Acrobat;

Справочно-правовая система Консультант Плюс;

С учетом специфики деятельности персональные компьютеры работников оснащены дополнительным программным обеспечением (персональные пакеты программного обеспечения для отделов): бухгалтерия: lC Бухгалтерия: Версии 8.1, Банк-Клиент.

Техническая архитектура представлена на рисунке

.28 U



2U

2U

**Папч ПiJHGЛL.**

Мврwруп,""'т□р

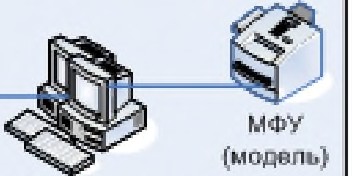
**СИ!:-ТJ31.-'18 8Щ 1ТЫ**

(""д•,..1

l

**l!!==,--,-ИСерве**,**р**На ,н"••

{.,од,..ль)



эu

**Дl,'ICIIOIJI.Jltr,аl н; ;:; 13**

(модель. □**бъа,)**

1 U **1111ИБП**\"'одель)

Маршрv,,,,атор

\модель)

Ома тфаt11, rотрrдниооо (,..,дельj

Уда. н:.нные rотрvдни1<>1

lntвrriet

**Оис:т1:1ма J.аuциrт,ы**

{•юД8Ло)

***ОЩ..Л* 4M8.IUHIИI- nнlt:I**

**OJМ.Лlrl •ПЮНl!'f!!Р Г JППlrJI**

**l:lr d;l-**■**•**■■**n..&­**

**.. lt.lUII\_III., ГPJIП-.i**■

Paoos • &Т<IНШ,,t,

(,.<>деnь)

•Pfl"P ,на,,.,,нио, Маршр,,,,.,.аrор

(Ш!д,,Л•) (,.OД8Jlb)

**Wlliil т ц ДQGTYM**

(м<>Двль)

Рисунок 3. Техническая архитектура ИС

# Характеристика комплекса задач, задачи и обоснование необходимости автоматизации

# Выбор комплекса задач автоматизации и характеристика существующих бизнес процессов

На предприятии используется программный комплекс «1С: Предприятие» конфигурация «IC: Бухгалтерия». Данная конфигурация позволяет отражать начисленную заработную плату в учете, но не обеспечивает расчет различных видов начислений и удержаний, особенно, если в организации работают наем­ ные рабочим - не резиденты РФ.

Главная цель реализации проекта - автоматизация процесса начисления заработной платы, хранения и обработки данных о ранее произведенных начис­ лениях, вывод отчетов.

Необходимо создать программу, которая позволяет автоматизировать ве­ дение базы данных о начислении заработной платы сотрудникам, работающим на предприятии.

о

(1)

t::;

.о

*]i*

(1)

:s:

lii

lii

**r:;[**

о

,. I

а.

2:

:I:

о

:о,:

,

IU

(У)

~~-~~

Табель учета рабочего времени

Расчетная ведомость

Наряд на сдельную работу

Результаты инвентаризации Положение по оплате труда

-

--

-

~~-~~

Отчетность

--

**j**

а.

'

lii

о

а.

t::;

:s:

а.

:..i.i.

**<Lo**

1-

**:s:**

2:

**r:;[**

<(

:I:

**х**

L!)

>

Расчет заработной П11 аты

Рисунок 4. Контекстная диаграмма расчета заработной платы

При анализе сущестующих бизнес-процессов можно увидеть, что на пред­ приятии используются две формы начисления заработной платы: для админи­ стративно-управленческого персонала - повременная, для основных рабочих - сдельная.

Входные данные - табель учета отработанного времени, результаты ин­ вентаризации, отчет по продажам, положение об оплате труда. Выходные дан­ ные - расчетная ведомость, отчетность.

Результаты анализа представлены в контекстной диаграмме расчета зара­ ботной платы.

Управляющими потоками функций являются нормативы и законодатель­

ство.

Потоками, осуществляющим преобразование входных потоков в выход­

ные является администратор (бригадир, руководитель подразделения), бухгал­ тер.

Выполним декомпозицию контекстной диаграммы.

о

Администрация

,

,

•

Рабочая бригада

'

I

:ii

•

Prtl"'-1f>THrtQ RP.nn• nrтt-...

О.четность

-

•

1D

t

.!J

**q**1о

1D

:s: 1о 2 оС1.

наряд на сдельную работv

результаты инвентаризации

табель учета

абочего времени оложение по

о плате труда

*::*

.-

*::*

-

'

оС1.

1о

:ts:

С1.

**q**

::s:i::

2

'

о

:о:i:

ro

*::* (\"')

С1.

">-'

**х**

>

LD

Рисунок 5. Детализация контекстной диаграммы Рассмотрим процесс начисления заработной платы цля рабочей бригады.

.;.;

""-

**.о**

:"i::

<(

"'

..

""'

"

,::

Нарядна rдел оную

раба,у

>--

,,

Положение па ома

труда

'+ Расчетс,цельнага

зараба11G1

·

,\_

Табельучета рабочего вре1.1ени

] Расчетдаnлат

1

'

"-

+

t

'

Расчет

удержан11й

0,четнасть

*:f*

""'

",:':

"-

Расчетная **веда**1.1 асть

lii

j

Учет11згатnменных 11зделий (выnQ11ненных рабат)

Рисунок 6. Декомпозиция блока расчета заработной платы рабочей брига­

ды

Администратор рабочей бригады заполняет наряд на сдельную работу, на основании которого рассчитывается сдельный заработок. Также администратор рабочей бригады составляет табель учета рабочего времени, на основании кото­ рого рассчитываются доплаты.

Входные потоки: табель учета рабочего времени, наряд на сдельную рабо- ту, положение об оплате труда.

Выходные потоки: расчетная ведомость, отчетность. Управление: нормативы, законодательство.

Механизмы исполнения: администратор, бухгалтер.

Рассмотрим процесс начисления заработной платы для административно­ го персонала.

.;;,

I "

"Q.

.о

<[

":,:

,,.,.,

:

.....

**Положение по ама-rе**

L+

:

Расчетнадба11Jк и

**пр**·**еuий**

-

,.

•

**Расчет доплат**

**t**

'

**01ЧеniОСТЬ**

Q.

!!!

Расчет

удержаний

1

-

:

ijj

1

1

j-

'1'

**Расчетная**

**ведоuость**

**Табельучета рабочего вреuени**

труда

Расчет" К/lада

Рисунок 7. Декомпозиция блока расчета заработной платы администра­

тивного персонала

Входные потоки: табель учета отработанного времени, положение об оплате труда.

Выходные потоки: расчетная ведомость, отчетность. Управление: нормативы, законодательство.

Механизмы исполнения: бухгалтер.

В результате анализа литературных источников и бесед с бухгалтерами предприятия, были сформулированы следующие требования к программе:

1. Полнота и подробность информации

Программа должна позволять вести базу данных о начислении заработной платы на предприятии, при этом выводимая информация должна быть полной, подробной, и разделена тематически, т. е. предоставлять возможность раздель­ ного просмотра начислений заработной платы, начислений индивидуального по­ доходного налога, отчислений в пенсионный фонд и в другие социальные фон­ ды, в разрезе структурных подразделений и персонально каждого сторудника.

1. Максимальное быстродействие

В разрабатываемой программе произведение всех расчетов должно быть максимально автоматизировано, чтобы позволить максимально сократить время,

затрачиваемое бухгалтерией на расчет заработной платы. Для того чтобы обес­ печить такую быстроту в работе необходимо, чтобы на момент проведения рас­ четов большая часть информации необходимой для проведения данных расчетов уже хранилась в справочниках баз данных.

Для проведения корректных расчетов индивидуального подоходного налога и размера социальных отчислений необходимо иметь следующую ин­ формацию:

* размер минимальной заработной платы
* значение месячного расчетного показателя Для каждого из работников:
* размер совокупного дохода
* количество человек, находящихся на его иждивении
* льготная категория (нет льгот, участник ВОВ, инвалид I или II группы, родитель инвалида с детства, инвалид 111 группы).

1. Вывод информации в виде отчетов.

Должен иметься модуль, позволяющий формировать годовые отчеты на каждого из сотрудников, а также просматривать эти отчеты на экране компьюте­ ра и отправлять их на печать.

Отчет по исчислению и удержанию индивидуального подоходного налога с физических лиц должен содержать следующую информацию:

* сумма начисленного совокупного дохода за текущий месяц
* сумма начисленного совокупного дохода с начала года
* доходы, не подлежащие налогообложению за текущий месяц
* сумма подоходного налога, удержанная в предыдущем периоде
* доходы, не подлежащие налогообложению с начала года
* налоговые вычеты за текущий месяц
* налоговые вычеты с начала года
* сумма облагаемого дохода фактическая
* сумма подоходного налога с дохода за истекший период
* сумма подоходного налога, подлежащая удержанию в текущем месяце Отчет по исчислению и удержанию страховых взносов должен содержать

следующую информацию:

* доходы, не подлежащие обложению страховыми взносами за текущий

месяц

* доходы, не подлежащие обложению страховыми взносами с начала года
* доходы, облагаемые страховыми взносами за текущий месяц
* доходы, облагаемые страховыми взносами с начала года
* расчетная сумма дохода, облагаемого страховыми взносами
* доходы физических лиц, с которых исчисляются страховые взносы
* сумма страховых взносов
* сумма страховых взносов с расчетного годового дохода по таблице
* сумма страховых взносов с дохода за истекший период
* сумма страховых взносов, удержанная в предыдущем периоде
* сумма страховых взносов, подлежащая удержанию в текущем месяце
* сумма страховых взносов подлежащего уплате в бюджет

1. Возможность проведения расчетов вручную

Автоматизированные расчеты - это, безусловно, хорошо, но иногда, в

особых случаях может возникнуть необходимость произвести вручную часть или все расчеты. Программа должна позволять это делать, и более того должна предоставлять всю необходимую справочную информацию о сотрудниках в окне начислений, с целью максимально облегчить работу пользователя.

1. Наличие справочной системы

В программе должна быть реализована справочная система, помогающая начинающему пользователю освоить все аспекты работы с программой.

В расчете заработной платы выделяют следующие этапы работы:

разово, по мере необходимости в информационную базу вводится

«условно-постоянная» информация кадрового учета, которая в дальнейшем используется в расчетах начислений и удержаний

в течение месяца вводятся документы, определяющие трудовые по­ казатели каждого работника, а также другие документы и сведения, влияющие на размер начисления или удержания (больничные листы и т.п.)

после завершения периода производится собственно расчет зара­ ботной платы и связанных с ней налогов и взносов. По результатам этого расчета формируются данные о суммах, подлежащих выплате работникам

С помощью данной подсистемы можно сформировать и распечатать необ­ ходимые документы, а также получить отчеты, связанные с начислением зара­ ботной платы.

# Определение места проектируемой задачи в комплексе за­ дач и ее описание

Для того, чтобы бизнес развивался успешно, необходимо заинтересовать работников в конечных результатах, чтобы сотрудники стремились увеличивать свой вклад в успех общего дела. Для реализации поставленных целей, в свою очередь, необходимо использовать методы оплаты труда, которые способствуют созданию деловой атмосферы и ориентируют работников на достижение лучших результатов. В то же время методы оплаты труда должны выполняться в сот­ тветсвтиии с требованиями законодательства.

Указанные задачи решаются в программной системе, обеспечивая автома­ тическое исчисление оплаты труда, налогов и взносов, прочих удержаний в со­ ответствии с действующим законодательством.

Проектируемая система должна полностью автоматизировать комплекс трудоемких расчетов с персоналом, начиная с ввода документов о фактической выработке, отработанном времени, перерывов в работе, подлежащих оплате, и заканчивая формированием документов на выплату зарплаты и отчетности в государственные контролирующие органы. Посредством актуализации справоч­ ников, автоматизируется расчет регламентированных законодательством начис­ лений и удержаний, отражение результатов расчетов в затратах предприятия.

В подсистеме расчета оплаты труда должны быть автоматизированы прак­ тически все виды начислений оплаты труда и связанных с ней удержаний, пало-

гов и взносов. На предприятии применяются две основные формы оплаты труда: повременная и сдельная. Повременная оплата труда ставит размер заработка в зависимость от отработанного времени. Сдельная оплата труда ставит заработок в зависимость от количества выполненной работы. В ТОО «Вершина применя­ ются повременно-премиальная и сдельно-премиальная системы оплаты труда.

Все начисления организации объединены в две большие группы:

основные начисления - это начисления, за то время, когда сотрудник находился на своем рабочем месте;

дополнительные начисления - это начисления за то время, когда со­ трудник отсутствовал на своем рабочем месте, но по закону это время подлежит оплате.

Каждый вид начислений характеризуется способом расчета и другими па­ раметрами.

Основные начисления могут рассчитываться следующими способами:

по тарифной ставке (месячной, дневной или часовой) за фактический пе­ риод действия начисления

сдельно за фактический период действия начисления

процентом (от сумм, начисленных по указанным базовым начислениям) фиксированной суммой

Дополнительные начисления могут рассчитываться следующими спосо-

бами:

по среднему заработку для отпуска

по среднему заработку для исчисления пособий по временной нетрудо­ способности

по среднему заработку (например, для оплаты командировок)

по месячному расчетному показателю за фактический период действия начисления

процентом (от сумм, начисленных по указанным базовым начислениям) Удержания могут рассчитываться следующими способами:

* процентом (от сумм, начисленных по указанным базовым начисле­ ниям)
* для удержания по исполнительным документам (т.е. с предвари­ тельным уменьшением суммы базовых начислений на сумму нало­ гов)
* фиксированной суммой

Расчёт заработной платыпроизводилсяисходя из количества отработанных часов, разряда, тарифной ставки, оклада работников, профсоюзных взносов, с учетом доплат завыполнение обязанностей временно отсутствующих работни­ ков, за работу в праздничные и выходные дни, а также премий.

Расчет НДФЛ производился по основной ставке, установленной в размере 13%, без учета имущественных вычетов на иждивенцев.

Проектируемая автоматизированная система должна решать следующие задачи:

! .Кадровый учёт сотрудников.

2.Учет количества часов, отработанных сотрудниками за период. З.Формированиебазовой тарифной ставки (часовой и дневной).

1. Расчет оклада.
2. Расчёт оплаты труда за отработанные часы.
3. Расчет оплаты труда за изготовленные изделия
4. Расчёт доплат за выполнение обязанностей временно отсутствующих работников, за работу в праздничные и выходные дни.
5. Расчет пособия по временной нетрудоспособности
6. Расчет сумм отпускных 10.Расчёт премий.
7. Расчёт начисленной заработной платы.
8. Расчет НДФЛ

12.Расчёт прочих удержаний. 13.Расчёт заработной платы к выдаче.

14. Учет затрат на оплату труда

Для разработки системы автоматизации вначаленеобходимопроанализи­ ровать данные, обязательныедля решения поставленных задач, т.е. составить пе­ речень данных, необходимых для еёрешения.

Сначала производятся расчеты всех начислений заработной платы, затем рассчитываются все удержания из заработной платы, после этого определяются суммы к выдаче на руки сотрудникам.

При этом часть данных является условно-постоянной информацией: кад­ ровые данные сотрудников; базовая тарифная ставка; виды и ставки доплат; та­ рифные коэффициенты; ставки премии.

Другая часть данные является переменной: количество отработанных ча­ сов; расчёт оплаты труда за отработанные часы; расчёт доплат за выполнение обязанностей временно отсутствующих работников, за работу в праздничные и выходные дни; расчёт премий; расчёт начисленной заработной платы; расчёт профсоюзных взносов, НДФЛ, заработной платы к выдаче.

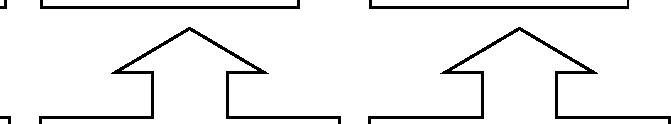
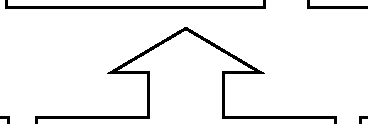
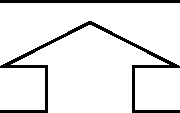
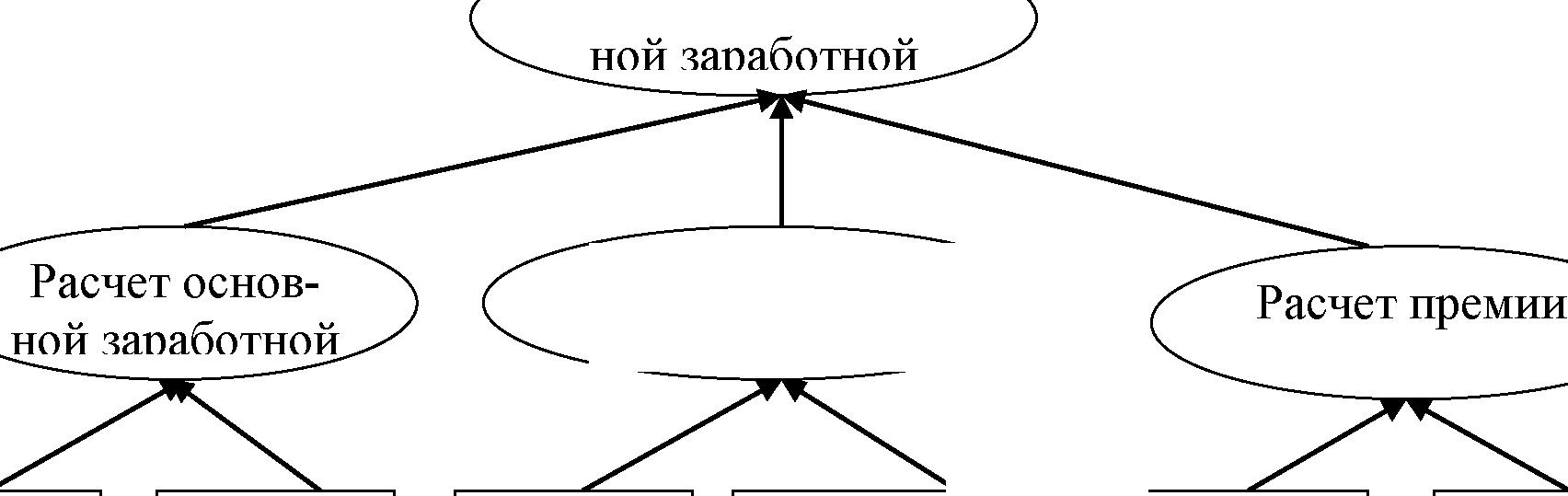
Последовательность выполнения задач рассмотрим в виде схемы пред­ ставленной на рисунке 4.

Расчет заработной

платы к вы.паче

Расчет НДФЛ и

пnvгих vпеnжа-

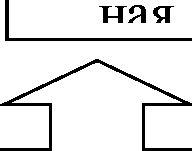


Расчет начислен-

Расчет дополни­

тельной 1аnабот-

Тариф-



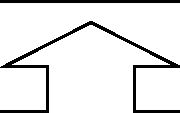
Тарифно­ квалификаци­ онный спnя-

Отрабо-

**ТЯННl'-.ТР**

Средняя Дни, подле-

яnпття- ЖЯПТТТР ()ПТТЯТР



Норматив- Табель учета но- рабочего вре-

Табель учета рабочего вре­

мени

пnяноные мени

Тариф-

н- :..т-й я-

Норматив­ но­ пnяноные

Ставка

ППf'l\lrTTTT

Норматив­ но­ пnяноные

Рисунок 8. Графическое представление последовательности задач

В качестве выходного документа по начислению заработной платы работ­ никам в ТОО «Вершина» используется расчетная ведомость, которая представ­ ляется на бумажном носителе.

Унифицированная форма первичной учетной документации N Т-51 «Рас­ четная ведомость» (ОКУД 0301010) утверждена постановлением Госкомстата России от 05 января 2004 г. № 1. Данная ведомость применяется для расчета за­ работной платы работникам организации. Ведомость составляется бухгалтерией в одном экземпляре.

Начисление заработной платы производится на основании данных пер­ вичных документов по учету выработки, фактически отработанного времени и других документов.

В расчетной ведомости заполняются все предусмотренные показатели. В каждую графу вписывается только один показатель. В случае отсутствия каких­ либо показателей, предусмотренных в расчетной ведомости, в соответствующей строке (графе) ставится прочерк.

Расчетная ведомость составляется ежемесячно не позднее 10 числа меся­ ца, следующего за расчетным и заполняется в рублях. Расчетную ведомость подписывает руководитель и главный бухгалтер организации (обособленного подразделения, исполняющего обязанности юридического лица по уплате нало­ гов и сборов) с указанием Ф.И.О.

Расчетную ведомость по форме № Т-51 можно составить по организации в целом (по всем работникам), по конкретному подразделению (по работникам данного подразделения) или по группе работников. Работников в расчетной ве­ домости можно упорядочить по табельным номерам или в алфавитном порядке, а для каждого подразделения составить свою расчетную ведомость.

Данные расчетной ведомости используются на последующих этапах учет­ ной работы для составления сводных специализированных регистров по учету расчетов по оплате труда.

Поля расчетной ведомости заполняются по следующим правилам (по но­ мерам колонок):

1. № п./п - порядковый номер записи в ведомости;
2. Табельный номер - заполняется на основании сведений, указанных в личных карточках работников (форма № Т-2);
3. Фамилия, имя, отчество работника - используются сведения о работни- ке;
4. Профессия или должность - заполняется на основании сведений, ука­ занных в личных карточках;
5. Размер тарифной ставки, оклада -указывается, если для работника установлена оплата исходя из тарифной ставки или должностного оклада;
6. Отработано дней (часов) рабочих - общее количество фактически про­ работанных в расчетном периоде дней, указывается на основании табе­ ля учета рабочего времени;
7. Отработано дней (часов) выходных и праздничных дней - заполняется в том случае, если работник работал в выходные или праздничные дни на основании табеля учета рабочего времени;
8. Начислено за текущий месяц - указываются начисления за текущий ме­ сяц по видам оплат: «повременно», «сдельно», «прочее»;
9. Начислено всего - отражается общая сумма начислений за месяц;
10. Налог на доходы физических лиц - показывается сумма налога, подле­ жащего удержанию с работника в текущем месяце. Эта сумма равна разнице между суммой налога, исчисленной с начала календарного го­ да по текущий месяц включительно, и суммой налога, исчисленной с начала календарного года по расчету за предыдущий месяц;
11. Прочие удержания - указываются суммы прочих удержаний из зара­ ботной платы работника: в счет погашения ссуды, алиментов на со­ держание несовершеннолетних детей и почтовых расходов на их пере­ вод, профсоюзных членских взносов, удержаний по иным исполни­ тельным документам. Здесь же показываются суммы, перечисляемые

организацией из заработной платы работника по его заявлению (на благотворительные цели, заработная плата, переводимая в учреждение банка и т. д.);

1. Удержано всего - указывается сумма значений граф 1О и 11;
2. Сумма задолженности за организацией - показывается долг за органи­ зацией по результатам расчета за предыдущие периоды;
3. Сумма задолженности за работниками - показывается задолженность работника по результатам расчета за предыдущие периоды;
4. Сумма к выплате - если разница между начисленной суммой (графа 9) и суммой, которая подлежит удержанию и зачету (графа 12), скоррек­ тированная на суммы, указанные в графах 13 или 14 положительная, то она показывается в данной графе

Для выплаты заработной платы сотрудникам используется платежная ве­ домость. В ней указывают фамилии, инициалы работников, их табельные номе­ ра, суммы к выдаче и расписку в получении заработной платы.

# Обоснование необходимости использования вычислитель­ ной техники для решения задачи

Расчет начисления заработной платы и удержаний налога на доходы фи­ зических лиц связан с обработкой больших объемов информации. Хотя количе­ ство первичных документов невелико, но объем информации, содержащейся в них весьма значительный. Рассмотрим график документооборота

-

>--

-

1

+

**Расчетная ведо11ость**

•1

**1**

**t**

**Расчетная ведо11ость**

**1**

1

**Расчетная ведо11ость**

**1**

-1

**Отчет ость**

**1**

**б. Составление отчетности**

**5.Расчетудержаний**

**4. Расчет доплат**

**3. Расчет повременнойоплаты труда**

**2.Расчет сдельнойоплаты труда**

**Наряд на сдельную работу**

**Нарядна сдельную работу**

**2. Учетколичества изготовленныхизделий, выполненныхработ**

**абельучета рабочего вре11е и**

**абельучета рабочего времени**

**1.Учетрабочеговремени**

**Бухгалтер**

**Администратор бригады**

**е**

**Расчетная ведо11ость**

**1**

Рисунок 9. График документооборота расчета оплаты труда

# Анализ системы обеспечения информационной безопасно­ сти и защиты информации

В настоящее время компании редко используют безопасные системы управления базами данных. В качестве основной причины этого можно назвать то, что потребители недооценивают серьезность последствий повреждения или уничтожения данных, а высокая стоимость систем безопасности снижает жела­ ние приобрести безопасные СУБД. Системы безопасности, как правило, разраба­ тываются оборонными структурами. Интересы оборонных и коммерческих структур не совпадают: оборонные структуры опасаются утечки секретных дан-

пых, а коммерческие структуры - разрушения информационных систем и ущерба для бизнеса.

Современные СУБД представляют собой систему мониторинга, контроли­ рующую матрицу полномочий. Вданной матрице колонки представляют собой информационные объекты, а строки - пользователей. В ячейках определяются полномочия. Подобная система безопасности не препятствует передачеполномо­ чий между пользователями, но не обеспечивает достаточной безопасности. Сво­ бодная передача полномочий между пользователями позволяет пользователю с более высоким грифом секретности передать данные пользователю с более низ­ ким уровнем секретности. В этой связи в настоящее время разрабатываются многоуровневые схемы защиты. Многоуровневая система защиты базируется на следующих принципах, отраженных ниже:

1. все данные имеют определенный гриф секретности (совершенно сек­ ретно, секретно, для служебного пользования, не секретно), каждый пользова­ тель принадлежит определенному классу допуска в соответствии с грифом, ко­ торый он имеет, материалы могут быть снабжены специальными пометками, например, «не для иностранных представителей»;
2. классы допуска имеют обратное наследование, то есть класс «совер­ шенно секретно» имеет доступ к документам «секретно»;
3. пользователи с высоким классом допуска могут читать объекты, имею­ щие более низкий гриф секретности, но они не имеют права создавать свои объ­ екты в рабочих областях пользователей с более низким классом допуска. Дан­ ный принцип предотвращает появление «тайных каналов» передачи полномочий от лиц с высшим классом допуска к лицам с низшим классом допуска.

Выполнение принципов многоуровневой модели расширяет представле­ ние о реляционной таблице. Реляционная модель должна стать многоэкземпляр­ ной.

Для программного обеспечения безопасности и защиты информации ру­ ководством предприятия принято решение применить продукты компании «Ла­ боратория Касперского».

Kaspersky Enpoint обеспечивает комплексную защиту сети от информаци­ онных угроз, не препятствуя при этом нормальной работе защищаемых узлов.

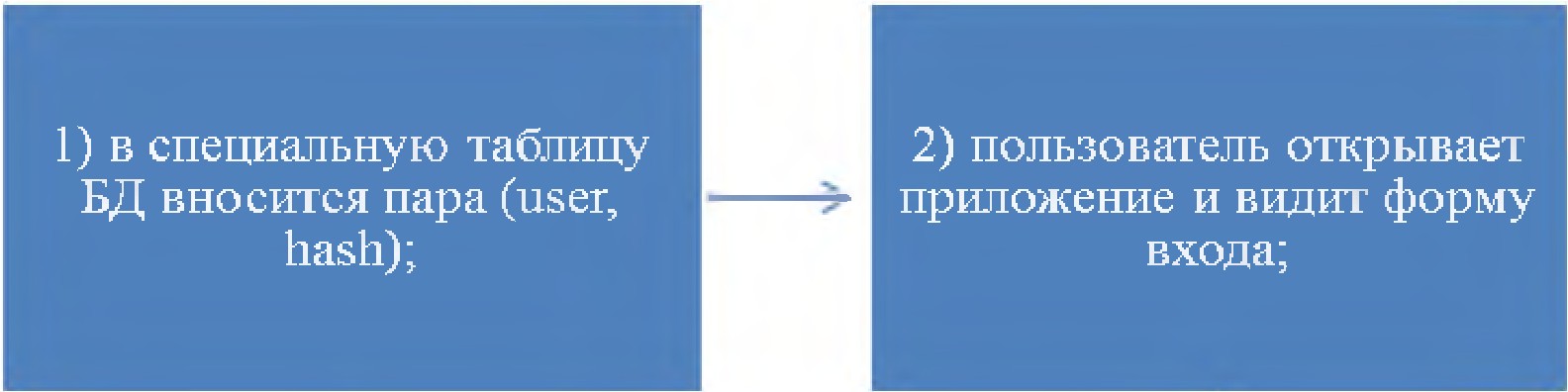
Важнейшей частью приложения является механизм аутентификации поль­ зователей, поскольку зачастую информация, хранимая в БД, предназначена для ограниченного круга лиц.

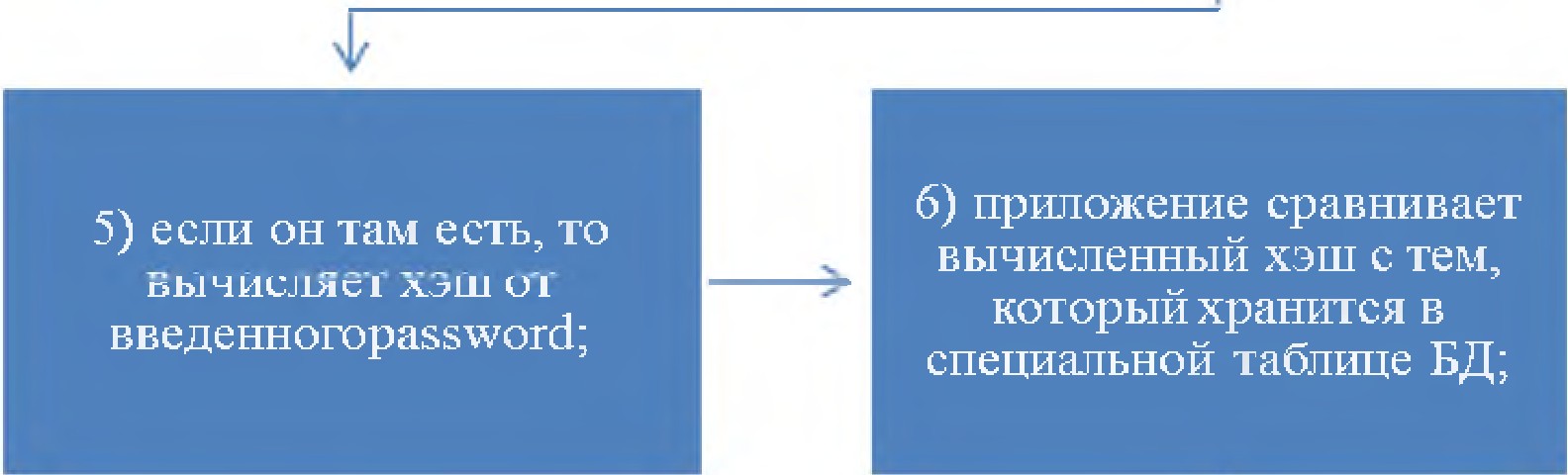
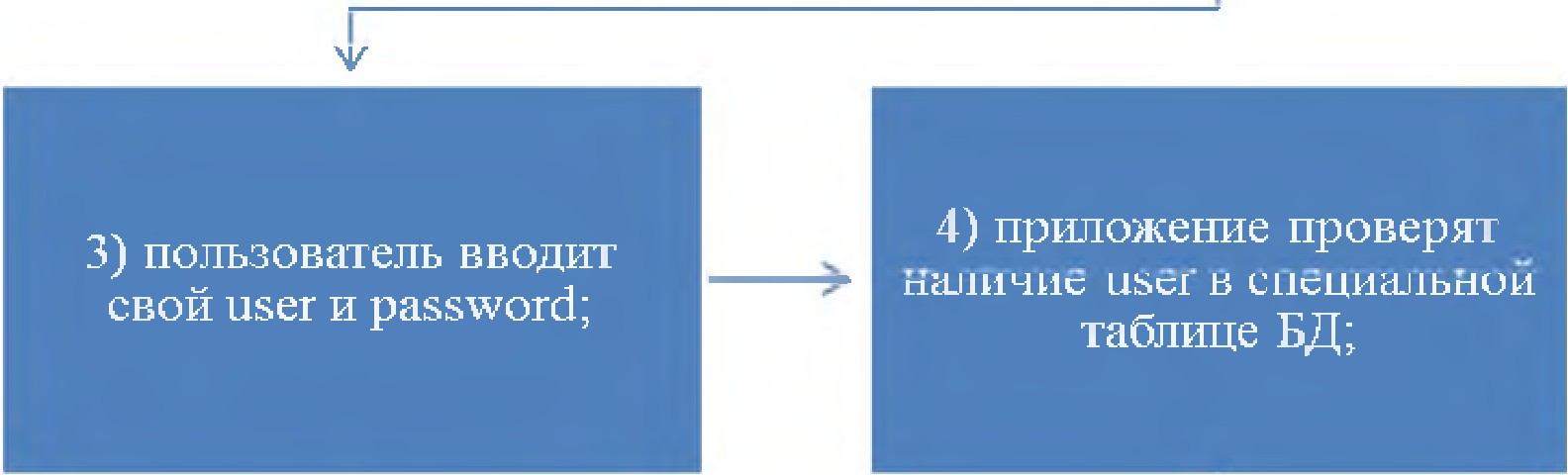
Самый простой подход к построению системы аутентификации заключа­ ется в сохранении паролей в теле программы, то есть ключи «зашиты» в исход­ ный код программного средства. Данный подход хоть и прост, но с точки зрения безопасности информационных систем неприемлем, поскольку даже начинаю­ щий реверс-инженер или исследователь без труда найдет «спрятанный» ключ. Даже если этот ключ будет разнесен по разным участкам программы, то это лишь несколько усложнит работу аналитика, который в конечном итоге обяза­ тельно взломает систему.

Наиболее оптимальным подходом для авторизации в подобных программ­ ных средствах является создание дополнительной таблицы, в которой хранятся имена пользователей и пароли. Необходимо отметить, что файл БД (mdf-файл) невозможно корректно прочитать с помощью текстового редактора, поскольку он имеет особую структуру построения. Тем не менее, и этот факт не может га­ рантировать полную безопасность.

Самый простой и эффективный способ преодоления этой проблемы - со­ хранение паролей не в открытом виде, а виде некоторого значения, которое вы­ числено от них. В качестве широко используемых алгоритмов можно назвать следующие наиболее популярные алгоритмы: MD-4, MD-5, MD-6, SHA-1 и дру­ гие. Указанные хэширующие алгоритмы необходимы в целях генерации уни­ кальной последовательности, фиксированной от длины от произвольного набора байт, в том числе и от паролей. Замечательны они тем, что, получив некоторый хэш от пароля, восстановить пароль по данному хэшу не представляется воз­ можным. Стоит отметить, что не рекомендуется использовать свои хэширующие алгоритмы, поскольку это может снизить безопасность системы.

Аутентификация с помощью специальной таблицы при использовании хэширующего преобразования выглядит следующим образом:





7) в случае равенства, пользователь допускается к работе с БД, в проrnвном

случае прruюжение закрывается.

РисунокlО. Порядок аутентификации

В качестве хэширующего алгоритма алгоритм МD-5является на настоя­ щий момент стандартом де-факто в области вычисления хэш-значений от набо­ ров байт.

* 1. **Анализ существующих разработок и выбор стратегии авто­ матизации «процесса начисления заработной платы, исчислению ин­ дивидуального подоходного налога и исчислению социального налога ТОО «Вершина»**

**дачи**

* + 1. **Анализ существующих разработок для автоматизации за­**

На рынке программного обеспечения в настоящее время имеется большое

количество различных программных продуктов, предназначены для автоматиза­ ции учета оплаты труда. Наиболее распространенными являютсяlС: - 65% • БЭСТ - 7% • Инфо-бухгалтер - 6% • Парус - 3% • Турбо-бухгалтер - 3% • Сред­ ства MS Office - 3% • Галактика - 2% • Самописная - 2% • Другое - 9.

Конфигурация «IC: Зарплата и Управление персоналом» применяется для комплексной автоматизации кадрового учета и расчета оплаты труда на малых и средних предприятиях, в том числе имеющих сложную юридическую структуру. Данная конфигурация ориентирована, прежде всего, на задачи кадровых служб и бухгалтерии, которые ведут регламентированный учет и расчеты с персоналом.

Система «Инфо-Бухгалтер» является интегрированной системой автома­ тизации бухгалтерского учета. Данная система объединяет в рамках одного ра­ бочего места различные возможности учета. Она предназначена для автоматиза­ ции бухгалтерского учета и делопроизводства предприятий различных форм собственности, позволяет вести бухгалтерский учет для нескольких организаций на одном компьютере, обеспечивает ведение синтетического и аналитического учета, как в денежном, так и в количественном выражении. Особенностью рас­ сматриваемой системы является понятие «План бухгалтерии», включающее в себя не только стандартный план счетов бухгалтерского учета, но и настройки типовых операций, справочников аналитики и другие виды настроек.

Подсистемами оперативного учета в программе являются: «Склад», «Зар­ плата», «Касса», «Банк» и др. Совершаемые хозяйственные операции регистри­ руются в названных подсистемах в форме операционных документов, а также в форме первичных документов.

«БухСофт: Зарплата, Табель, Кадры» программа для кадрового, табельно­ го и персонифицированного учета, расчета зарплаты на предприятии со всеми необходимыми возможностями и с автоматическим формированием первичных и отчетных документов.

Таблица 2 Функциональные возможности систем автоматизации расчета заработной платы

|  |  |  |  |
| --- | --- | --- | --- |
| **Показатель** | **«БухСофт: Зар- плата, Табель, Кадры»** | **«lC: Зарплата и Управление персо- налом»** | **«Инфо-Бухгалтер: Зарплата и кадро- вый учет»** |
| Совместное использование в локальной сети | - | + | + |
| Планирование потребно- стей в персонале | + | - | - |
| Отражение начисленной зарплаты и налогов в за- тратах предприятия | - | - | - |
| Стоимость лицензии на одно рабочее место | 16 ООО рублей в год | 17 ООО руб. на одно место | 36 300 руб. сетевая версия |

Исходя из проведенного анализа, можно сделать вывод, что программное решение, хоть и является комплексным и, на первый взгляд, достаточным, воз­ никает множество вопросов, по наладке и обучению персонала, при постоянном развитии компании и вводом новых дочерних организаций и соответственно пользователей системы. Кроме того, даже наиболее распространенная система 1С имеет существенные недостатки, такие как: невозможность ведения учета в динамике, Невысокая скорость проведения вычислений, не является универ­ сальной, а также лишена возможности работы с графикой. В этой связи целесо­ образно разработать собственную, легко адаптируемую программу способную интегрироваться с используемой в центральном офисе системы 1С.

## Выбор и обоснование стратегии автоматизации задачи

В настоящее время многие компании стараются приобретать готовые па­ кеты и технологии, а если необходимо, то добавлять к ним свое программное обеспечение, так как разработка собственных информационных систем и ин­ формационных технологий связана с высокими затратами и риском. Эта тенден­ ция привела к тому, что поставщики информационных систем изменили ранее существовавший способ выхода на рынок. В настоящее время, как правило, раз-

рабатывается и предлагается базовая система, которая затем адаптируется в со­ ответствии с индивидуальнымипожеланиями клиентов. При этом пользователям предоставляются консультации, помогающие минимизировать сроки внедрения систем и технологий, наиболее эффективно их использовать, повысить квалифи­ кацию персонала. Тем не менее, последнее тоже связано с риском, так как обу­ ченный персонал тоже может покинуть компанию.

Можно выделить четыре вида стратегии автоматизации:

* хаотичная автоматизация;
* автоматизация по участкам;
* автоматизация по направлениям;
* комплексная автоматизация.

В нашем случае будет выбрана стратегия автоматизации по участкам, так как предполагается модификация имеющейся на предприятииавтоматизирован­ ной системы расчета и начисления заработной платы для автоматизированного выполнения учетных операций.

# Выбор и обоснование способа приобретения ИС для авто­ матизации задачи

Действия пользователя, начиная с момента определения и формализации решения о необходимости приобретения информационной системы до момента внедрения информационной системы, называется способ приобретения инфор­ мационной системы. Можно выделить следующие способы приобретения ин­ формационной системы:

разработка покупка

покупка + доработка

аренда

Рассмотрим перечисленные выше способы на предмет его применения к автоматизации поставленной задачи.

При покупке готового решения нужно выбрать такое решение, которое будет полностью соответствовать схемам расчета оплаты труда, применяемых

ТОО «Вершина». Это является весьма сложным по причине того, что произво­ дители систем автоматизации пытаются удовлетворить требования всех потен­ циальных клиентов. В результате готовые системы или обладают лишними функциями, которые никогда не будут востребованы пользователем, или систе­ ма представляет собой каркас, который необходимо дорабатывать до нужд кон­ кретного пользователя. Оба эти варианта экономически не выгодны для автома­ тизации поставленных задач, поэтому не подходят для ТОО «Вершина».

Более того, готовая информационная система зачастую хуже приспосаб­ ливается к нуждам конкретного предприятия, в ней не предусмотрены возмож­ ности для будушего информационной системы, имеет длительный срок адапта­ ции под нужды конкретной организации, высокие издержки на обслуживание.

При заказе разработки информационной системы у стороннего разработ­ чика можно получить информационную систему, которая полностью соответ­ ствует бизнес-правилам, действующим ворганизации в настоящее время. Тем не менее, при изменении бизнес-правил потребуется переработка информационной системы приведет к значительным затратам.

Покупка и доработка являются хорошим вариантом, если требует неболь­ шой объем доработки. Тем не менее, данная доработка бывает невозможна в рамках модели ядра информационной системы из-за отсутствия подробной до­ кументации.

Аренда информационной системы имеет существенные преимущества:



Тем не менее, в случае аренды информационной системыорганизация рискует лишиться части собственных ресурсов и возможностей, а также вывести за свои пределы многие виды деятельности. Также значительно возрастает риск разглашения конфиденциальных данных.

В этой связи информационная система будет разрабатываться в ТОО

«Вершина» собственными силами, так как организация располагает квалифици­ рованными сотрудниками для разработки **и** внедрения данной информационной системы. Разработка собственной системы предполагает наличие у предприятия специалистов, способных провести анализ бизнес-процессов, разработать и внедрить систему их автоматизации.

Разрабатываемая информационная система будет включать весь необхо­ димый функционал, и учитывать все особенность работы **и** пожелания сотруд­ ников данной компании, так как будет являться индивидуальным (уникальным) продуктом, что позволяет быстро изменить функциональность в соответствии с требованием внутреннего заказчика.

Для разработки приложения понадобятся следующие инструменты:

клиентской машине

Операционная система Windows предназначенная для функционирования на

Операционная система Windows серверного типа, предназначенная для функционирования на сервере и обеспечивающая многопользовательское подключение к клиент-серверной СУБД

системы

многопользовательской работы в клиент-серверном варианте функционирования

СУБД MS SQL Server для обеспечения хранения базы данных и

# Обоснование проектных решений

# Обоснование проектных решений по информационному обеспечению

Под информационным обеспечением понимают совокупность данных, ко­ торая представлена в определенной форме для компьютерной обработки.

Кроме того, под информационным обеспечением информационной систе­ мы понимают также совокупность единой системы классификации и кодирова­ ния информации, унифицированных систем документации и информационных массивов [12].

Информационное обеспечение состоит из двух видов компонентов: ком­ поненты внемашинного и внутримашинного информационного обеспечения. К внемашинному информационному обеспечению относят различные классифика­ торы технико-экономической информации и документы, а к внутримашинному - макеты и экранные формы для ввода первичных данных или вывода результат­ ной информации, структуру информационной базы, состояющей из входных, выходных файлов, базы данных.

В целях приспособления экономической информации чтобы эффективно­ го проводить процессы поиска, обработки и передачи по каналам связи, ее необ­ ходимо представить в цифровом виде. Для этого ее нужно сперва классифициро­ вать (упорядочить), а затем закодировать (формализовать) с использованием классификатора.

Под классификатором понимают документ, при помощи которого осу­ ществляется формализованное описание экономической информации в элек­ тронной информационной системе, которая содержит названия объектов, клас­ сификационных группировок и их кодовые обозначения [13].

Классификаторы деляться по сфере действия на международные, общего­ сударственные (общесистемные ), отраслевые и локальные.

В качестве основного компонента внемашинного информационного обес­ печения можно выделить применяемую в процессе управления экономическим объектом систему документации. В части внутримашинного информационного обеспечения основным компонентом является описание экранных форм для вво­ да и вывода информации.

Под документом понимают определенную совокупность данных, исполь­ зуемую для решения экономических задач, расположенную на материальном но­ сителе в соответствии с установленной формой.

Под системой документации понимают совокупность взаимосвязанных форм документов, которые регулярно используются для управления экономиче­ ским объектом. В качестве отличительной черты системы экономической доку­ ментации можно назвать большое разнообразие видов документов. [13]

В число основных документов, используемых для формирования постоян­ ной информации, можно отнести карточки кадрового учета, отражающие посто­ янную информацию о каждом сотруднике. Данную информацию следует хра­ нить в справочных таблицах. Вместе с тем, в справочных таблицах следует хра­ нить информацию о видах начислений и удержаний, разрядах, тарифных став­ ках, процентах начислений и т.п.

В процессе проектировании экранных форм для ввода с постоянной ин­ формации необходимо помнить, что данные макеты используются для ввода и актуализации записей информационной базы, в этой связи для их проектирова­ ния целесообразно применить анкетную форму расположения реквизитов, кото­ рая будет удобна в целях выполнения этих операций.

На основании табеля учета рабочего времени вносится информация об от­ работанном времени. Информацию из листков на доплату, актов, приказов вно­ сят в целях начисления доплат и дополнительной заработной платы. Данная ин­ формация считается переменной и должна обновляться каждый месяц.

В процессе разработки структуры экранных форм для ввода данных пер­ вичных документов с переменной информацией следует использовать комбини­ рованную форму, которая будет максимально приближена к той форме, что была использована для построения самого документа. Поля формы следует распола­ гать в последовательности, которая соответствует логической структуре доку­ мента и файлов с оперативной информацией. Соответствующая структура экранной формы будет сокращать трудоемкость операций ввода информации в информационную базу.

В основе выбора экранных форм закладываются следующие принципы:

минимальной трудоемкости и стоимости ввода информации в компьютер,

экран,

максимальной степени читабельности результатной информации, выводимой на

максимальной надежности и достоверности выполнения операций.

В качестве основной части внутримашинного информационного обеспе­ чения выступает информационная база.

Под информационой базой (сокращенно - ИБ) понимают определенным образом организованную совокупность данных, которая храниться в памяти вы­ числительной системы в форме файлов, при помощи которых удовлетворяются информационные потребности управленческих процессов и решаемых задач [З].

Можно выделить следующие способы организации информационной ба­

зы:

прикладных программ;

совокупность локальных файлов - поддерживается функциональными пакетами

т.е. СУБД.

универсальных программных средств загрузки, хранения, поиска и ведения данных,

интегрированная база данных - основывается на использовании

Множественное дублирование данных в информационной системе ведет к несогласованности данных в разных приложениях, сложности доступа к инфор­ мации, поэтому может применяться только в специализированных приложениях. Это является проблемой для организации локальных файлов.

Под базой данных (сокращенно - БД) понимают поименованную совокуп­ ность данных, которая отражает совокупность объектов и их отношений в рас­ сматриваемой предметной области [З].

В качестве основных способов организации баз данных можно выделить два способва: создание централизованных и распределенных баз данных. Способ организации информационной базы выбирается по следующему критерию: при

помощи которого из них будет достигнуты минимальные трудовые и стоимост­ ные затраты на проектирование структуры информационной базы, программно­ го обеспечения системы, системы ведения файлов. Посредством анализа данного критеория, и кроме того, в целях обеспечения надежности хранения данных, вы­ бран централизованный способ организации базы данных.

По способу установления связей между данными выделяют три вида мо­

делей:

Иерархическая

установления связей

способам

Виды моделей по

.........

Реляционная Сетевая

Файлы и таблицы являются основными компонентами каждой из выше­ указанных моделей.

Под иерархическими моделями данных понимают графовую модель с вершинами-таблицами. В таких моделях есть один файл, являющийся входом в сответствующую структуру. Между файлами в данной модели формируются от­ ношения соподчиненности. Один файл может иметь одну исходную вершину и несколько подчиненных. В качестве основного типа отношений можно назвать

**ОДИН КО МНОГИМ.**

*Что* касается сетевых моделей, то в них каждый файл может стать точкой входа в систему, и может быть связан со всяким числом других файлов. Типы отношений в сетевых моделях: один к одному, один ко многим и многие ко мно­ **гим.**

Реляционная модель данных является наиболее популярной и широко распространенной. При такой организации вся информация представлена в виде файлов базы данных (таблицы) и отношений. Таблицы - совокупность записей

(строк, кортежей). Тип связей между таблицами - типа один ко многим, многие ко многим. Каждое отношение обладает ключом, являющимся полем записи (ат­ рибут), который однозначно идентифицирет ее. Именно это свойство реляцион­ ной модели данных элиминирует дублирование информации, ускоряет поиск и доступ к конкретным данным.

Подход к структурированию и целостности данных, который принят в ре­ ляционной модели, позволяет удобно организовать и упорядочить процесс про­ ектирования и реализации сложных баз данных, а реляционные операции обла­ дают мощными возможностями управления данными и их обработки.

Принимая во внимание все преимущества вышеназванной реляционных модели данных в целях представления информации, которая обрабатывается при решении задачи, целесообразным будет являться использование именно данного типа моделей базы данных.

Система, которая разрабатывается в проекте, является предназначенной в целях работы в операционной системе Windows. Необходио, чтобы программа работала в любой операционной системе семейства Windows: Windows7/8/8.l/10, и кроме того на серверных вариантах данных операционных систем. Выбор дан­ ной операционной системы обусловлен развитым удобным графическим интер­ фейсом пользователя данного семейства операционных систем, и ее распростра­ ненностью на отечественном рынке. Большинство фирм-производителей про­ граммного обеспечения, включая системы разработки приложений, тоже ориен­ тированы на данную группу операционных систем. В этой связи на рынке суще­ ствует большое количество систем программирования, которые позволяют раз­ рабатывать ПО для Windows.

Операционная система Windows обладает достаточно большим набором утилит и служебных программ, которые предназначены в целях администриро­ вания, управления и диагностики системы, а также прикладных приложений. Более того, данная система поддерживает обширный программный интерфейс в целях написания собственных приложений для работы в этой операционной сре­ де.

В селдствие того, что программа хранит информацию в базе данных, необходимо наличие соответствующей СУБД.

Программа имеет ориентацию на работу с базами данных формата: MS SQL Server. Поэтому данная СУБД должна быть установлена на компьютере. В том случае если база MS SQL Server функционирует на сервере, то нужно обес­ печить подключение компьютера к данному серверу. Чтобы работать с базой данных MS SQL Server на удаленном компьютере, нужен открытый общий до­ ступ к базе, которая расположена на удаленном компьютере, через протокол SMB.

Кроме того, необходимо предоставить удаленному пользователю, который подсоединяется к базе, права на чтение, запись, изменение данных на компьюте­ ре, который будет хранить базу данных. Кроме того, в целях получения доступа к данным нужна система работы с базами данных MS SQL Server версии не ниже 2012 фирмы Microsoft.

# Обоснование проектных решений по программному обес­ печению

Под программным обеспечением (ПО) понимется совокупность уомпью­ терных программ, реализующих функции и задачи ИС и обеспечивают устойчи­ вую работу задействованных комплексов технических средств.

Обычный состав программного обеспечения информационной системы следующий:

общесистемные программы

Программное обеспечение ИС

Справочно­ методические материалы

Специальные программы

Общесистемное программное обеспечение состоит из программ, ориенти­ рованных на широкий круг пользователей и предназначены для организации вы­ числительного процесса и выполнения сравнительно часто встречающихся вари­ антов обработки информации:

расширение функциональных возможностей компьютера

автоматизация процесса планирования очередности вычислительных работ оптимизация работы пользователей-программистов

Специальное (функциональное) программное обеспечение представляет собой совокупность программ, имеющих конкретные функциональные назначе­ ния. «Оно включает пакеты прикладных программ, осуществлявших организа­ цию данных и их обработку при решении функциональных задач информацион­ ной системы» [З].

При комплектовании комплекса технических средств для разработки си­ стемы, наряду с самими техническими средствами, важен выбор операционной системы, поскольку именно они управляют техническими средствами компью­ тера, поддерживает запуск и выполнение тех или иных приложений, обеспечи­ вают защиту данных, организует взамиодействие программ, выполняет различ­ ные сервисные функции. Каждая исполняющаяся программа пользуется сред­ ствами, предоставляемыми операционной системой.

На компьютерах, используемых в качестве рабочих станций пользовате­ лей, чаще всего применяются операционные системы семейства Windows от фирмы Microsoft.

При разработке, проектируемой ИС в качестве операционной среды для разработки и применения была выбрана операционная система семейства Windows, в частности операционная система Windows 10. Такой выбор обуслов­ лен тем, что на сегодняшний день Windows 1О является одной из наиболее рас­ пространенных операционных систем, и которая хорошо поддерживается и об­ новляется Microsoft. Операционная система Windows 1О обеспечивает стабиль­ ность работы, предоставляя пользователям возможность сосредоточиться на вы­ полняемой работе.

42

Важнейшим требованием, предъявляемым к проектированию информаци­ онных систем, эксплуатируемых совместно на технологической базе ограничен­ ных возможностей, является их большая однородность, которая обеспечивает:

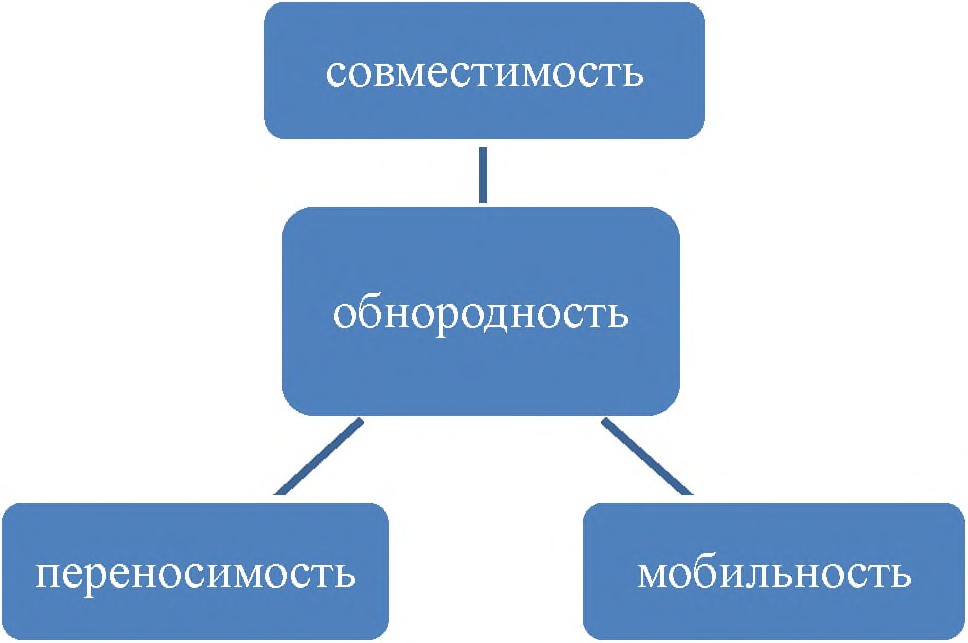
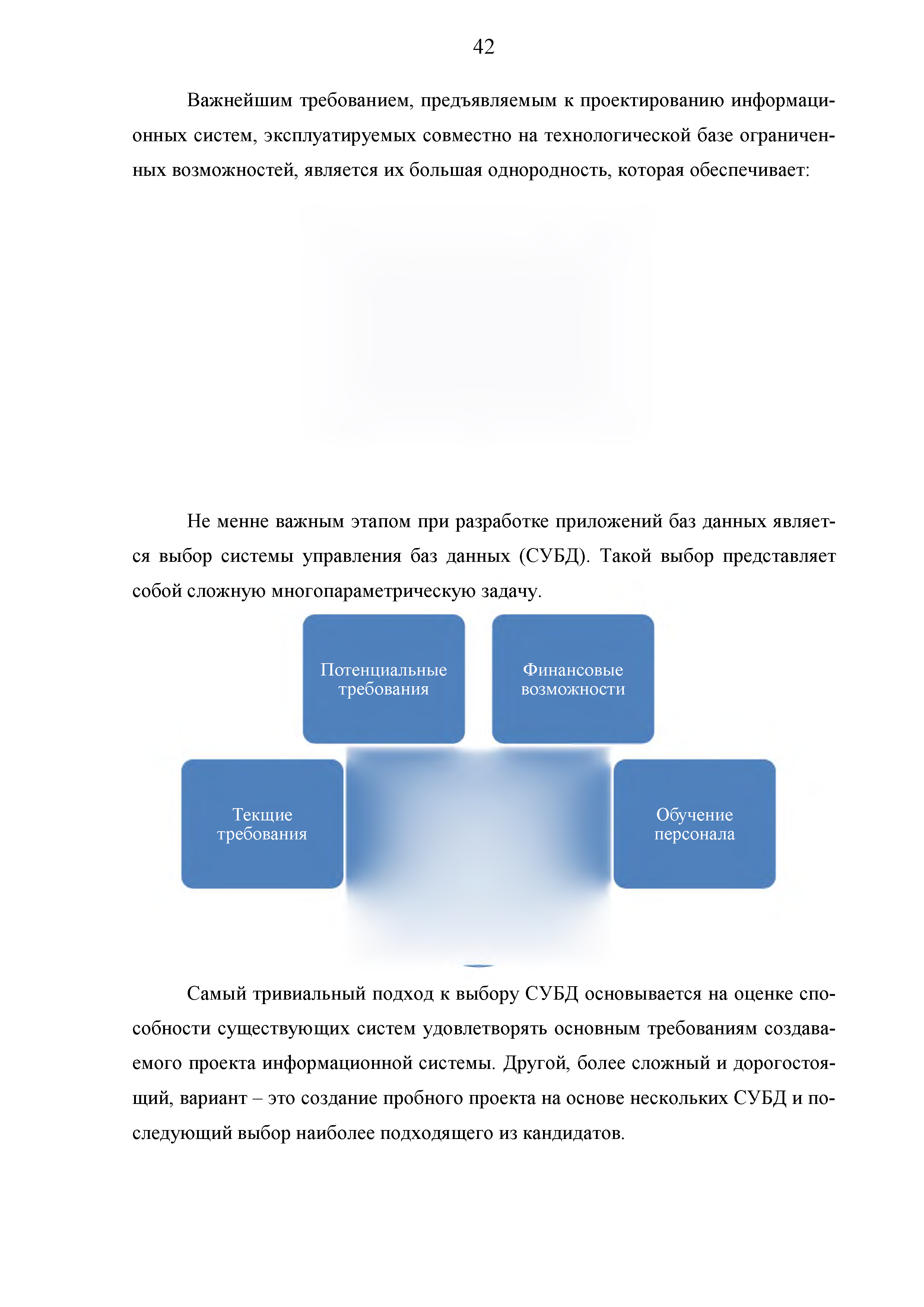
Не менне важным этапом при разработке приложений баз данных являет­ ся выбор системы управления баз данных (СУБД). Такой выбор представляет собой сложную многопараметрическую задачу.

Потенциальные требования

Финансовые возможности

Текщие требования

Обучение персонала



Самый тривиальный подход к выбору СУБД основывается на оценке спо­ собности существующих систем удовлетворять основным требованиям создава­ емого проекта информационной системы. Другой, более сложный и дорогостоя­ щий, вариант - это создание пробного проекта на основе нескольких СУБД и по­ следующий выбор наиболее подходящего из кандидатов.

Текущие и потенциальные требования можно разбить на следующие крие-

терпи:

Производительность

Моделирование данных

Надежность

Особенности разработки приложений

Особенности архитектуры и функциональные возможности

Контроль работы системы

Требования к рабочей среде

Смешанные критерии

Таким образом, выбор СУБД является трудной задачей, так как имеется большое количество требований, которые постоянно меняются. Кроме того, ин­ формация о СУБД часто носит рекламный характер, не позволяющий сделать правильное суждение.

В этой связи процедуру выбора СУБД следует проводить в три этапа:

Этапы выбора СУБД

оценка предлагаемых продуктов на пригодность оценка технических характеристик

оценка производительность

Рашифруем эти этапы:

Основные показатели пригодно1.;ти

вид программного продукта;

категории пользователей (профессионалы, админы, конечные пользователи)

===

удобство и простота использования

модель представления данных. качество средств разработки.

качество средств защиты и контроля корректности базы данных. качество коммуникационных средств.

В качестве рекомендаций отметим, что:

Рекомендуется использовать наиболее распространенную модель данных - реляционную. Перспективными являются модели с объектной ориентацией, по­ скольку они обладают большими возможностями отражения семантики пред­ метной области.

При оценке качества средств разработки учитывается следующее:

* возможности создания пользовательских интерфейсов,
* мощность языка создания программ,
* автоматизация разработки различных объектов:
  + экранных форм,
  + отчетов,
  + запросов.

Предпочтение отдается системам, имеющим полнофункциональные генераторы и обеспечивающим удобство работы пользователя.

Доступ к функциям защиты должен предусматриваться на уровне средств разработки программ и на уровне пользователя. К важнейшим функ­ циям контроля корректности относятся: обеспечение уникальности записей БД по первичному ключу, автоматический контроль целостности связей между таблицами во время выполнения операций обновления, вставки и уда­ ления записей, проверка корректности значений в БД.

При оценке качества коммуникационных средств обращают внимание на следующие свойства программных продуктов:

### Качество

поддержка сетевых протоколов,

### коммуникационных

средств

поддержка стандартных интерфейсов с Бд,

наличие средств групповой работы с информацией Бд,

способность использовать и модифицировать БД других форматов без импортирования или преобразования;

надежность фирмы - разработчика

Последний пункт - надежность фирмы-разработчика пакета, как правило, дает следующие преимущества:

высокое качество продукта,

наличие документации и методических материалов

наличие «горячей линию> для консультаций по возникающим про­ блемам

### Основные технические характерстики

общие параметры (операционная среда, потребность в оперативной памяти, ограничения на максимальный объем БД и др.);

ограничения на операции над данными;

типы данных;

возможности средств формулировки и выполнения запросов; работа в многопользовательских средах;

инструментальные средства разработки приложений; импорт и экспорт.

**Оценка производительности** производится методом тестирования с по­ мощью эталонных тестов из набора АSЗАР (ANSI SQL Standard ScalaЫe and PortaЫe). В них контролируется широкий спектр часто встречающихся операций БД и моделируются однопользовательские и многопользовательские среды.

Ниже, в таблице приведена сравнительная таблица трех распространенных систем управления базами данных, конкурирующих на рынке программного обеспечения по основным показателям.

**с.равнение СУБД**

**Таблица 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **IПоказатели** | **Microsoft SQL Server** | **MySQL** | **Post2reSQL** |
| 1 | 2 | 3 | 4 |
| Ооддерживаемые опе- рационные системы | Windows Desktop/Server | Windows Desk-  top/Server , Linux, Unix, Мае | WindowsDesk- top/Server,Linux,  Unix, Мае |
| 1Условия лицензирова-  **ния** | Коммерческий продукт с закрытым исходным ко- дом. Есть бесплатная вер-  сия с ограничением опера-  тивной памяти до 4 Гб. | Коммерческая лицензия иGNUGPL. | Лицензия ВSD Open Source. |
| Наличие предустанов- енных драйверов  в ОС семейства  №indows | Да | Нет | Нет |
| Наличие драйверов  ODBC, JDBC, ADO.NET | Да | Да | Да |
| Поддержка репликации | Да, встроенная и разных типов. Но внесение струк- турных изменений после  начала репликации - очень сложный процесс. | Да, включая mater-master ре- пликацию. | Да, но с помощью сторонних продук- тов с открытым ис- ходным кодом. **Ре-** пликация всех ти-  ПОВ. |
| 8озможность писать ранимые функции  на разных языках про- граммирования | Да, теоретически на любом языке, поддерживающем **CLR,** например VisualВasic.NET, С#, IronPython, но сначала надо скомпилировать код  в библиотеку dll. | Нет (кроме С и Pl/SQL) | Да, наиболее полная поддержка из всех рассматриваемых. |
| 8озможность создавать  пользовательские агре- гированные функции | Да - любой .NET язык, кроме TRANSACT SQL. | Да, только на С | Да - на PL language и встроенных  С, SQL, PLPgSQL. |
| !Разграничение доступа  к столбцам | Да | Да | Да |
| Поддержка DISTINCT  ON | Нет | Нет | Да |
| Ооддержка WIТH  OLLUP | Да | Да | Да |
| Поддержка WIТH CUBE | Да | Нет | Нет |
| ПоддержкафункцийОV  E:R..PARТIТION ВУ | Да | Нет | Да, причем лучше,  чемвМS SQL |
| Ооддержка даты  и времени | Да | Да (но без вре-  менной зоны) | Да |
| lА\_утентификация | Средствами  БД и ActiveDirectory | Средствами БД | Много разных мета-  дов, включающих предыдущие |

**Продолжение таблицы 3.**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Поддержка рекурсив- ных запросов | Да | Нет | Да |
| Производительность планировщика запросов  цля сложных запросов | Средняя (умеет параллель- ные запросы «из коробки») | Очень хорошая | Плохая |

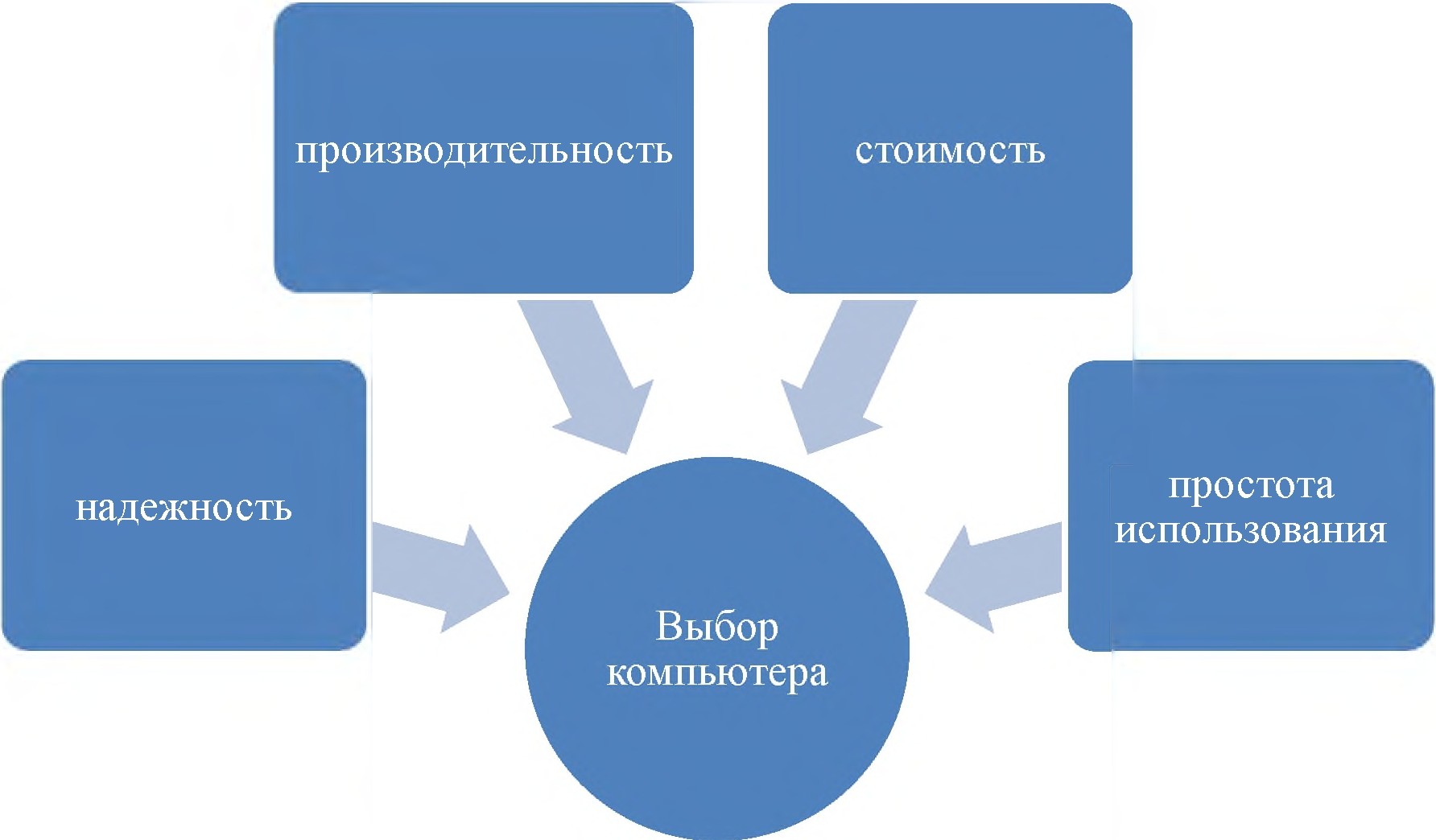
Таким образом, для проекта, рассматриваемого в данном дипломном про­ екте наиболее приемлема СУБД MS SQL.

# Обоснование проектных решений по техническому обеспе­ чению

Техническое обеспечение представляет собой комплекс технических средств, предназначенных для работы информационной системы, а также доку­ ментация на эти средства и технологические процессы.

Комплекс технических средств **ИС:**

* + - * компьютеры;
      * устройства сбора, хранения, обработки и вывода информации;
      * средства передачи данных;
      * эксплуатационные материалы.



Для каждого из элементов данной схемы выделяется перечень критериев, наиболее важных при осуществлении выбора технического обеспечения. Эти критерии таковы:

тактовая частота процессора; разрешение монитора;

объем оперативной памяти.

Автоматизированная система расчета заработной платы, НДФЛ и соци­ ального налога предназначена для функционирования в операционной системе Windows, поэтому для эффективной работы разрабатываемой системы необхо­ димо обеспечить такие требования к архитектуре и параметрам технических средств, которые позволили бы эффективно функционировать самой операцион­ ной системе.

Автоматизированная система расчета заработной платы предназначена для функционирования на компьютере типа IBMPC с техническими характери­ стиками не менее:

Intel Core 2 Duo 2500 МГц,

ОЗУ -4 ГБ,

видеокарта Geforce 9800 (1 Гб), HDD-500 GB,

17'' ЖК - мониторы,

Стандартная клавиатура, оптическая мышь.

Для печати отчетов и выходных документов необходим принтер, совме­ стимый с компьютером вышеперечисленной комплектации.

В случае варианта многопользовательской работы с системой понадобится использование одного ПК в качестве файлового сервера для доступа к базе. Це­ лесообразно разместить эту базу данных на сервере, а с клиентских компьютеров осуществлять подключение к ней через протокол smb или через ODBC поверх smb. В качестве сервера может использоваться компьютер типа IBMPCc харак­ теристиками не менее:

Intel Core 2 Quadra 2500 МГц, ОЗУ - 16 ГБ,

видеокарта Geforce 9800 (1 Гб), HDD- IТБ,

В качестве операционной системы на сервере не совсем целесообразно

использовать операционную систему семейства Unix, такую как, например, Linux из-за недостаточно хорошей поддержки ODBC.

Так как клиентские компьютеры будут подключаться к серверной базе данных, и сервер и клиентские машины должны функционировать в локальной сети. Это может быть, как одноранговая сеть, так и сеть с доменной организаци­ ей.

#### 11 ПРОЕКТНАЯ ЧАСТЬ

#### Разработка проекта автоматизации

#### Этапы жизненного цикла проекта автоматизации

Понятие жизненного цикла является базовым в проектировании информа­ ционных систем. Под жизненным циклом информационной системы понимают непрерывный процесс, начинающийся с момента принятия решения о создании информационной системы и заканчивающийся в момент исключения ее из использования.

Жизненный цикл информационной системы состоит из последовательных этапов, представленных на схеме ниже:

анализ требований

проектирование

кодирование (программирование)

тестирование и отладка

эксплуатация и сопровождение

разв1п11е

изъятие из экплуатации

Выбор модели жизненного цикла определяет течение самого жизненного цикла информационной системы.

В настоящий момент насчитывается несколько стандартов, которые ре­ гламентируют жизненный цикл программного обеспечения, а иногда и сами процессы разработки.

Ниже перечислены наиболее известные стандарты:

ГОСТ 34.601-90 является государственным стандартом, который распространяется на автоматизированные системы и устанавливает стадии и этапы их создания. Более того, данный стандарт включает описание содержания работ на каждом этапе. В большей степени, стадии и этапы работы, отраженные в стандарте, относятся к каскадной модели жизненного цикла;

ISO/IEC 12207:1995 является стандартом, регламентирующим про­ цессы и организацию жизненного цикла. Данный стандарт распро­ страняется на все виды заказного программного обеспечения. Стан­ дарт не регламентирует содержание фаз, стадий и этапов;

Custom Development Method (методика Oracle) по разработке при­ кладных информационных систем является технологическим мате­ риалом, который детализирует этапы работы до уровня заготовок проектных документов, которые рассчитаны на использование в проектах с применением Oracle. Он применяется для классической модели жизненного цикла, то есть в нем предусмотрено описание всех фаз и этапов. Кроме того, он используется для технологий

«быстрой разработки» (Fast Track) или «облегченного подхода», ко­ торые рекомендуются для малых проектов;

Rational Unified Process (RUP) описывает итеративную модель раз­ работки, которая включает 4 фазы: начало, исследование, построе­ ние и внедрение. Каждая из перечисленных фаз может быть разбита на этапы, или итерации, по итогам которых выпускается версия для внутреннего или внешнего использования. Последовательное про­ хождение всех этапов является циклом разработки, каждый цикл оканчивается формированием версии системы. В случае если после этого деятельность над проектом не заканчивает, то полученный продукт продолжает и дальше развиваться и снова проходит обо­ значенные выше фазы. Основной сутью работы в рамках RUP явля­ ется создание и сопровождение моделей на базе UML.

Microsoft Solution Framework (MSF) является сходной с RUP, так как включает тоже 4 фазы: анализ, проектирование, разработка, стабилизация. MSF является итерационной моделью, которая пред­ полагает использование объектно-ориентированного моделирова­ ния. MSF по сравнению с RUP ориентирована в большей степени на разработку приложений.

Extreme Programming (ХР) является самой новой среди рассматри­ ваемых методологий. Она сформировалось в 1996 году. В ее основе лежит командная работа: эффективная коммуникация между заказ­ чиком и исполнителем в течение всего проекта по разработке ин­ формационной системы, а разработка ведется с использованием по­ следовательно корректируемых прототипов.

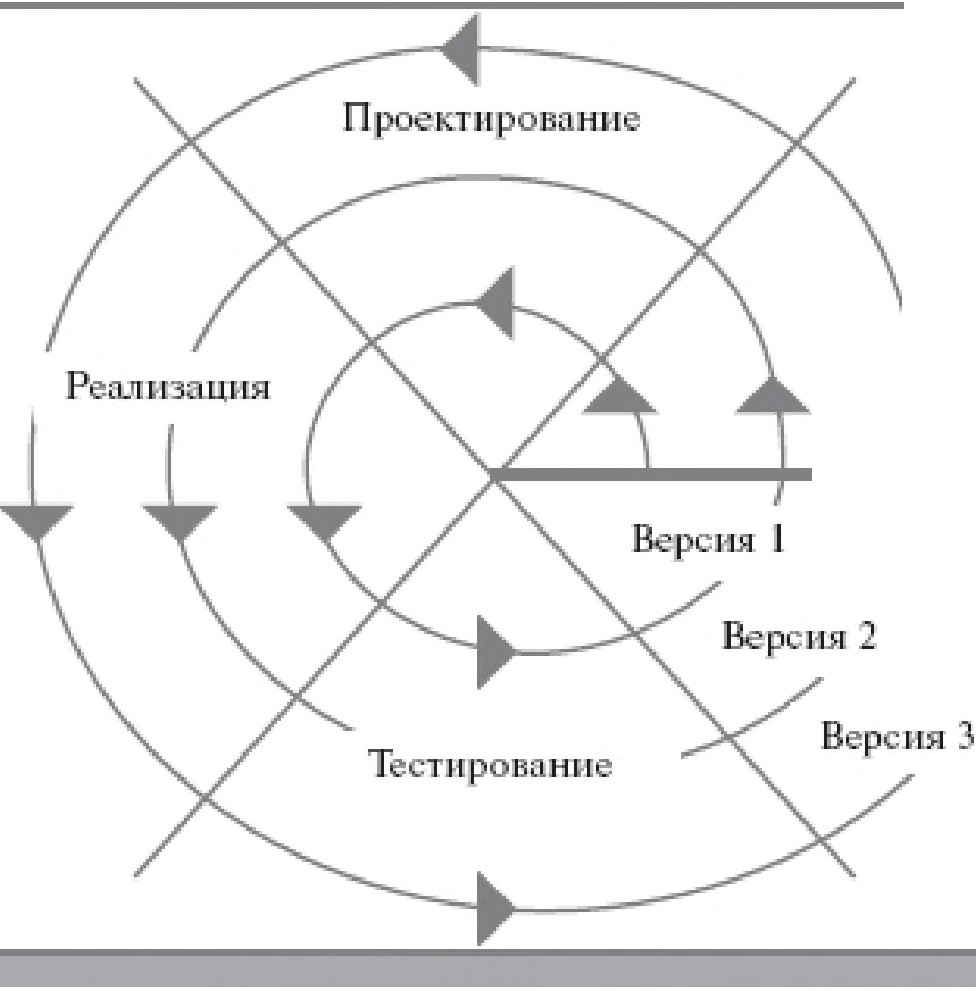
Стандарт ISO/IEC 15288:2002 включает 6 стадий жизненного цикла системы. Данные стадии представлены формированием концепции (анализ потребностей, выбор концепции и проектных решений); разработкой (проектирование); реализацией (изготовление); эксплу­ атацией (ввод в действие и использование); поддержку (обеспече­ ние функционирования); снятие с эксплуатации (прекращение ис­ пользования, демонтаж, архивирование). В практических целях очень важно детализировать стандарт до уровня целей, результатов и конкретных действий. Кроме того, помимо процессов стандарт определяет двести восемь действий и сто двадцать три различных результата этих действий.

Стандарт ISO/IEC 12207 не предлагает какой-либо конкретной модели жизненного цикла и методов разработки, он лишь предоставляет общие реко­ мендации для любых моделей жизненного цикла. Модель является структурой, которая определяет порядок выполнения и взаимосвязи процессов, действий и задач в период жизненного цикла.

Существует две основные модели жизненного цикла, которые применяют­ ся на практике: каскадная и спиральная модели. В каскадной модели (поэтапная модель с обратной связью) процесс разработки проходит последовательно от од-

ной фазы к другой, причем переход к следующей фазе происходит лишь после завершения предыдущей.

В спиральной модели, представленной на рисунке **1**Оа, видно, что на каж­ дом витке спирали выполняется создание очередной версии продукта, уточня­ ются требования проекта, определяются его качество и планируются работы следующего витка. Таким образом, продукт постоянно и постепенно улучшается до тех пор, пока не получен желаемый результат. Это является особенно важ­ ным, когда заказчик сам точно не понимает все возможности ИС и, соответ­ ственно, не полностью осознает, что хочет получить в итоге. В этой связи важно уделить особое внимание начальным этапам разработки - анализу и проектиро­ ванию, когда реализуемость тех или иных технических решений проверяется и обосновывается посредством макетирования, или создания прототипов.



**Разработка**

**тр бош1кий**

**ВЕ,од** з дейсгшие

**·11pororri1пoв с.иС"fемм**

/

Рисунок 10 (а). Спиральная модель

В разрабатываемом проекте для жизненного цикла была выбрана поэтап­ ная модель, потому что она больше подходит для разрабатываемой системы. Как свойственно выбранной модели, переход к каждому следующему этапу проис­ ходит только после завершения всех работ на предыдущем этапе (рисунок **11**), в том числе подготовка полного пакета документации, достаточная для того, что­ бы разработка могла быть продолжена другой группой разработчиков. В число

преимуществ данной модели можно включить возможность планирования сро­ ков завершения работ и затрат на их выполнение.

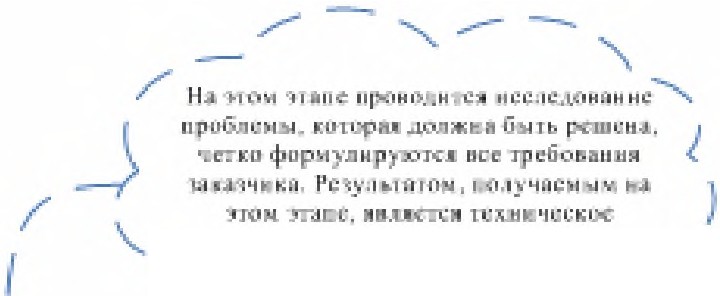
**Анали.з**

..... ... --..,.....

Проектирование

,---

/" **Pi11po6**□**тi,;.1 1Dp0ifк711 :i;.1J]eШ!f'u1e1i1,** "



\

**:н!, ::шеш . ror.,к<1u::11111.:-f' ('(1 ar::i!!1o:111** /

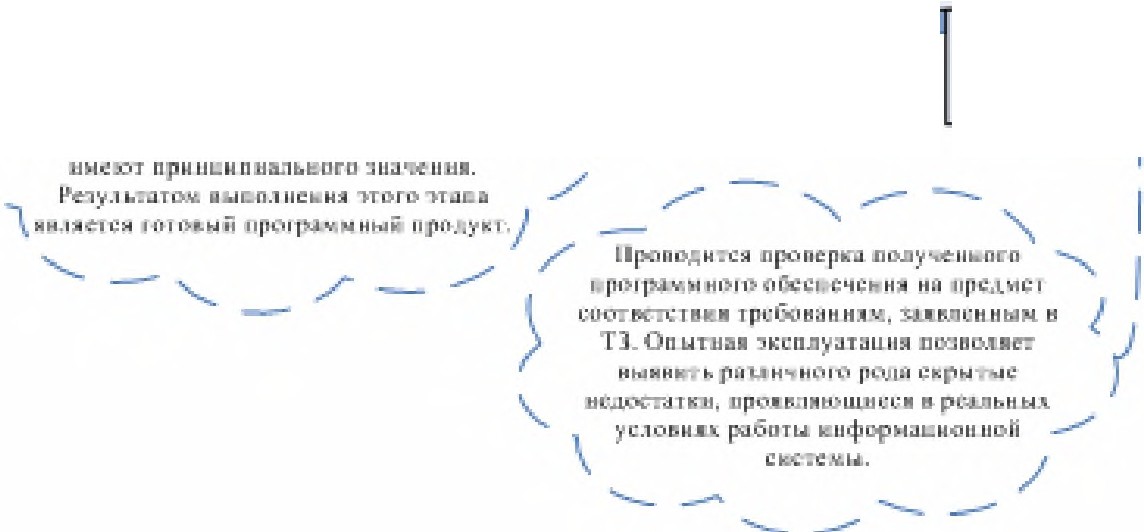
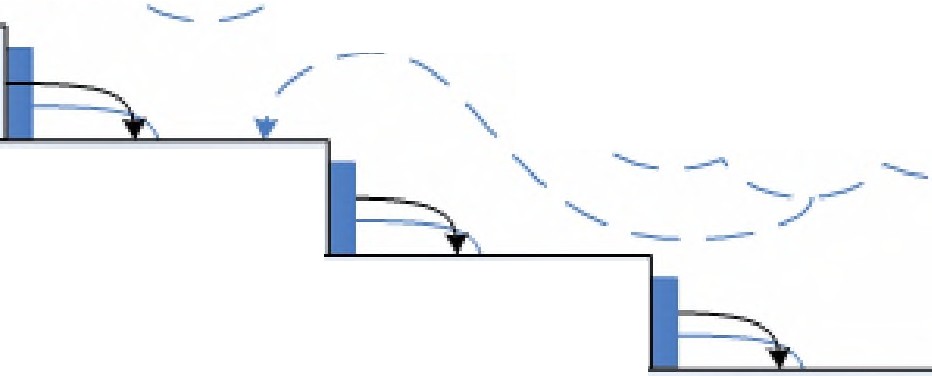
**llt< 1'Cjl(:: 1нf1i.11"'-1У!Н :'lj11"!<нAi;111,** /

'-

**R:Hl.1if'ПilOJ1Яl1)IIIШI r.:,,e,;1,11; T(1i:: lill::IWll:H,HI,**

*>*

**•• 1t•:!i·.1,!(r•1o ,1...... "'1 "т·-1,\_ Pe1:t h 1.,."1** *(*



') ОГО **")";,1.Пia] .ur.:-c IUIN.П.1ifП прп IП"** }

**,r;n \_11 11:!!illTl:lllliik.., l:!П Dlll,e-iJ е**

' **tt(:1'kl;,.1'!oJ1Jillilkf(:...U...н""1,1JtJ1.,1**■ **ji'=•·• l,:'i1,н1i**

\ **IJ l.il'1111.**

---

./ --""------ "-

**еалиэация**

**c;J:,1111<:i lщ ю111fi "'g:.:. r-JJ,** "(, **!1f;'lщr;: нш1С'1 1 ............**

/ **zpiJljQ"ПI IIIJШlfPiJR..., uru r;:,{i1,;jl;(Jc-ti.1111UM 11-**

*)*

**,:оотu 11 r::** □**pD•f'E:'ILIL.11o111 рtш 11ж1,1 ,**

*)*

**IПП.\\_8!1Ч ::шI W Шс IIJ. npe:.11I.&.д)'llli:,\li 1'':' i::.**

)-

**М1,аu,дщ ltL=lltl.a1b:!:S("lttfl.ll.JММ pC-1.111".ilU.IIHII, L**

\

*J* Внедрение

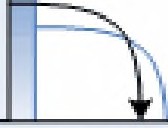
**Сопровождвни** е

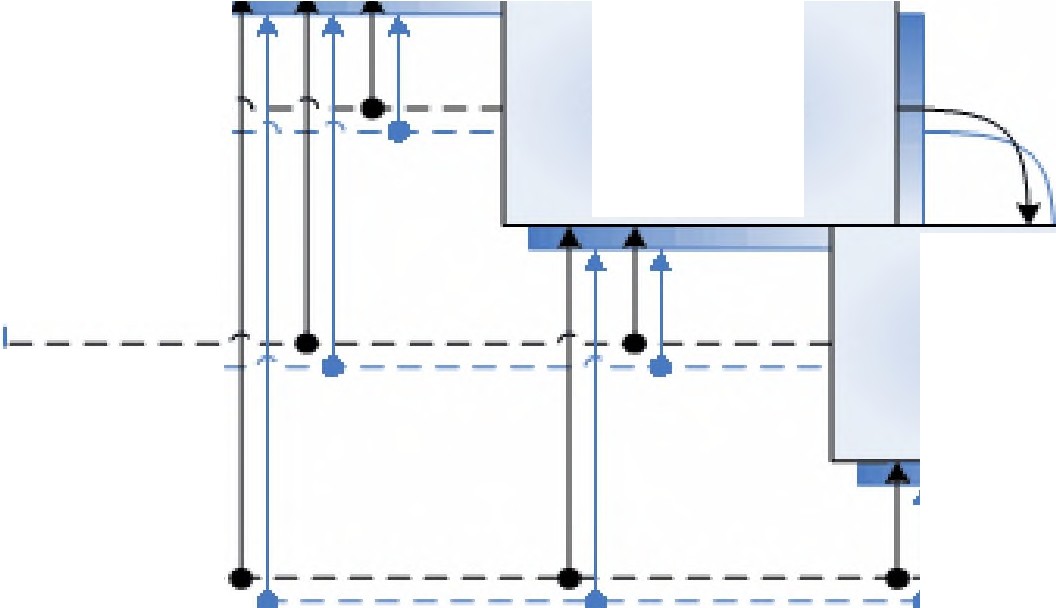
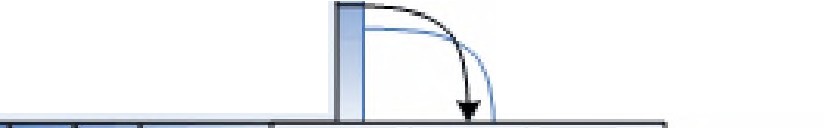
Рисунок 10 (6) Каскадная схема разработки ПО.

Что касается каскадной модели разработки, то можно отметить, что она хорошо подходит для построения систем, в которых в самом начале разработки можно достаточно точно и полно сформулировать все требования, чтобы предо­ ставить разработчикам свободу реализовывать их как можно лучше с техниче­ ской точки зрения. Тем не менее, в случае, если в середине разработки вскрыва­ ются ошибки, которые были допущены в начале разработки проекта, то необхо­ димо будет прибегнуть к интроверсии проекта и реальная схема каскадной мо­ дели приобретет иной вид (поэтапная модель).

Анализ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| - | - | - | - | - |  | **С(шр-оеождение** |

Рисунок 11. Поэтапная модель



**1**

I 1 **1** 1

1 1

**1** 1

'**1** 11

**1** 1 '-1;

ПрQе с;тt1рQв<1н111е

**1** 1

'**1**+1 \_:-\_:-\_:-\_:-

**Реализ.аци11**

**Внедрение**

]\_ \_

**L--------**

## ние

* + 1. **Ожидаемые риски на этапах жизненного цикла и их описа­**

Необходимо понимать, что любой проект, нацеленный на создание ин­

формационной системы организации, включает в себя множество задач, связан­ ных с проектированием информационной системы, разработкой программного обеспечения, внедрением и общим управлением проектом. Каждая из представ­ ленных выше задач - сама по себе мини-проект, который обладает своими осо­ бенностями. В этой связи в процессе разработки появляются различные риски.

Выделяют два вида рисков: риски заказчика и исполнителя. Рассмотрим примеры данных рисков на рисунке ниже.

### Риски заказчика

* + - * неполное достижение цели
      * не эффективное расходование средств

### Риски исполнителя

* повышение фактической себестоимости
* программно-технические риски



Более того, в связи с тем, что нужно вести параллельно крайне различные по своему характеру проекты, уровень риска таких проектов во много раз воз­ растает.

В таблице 4 отображены наиболее характерные риски и методы из мини­ мизации.

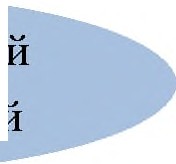
**Таблица 4 Вероятные 1>иски проекта и способы их минимизации**

|  |  |  |
| --- | --- | --- |
| **Виды рисков** | **Пути уменьшения рис­**  **ка** | **Пути снижения вероятно­**  **сти возникновения риска** |
| Риски, связанные с мас­ штабом проекта | Детально проанализиро­ вать каждый этап работ, наладить взаимодействие участников, организации работ | Детально проработать про­ грамму качества, отработать управление конфигурацией проекта, провести специ­ альные процедуры взаимо­ действия участников |
| Риски, связанные с недо­ статочным опытом в сфере информационных техноло­  гий | Провести обучение поль­ зователей, включая руко­ водство, соблюдать тех­  нологий работы | Разработать и утвердить концепцию проекта на воз­ можно более ранней его  стадии |
| Технические риски проек­ та | Провести строгий отбор проектной команды по квалификационным кри­ териям. Обучить участ­ ников проекта техноло­ гии проектных работ, ин­  струментальным сред­ ствам | Использовать стандарты предприятия на проектные работы, разработать стан­ дарты проекта |
| Организационные риски проекта | Обучить участников про­  екта, тренинги, команды, предоставить как можно более подробную форма­  лизацию деятельности | Включить в команду адми­ нистратора проекта, прове­ сти детальное распределе­ ние ролей в проекте |
| Операционные риски про­ екта | Многократно протести­ ровать созданные про­ дукты, провести тща­ тельную экспертизу до­ кументов | Строго выполнять процеду­ ры программы качества |

# Организационно-правовые и программно-аппаратные средства обеспечения информационной безопасности и защиты ин­ формации

У каждой информационной системы существует потенциальные угрозы безопасности, как например, воровство данных или срыв работы системы. В этой связи разработчикам информационной системы следует тщательно проду­ мать её безопасность.

Существует четыре уровня защиты, которые предполагает комплексная защита информации в информационных системах. Они представлены на схеме ниже.



правовой

административный

/организационный

Рассмотрим подробнее каждый из них. Правовой уровень представлен юридическими нормами и законами. Данный уровень необходим для предот­ вращения незаконного использования секретных и конфиденциальных данных или информации, которая является объектом авторского права.

Административный уровень обеспечения информационной безопасности информационной системы необхдим для определения должностных инструкций. На этом уровне определяется, кто и какую информацию может собирать и хра­ нить. Административный уровень определяет способы доступа к информации и условия ее распространения, права и обязанности работников, их компетенция и ответственность, должностные инструкции.

На аппаратно-программном уровне представлено применение процедуры идентификации пользователя, открывающей доступ к данным и программным средствам. Аппаратную защиту можно выполнить посредством кодовой карточ­ ки, ключа и т.п. Самым слабым местом в системе защиты информации считается персонал, так как люди обычно не хотят использовать любые дополнительные средства защиты, которые могут создать им неудобства в процессе трудовой де­ ятельности. В этой связи для применения аппаратных средств защиты необхо­ дима поддержка руководства.

Под криптографической защитой понимают шифрование данных для скрытия от злоумышленника их смысла. Смысл данных будет недоступен поль­ зователю, пока он не будет идентифицирован по личному ключу.

Разработка средств информационной безопасности в информационной си­ стеме осуществляется на следующих принципах: системный подход, разделение полномочий, контроль функционирования системы защиты, регистрация попы­ токвторжения, непрерывное развитие системы, минимизация полномочий, обес­ печение надежности системы защиты, разнообразие средств борьбы с вредонос­ ными программами, обеспечение экономической целесообразности.

В результате решения проблем безопасности информации разрабатывае­ мая информационная система должна обладать следующими основными призна­ ками:

наличие механизма предотвращения несанкционированного доступа;

дифференциация информации различной степени конфиденциальности;

целостность программного обеспечения и информации.

разграничение полномочий субъектов доступа к компонентам информационной системы;

связи на дальние расстояния;

управление потоками информации, как в локальных сетях, так и при передаче по каналам

обеспечение криптографической защиты информации при передаче данных;

наличие средств восстановления системы защиты информации.

В основу нормативно-правового обеспечения информационной безопас­ ности входят нормы и регламенты деятельности организации, служб и средств, реализующих функции защиты информации, разнообразные методики, органи­ зующие деятельность пользователей при выполнении своихфункций в условиях жестких требований соблюдения конфиденциальности.

## Информационное обеспечение задачи

## Информационная модель и её описание

Информационная модель представляет собой схему движения входных, промежуточных и результативных потоков и функций предметной области. Кроме того, она объясняет, на основе каких входных документов и какой нормативно-справочной информации происходит выполнение функций по обработке данных и формирование конкретных выходных документов.

На этапе концептуального проектирования можно выделить следующие типы, относящиеся к данной предметной области: «Реквизиты фирмы», «Спра­ вочная информация о сотрудниках», «Справочная информация по должностям»,

«Справочная информация об уровне МРП», «Справочная информация об уровне МРОТ» и «Начисления заработной платы».

Атрибуты типа «Реквизиты фирмы»:

* Наименование фирмы;
* Адрес;
* Код налогового органа. Атрибуты типа «Сотрудники»:
* ФИО;
* Должность;
* Дата приема;
* Дата увольнения;
* Сотрудник уволен;
* Оклад;
* Тариф;
* Количество иждивенцев;
* Название НПФ;
* Адрес;
* Серия и номер паспорта;

Кем выдан паспорт, дата выдачи;

* Льготы.

Атрибуты типа «Должность»:

* Название должности.

Атрибуты типа «Справочная информация об уровне МРП»:

* Дата изменения;
* Уровень МРП.

Атрибуты типа «Справочная информация об уровне МРОТ»:

* Дата изменения;
* Уровень МРОТ.

Атрибуты типа «Начисления заработной платы»:

* ФИО;
* Начислено доходов;
* Удержано;
* Начислено за отчетный месяц;
* Начислено с начала года;
* Налоговые вычеты за отчетный месяц;
* Налоговые вычеты с начала года;
* Доход, не подлежащий налогообложению за отчетный месяц;
* Доход, не подлежащий налогообложению с начала года;
* Сумма облагаемого дохода;
* Подоходный налог по ставкам;
* Сумма подоходного налога за истекший период;
* Сумма подоходного налога в предыдущем периоде;
* Сумма подоходного налога подлежащего удержанию в текущем ме- сяце;
* Доход, облагаемый СН за текущий месяц;
* Доход, облагаемый СН с начала года;
* Доход, не подлежащий обложению СН (кроме ОПВ) за текущий ме- сяц;
* Доход, не подлежащий обложению СН (кроме ОПВ) с начала года;
* Расчетная сумма дохода облагаемого СН;
* Сумма СН с расчетного годового дохода по таблице;
* Сумма СН за истекший период;
* Доход, с которого исчисляются социальные отчисления;
* Сумма социальных отчислений;
* Сумма социального налога, подлежащая уплате в бюджет.

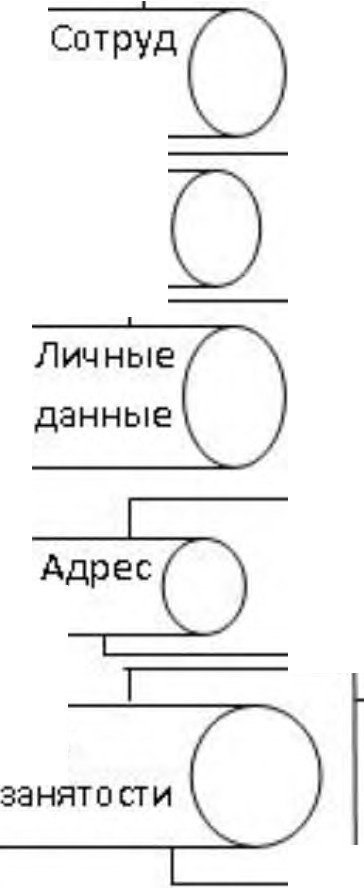
Таким образом, получена концептуальная модель проектируемой пред­ метной области. На этапе логического проектирования разработанная модель преобразуется в более «пригодную» структуру для реализации ее в среде суще­ ствующих СУБД, причем с ориентацией на выбранную СУБД. Разрабатываемое приложение представляет собой локальную БД, т.е. хранимые данные находятся на том же компьютере, что и приложение для работы с БД. В таком случае гово­ рят, что база данных имеет локальную архитектуру.

В программе должны использоваться следующие данные:

Входные данные: информация о реквизитах фирмы; информация о долж­ ностях и подразделениях; уровне МРП и МРОТ; информация, о сотрудниках включая ФИО, дату приема, дату увольнения, размер оклада, количество человек находящихся на иждивении, наличие льгот.

Выходные данные: расчет начисленных и удержанных сумм, расчет сумм подлежащих выплате, суммы начисленного налога на доходы физических лиц, суммы социального налога.

ис Бухгаптер



ники

Ведение

справочник ов

Должн

ости

Расчет

заработной платы

Вид

начисления

Формирова

ние отчетов

Вид

Начисление

зарплаты

Паспортные

данные

Расчетная

веда масть

Отчетность

Бухгаптер

Рисунок 12. Информационная модель

## Характеристика нормативно-справочной, входной и опера­ тивной информации

Информация, используемая системой, делится на две части: условно­ постоянная и текущая.

Условно-постоянная информация изменяется сравнительно редко. Такая информация хранится в справочных таблицах.

Перечень справочных таблиц представлен в таблице 5.

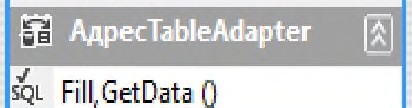
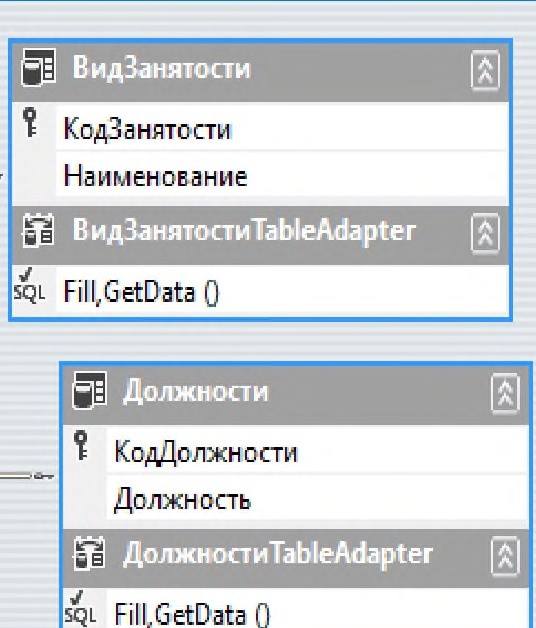
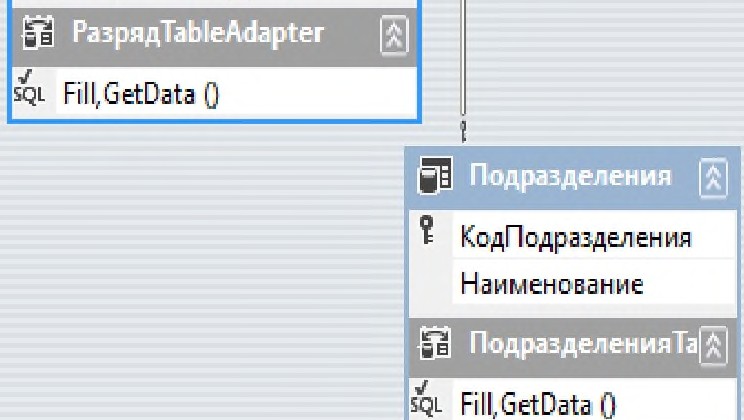
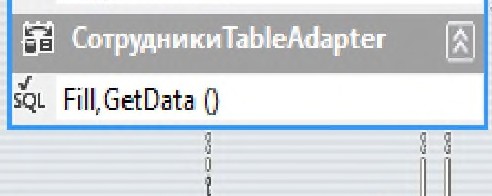
**Таблица 1**

**Перечень справочных таблиц**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название спра- вочной таблицы** | **Ответственный за ведение** | **Средний объем спра- вочника в**  **записях** | **Средняя ча- стота актуа- лизации** | **Средний объ- ем актуали- зации в%** |
| Подразделение | Администратор | 10 | редко | 10 |
| Должность | Администратор | 50 | редко | 10 |
| Разряд | Администратор | 18 | редко | 100 |
| Льгота | Администратор | 10 | 1 раз в год | 30 |
| Вид начисления | Администратор | 30 | редко | 10 |
| Вид удержания | Администратор | 20 | редко | 10 |
| Вид занятости | Администратор | 10 | редко | 10 |
| Статус налогопла-  тельщика | Администратор | 10 | 1 раз в год | 10 |
| Сотрудники | Пользователь | 500 | 1 раз в месяц | 20 |
| Личные сведения | Пользователь | 500 | 1 раз в месяц | 10 |
| Паспортные дан-  ные | Пользователь | 500 | 1 раз в месяц | 10 |
| Адрес | Пользователь | 500 | 1 раз в месяц | 10 |

Реквизитный состав справочных таблиц представлен на рисунке ниже.

l:ji] **Сотрудники** []



КодАдр,еса Индекс Регион Район

НаселенныйПункт Улица

Дс>м Квартира

КодСотrудника ЛичныеДанные Дс>лжность Подразделение Да аПр,иема

Да аУвольнения Ра3ряд ВидЗанятости

l:ji] **Адрес** @

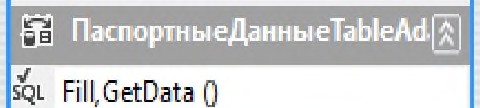
l:ji] **Ра3ряд** @

|  |  |  |  |
| --- | --- | --- | --- |
| l:ji] **ЛичныеДанные** |  | [] |  |
| ! Код Фамилия Имя Отчктво  ДатаРаждения МестоРождения  инн  снилс  Паспоf"Г Адрес Гражданство  . **-J.**  s L Filll,GetData 0 | ., -· ., | |  |

Код Разряд

ТаифныйКоэффициент

l:ji] **ПасnортныеДанны е** []



КодПа спорта Серия

Нмер КемВыдан ДатаВ.ыдачи

КодПодразделения ДатаСведений СрокДействия

Приведем характеристику таблиц базы данных

Таблица 6

Описание таблицы «Подразделения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Идентификатор поля | Тип поля | Длина поля | Прочее |
| Код подразделения | КодПодразделе-  **ния** | Int |  | Ключевое поле |
| Наименование подраз- деления | Наименование | Nvarchar | 50 |  |

Таблица 7 Описание таблицы «Разряд»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Идентификатор  поля | Тип поля | Длина поля | Прочее |
| Код разряда | Код | Int |  | Ключевое поле |
| Разряд | Разряд | Int |  |  |
| Тарифный коэффици-  ент | ТарифныйКоэф-  фициент | float |  |  |

Таблица 8

Описание таблицы «Должности»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Идентификатор поля | Тип поля | Длина поля | Прочее |
| Код должности | КодДолжности | Int |  | Ключевое поле |
| Название должности | Должность | Nchar | 50 |  |

Таблица 9

Описание таблицы «Вид занятости»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Идентификатор  поля | Тип поля | Длина поля | Прочее |
| Код занятости | КодЗанятости | Int |  | Ключевое поле |
| Наименование | Наименование | Nchar | 50 |  |

Таблица 10

Описание таблицы «личные данные»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Идентификатор  поля | Тип поля | Длина поля | Прочее |
| Код | Код | Int |  | Ключевое поле |
| Фамилия | Фамилия | Nchar | 30 |  |
| Имя | Имя | Nchar | 30 |  |
| Отчество | Отчество | Nchar | 30 |  |
| Дата рождения | ДатаРождения | Date |  |  |
| Место рождения | МестоРождения | Nchar | 50 |  |
| инн | инн | Int |  |  |
| снилс | снилс | Nchar | 15 |  |
| Паспорт | Паспорт | Int |  |  |
| Адрес | Адрес | Int |  |  |
| Гражданство | Гражданство | Nchar | 10 |  |

Таблица 112 Описание таблицы «Паспортные данные»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Идентификатор по-  ля | Тип поля | Длина поля | Прочее |
| Код паспорта | КодПаспорта | Int |  | Ключевое поле |
| Серия | Серия | Int |  |  |
| Номер | Номер | Int |  |  |
| Кем выдан | КемВыдан | Nchar | 50 |  |
| Дата выдачи | ДатаВыдачи | Date |  |  |
| Код подразделения | КодПодразделения | Nchar | 10 |  |
| Дата сведений | ДатаСведений | Date |  |  |
| Срок действия | СрокДействия | Date |  |  |

Таблица 12

Описание таблицы «Адрес»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Идентификатор поля | Тип поля | Длина поля | Прочее |
| Код адреса | КодАдреса | Int |  | Ключевое поле |
| Индекс | Индекс | Nchar | 10 |  |
| Регион | Регион | Nchar | 50 |  |
| Район | Район | Nchar | 50 |  |
| Населенный пункт | НаселенныйПункт | Nchar | 50 |  |
| Улица | Улица | Nchar | 50 |  |
| Дом | Дом | Nchar | 10 |  |
| Квартира | Квартира | Nchar | 10 |  |

Таблица 13

Описание таблицы «Сотрудники»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Идентификатор  поля | Тип поля | Длина поля | Прочее |
| Код сотрудника | КодСотрудника | Int |  | Ключевое поле |
| Личные данные | ЛичныеДанные | Int |  |  |
| Должность | Должность | Int |  |  |
| Подразделение | Подразделение | Int |  |  |
| Дата приема | ДатаПриема | Date |  |  |
| Дата увольнения | ДатаУвольнения | Date |  |  |
| Разряд | Разряд | Int |  |  |
| Вид занятости | ВидЗанятости | Int |  |  |

Таблица 143

Описание таблицы «Вид начисления»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Идентификатор  поля | Тип поля | Длина поля | Прочее |
| Код вида начисления | КодВидаНачисле- ния | Int |  | Ключевое поле |
| Наименование | Наименование | Nchar | 50 |  |

Описание таблицы «Вид удержания»

## Характеристика результатной информации

Описание результатных документов

Таблица 15

Таблица 16

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование поля | Идентификатор поля | Тип поля | Длина поля | Прочее |
| Код вида удержания | КодВидаУдержания | Int |  | Ключевое поле |
| Вид удержания | ВидУдержания | Nchar | 50 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Наименование** | | **Реквизиты** | **Таблицы, на основе кото- рых формиру-**  **ется документ** | **Частота фор- мирования** | **Способ ставки** | **до-** |
| Начисление платы | зар- | Месяц начисления Подразделение **ФИО** сотрудника Вид начисления  Сумма | Сотрудники Вид начисления Начисление | 1 раз в месяц | Экранная ма | фор- |
| Удержания зарплаты | из | Месяц удержания Сотрудник  Вид удержания  Сумма | Сотрудники Вид удержания Удержания | 1 раз в месяц | Экранная ма | фор- |
| Расчетная масть | веда- | Месяц Подразделение Сотрудник  Вид начисления Сумма  Итого начислений Вид удержаний Сумма  Итого удержаний  *К* выдаче на руки | Сотрудники Вид начисления Начисление Вид удержания Удержания | 1 раз в месяц | Печатная ма | фор- |
| Платежная масть | веда- | Месяц Подразделение Сотрудник  *К* выдаче на руки | Сотрудники Вид начисления Начисление  Вид удержания Удержания | 1 раз в месяц | Печатная ма | фор- |

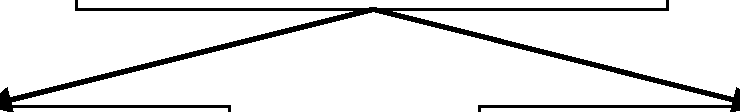
## Программное обеспечение задачи

## Общие положения (дерево функций и сценарий диалога)

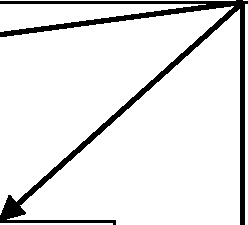
### В разработанном модуле предусмотрен только один пользователь бухгалтер. Дерево функций модуля представлено на рисунке 13.

6C7-

Функции бухгалтера



Основные Вспомогательные



Расчет начислений

Расчет удержаний

Расчет социального налога

Заполнение справоч­ ной информации

Рисунок 13. Дер;,ево функций мBодуля9 0

С(9ценарий диалога, котор>ый предста?влен1на рисунке 14. A,

Авторизация Вход в главное меню Выход

1

Главное меню Справочники Документы и расчеты Отчеты

Справка Выход

1

1 1 1 1

Справка

О программе

Отчеты

Начисление за месяц Начисление по подразделению Удержания

Документы и рас­ четы

Начисление по окладам Начисление повременной оплаты

Начисление премии Начисление за ночные Начисление по больнич­ ному

Начисление отпускных Начисление сдельной оплаты

Удержания

Начисление социального налога

Справочники

Адрес

Вид занятости Должности Личные данные Льготы

Паспортные данные Подразделения Разряд

Реквизиты организации Сотрудники

Статус налогоплатель­ щика

Виды начислений Виды удержаний

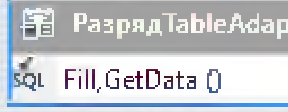
Рисунок 14. СцAен, ар(9ий диалога

# Характеристика базы данных

в разработанной системе автоматизации используется СУБД MS SQL Server 2012. Обоснование ее выбора приведено в разделе

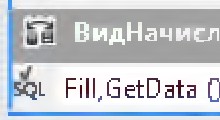
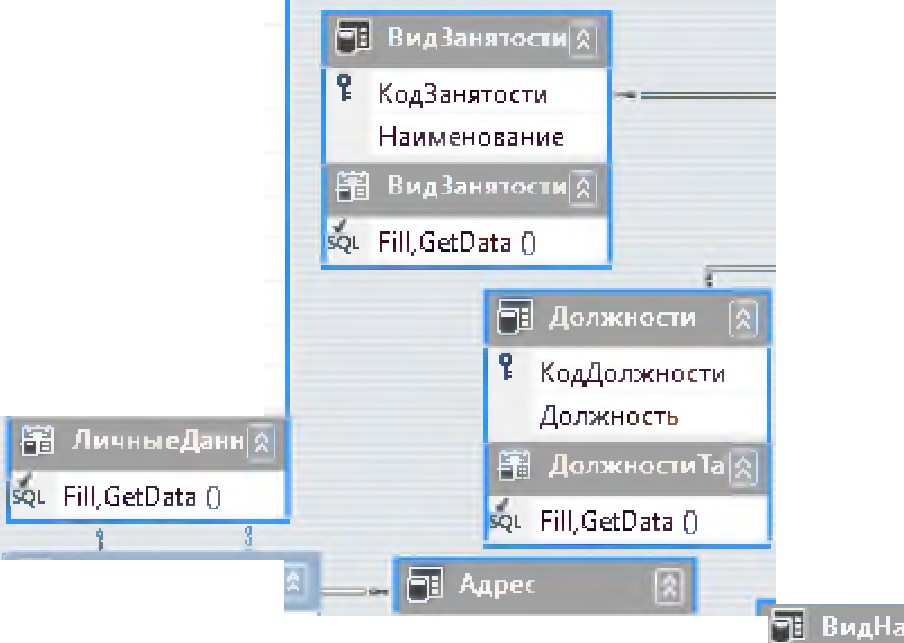
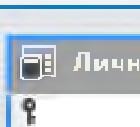
1.4.3. В решаемой задаче используется следующая ЕR-модель, отображенная на рисунке 15, описывающая взаимосвязь таблиц в БД.

i}



**'i!** Код Ра,ряд

ТаифныйКоэффициент



. -].

Код Фамилия Имя Отчество

Дата Рождения

Место Рождения

инн снилс

Паспорт

Адрес Гражданство

**Сшрудники**

'! КодСотрудника

• =========== -'! ЛичныеДанные

Должность Подраз.деление ДатаПриема ДатаУвольнения Разряд

ВидЗанятости

\!)"

'! КодПодраз.деления Наименование

**1 '** Fill,GetData О

!

* -==--------'1

i:)11 **РаарDд**

**l.;i,** Fill,GetData О

i'ё **lim =2111**

'! КодПаспорта Серия

Нмер

Кем Выдан ДатаВыдаци КодПодразделения Дата Сведений

СрокДействия

***'i*** КодАдреса Индекс Регион Район

НаселенныйПункт Улица

Дом

Квартира

----

'! КодВидаНачисления

Наименование

***Ji,*** Fill,GetData О

|  |
| --- |
| - **.::iai"."..**  **'i!** Код  Код( отрудни ка КодВидаНачисления ДатаНачисления  Сумма |
|  |
| **1 ,** Fill,GetData 0 |

Рисунок 15. ЕR-диаграмма базы данных

# Структурная схема пакета (дерево вызова программных модулей)

Как правило, разработка программного обеспечения выполняется в два

этапа:

1. проектирование логики программы;
2. разработка и отладка кода программы.

В число задач проектирования логики информационной системы включа­ ют определение состава программных модулей, выделение пакетов модулей и определение связей между ними.

В значительной степени зависит от используемых инструментальных средств разработки разработка кода программ.

По итогам анализа функций, которые должны выполняться в системе ав­ томатизации процесса расчета заработной платы и исчисления налогов, был установлен состав программных модулей и определены связи между ними.

Кроме того, были определены управляющие программные модули, кото­ рые призваны выводить на экран кадры меню и передавать управление другим модулям в зависимости от того, какой пункт меню выбирает пользователь.

К управляющим модулям можно отнести модуль загрузки основного ме­ ню, формирования отчетов, настройки справочников.

Модули, выполняющие функции ввода конкретных видов начислений, просмотра и корректировки данных расчетов, печати документов, обработки и выдачи на просмотр и печать отчетов, загрузки, актуализации и выдачи на пе­ чать справок составляют второй класс.

Далее, на основе результатов, которые были получены в предыдущем пункте, составляется дерево программных модулей.

Описание программных модулей представлено в таблице

**описание программных модулеи**

|  |  |
| --- | --- |
| **Наименование модуля** | **Функции модуля** |
| MI. Модуль авторизации | Позволяет пользователю авторизоваться в системе. Загружает модуль главного меню |
| М2. Модуль регистрации | Обеспечивает регистрацию нового пользователя |
| МЗ. Модуль главного меню | Обеспечивает вызовы других модулей программы |
| М4. Модуль справочника «Вид занятости» | Обеспечивает редактирование таблицы «Вид заня- тости» |
| М5. Модуль справочника «Долж- ности» | Обеспечивает редактирование таблицы «Должно- сти» |
| Мб. Модуль справочника «Под-  разделения» | Обеспечивает редактирование таблицы «Подраз-  деления» |
| М7. Модуль справочника «Раз- ряд» | Обеспечивает редактирование таблицы «Разряд» |
| М8. Модуль справочника «Вид  начислений» | Обеспечивает редактирование таблицы «Вид  начислений» |

**Таблица 4**

**Продолжение таблицы 17**

|  |  |
| --- | --- |
| **Наименование модуля** | **Функции модуля** |
| М9. Модуль справочника «Вид  удержаний» | Обеспечивает редактирование таблицы «Долж-  **НОСТЮ)** |
| MIO. Модуль справочника «Долж-  **НОСТИ))** | Обеспечивает редактирование таблицы «Вид  удержаний» |
| **MII.** Модуль справочника «Со-  трудники» | Обеспечивает редактирование таблицы «Сотруд-  **НИКЮ)** |
| М12. Модуль справочника «Лич-  ные данные» | Обеспечивает редактирование таблицы «Личные  данные» |
| **МIЗ.** Модуль справочника «Пас-  портные данные» | Обеспечивает редактирование таблицы «Пас-  портные данные» |
| М14. Модуль справочника «Адрес» | Обеспечивает редактирование таблицы «Адрес» |
| М15. Модуль справочника «Рекви-  зиты организации» | Обеспечивает редактирование таблицы «Рекви-  зиты организации» |
| Мlб. Модуль расчета «Начисления  по окладам» | Обеспечивает расчет оплаты труда сотрудников,  получающих оклад |
| MI 7. Модуль расчета «Начисление  повременной оплаты труда» | Обеспечивает расчет оплаты труда сотрудников,  получающих простую повременную оплату труда |
| М18. Модуль расчета «Начисление  премии» | Обеспечивает расчет премий сотрудникам |
| М19. Модуль расчета «Начисление  за ночные» | Обеспечивает расчет доплаты за работу в ночное  время |

дам».

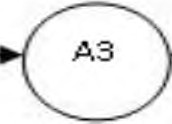
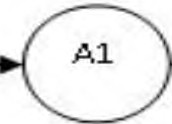
## Описание программных модулей

Рассмотрим структуру модуля Мlб Модуль расчета «Начисления по окла-

Начало

Установка соединения с БД

Расчет оклада по отработанному времени

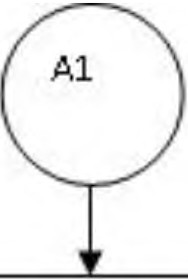


Сохранение рассчитанного оклада

Выход

Рисунок 16. Блок-схема модуля расчета оплаты труда по окладам

Работа блока расчета оплаты по окладу.



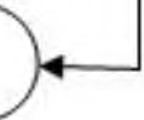
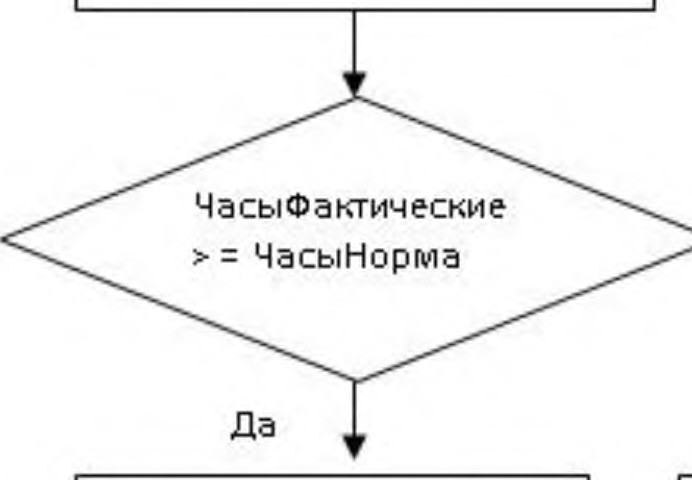
Загруз ка таблицы

«Сотрудники»

Определение раз ряда сотрудника

Загруз ка таблицы ((Раз ряд»

Определение тарифного коэФФици ента сотрудника



Нет

Оклад= Тарифная ставка

\*

ТарифныйКоэффици ент

Конец

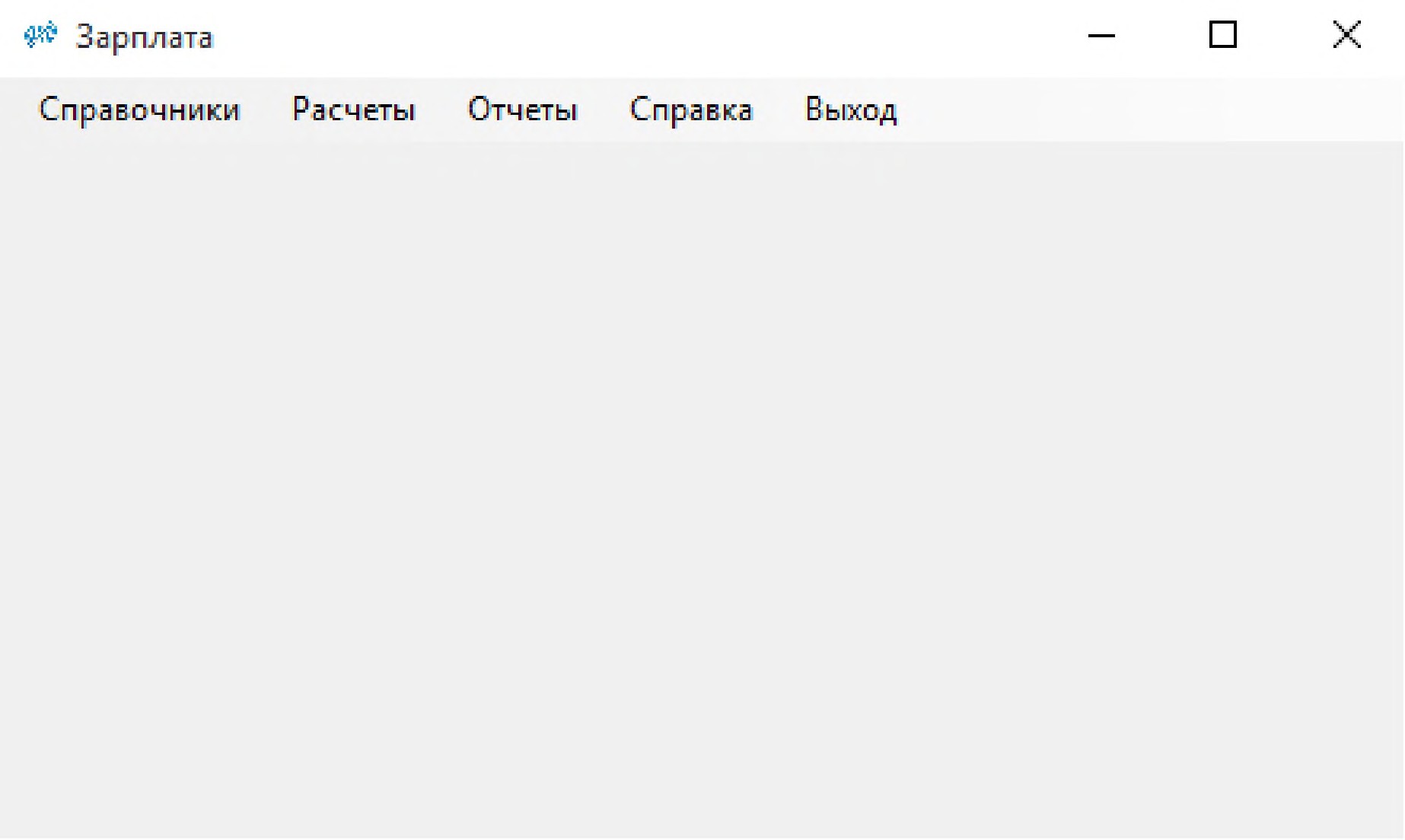
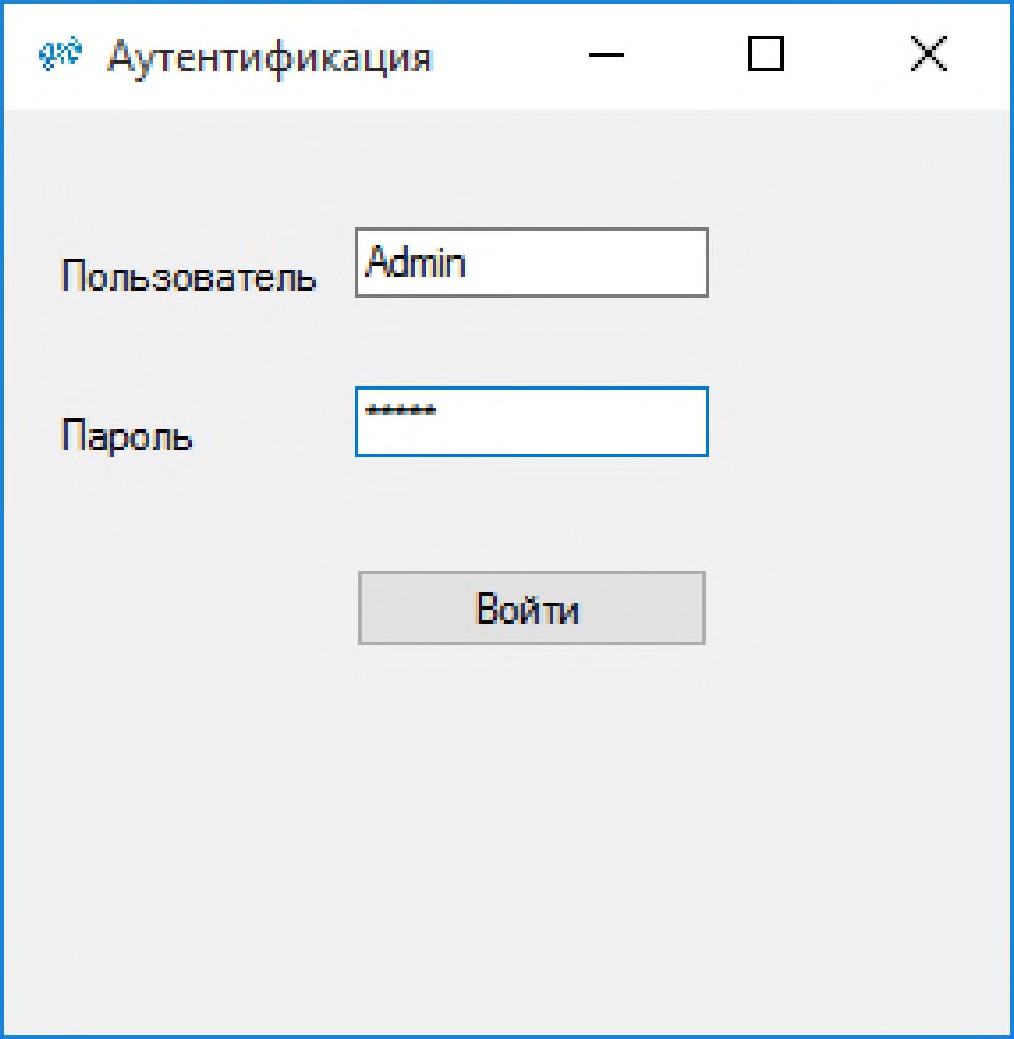
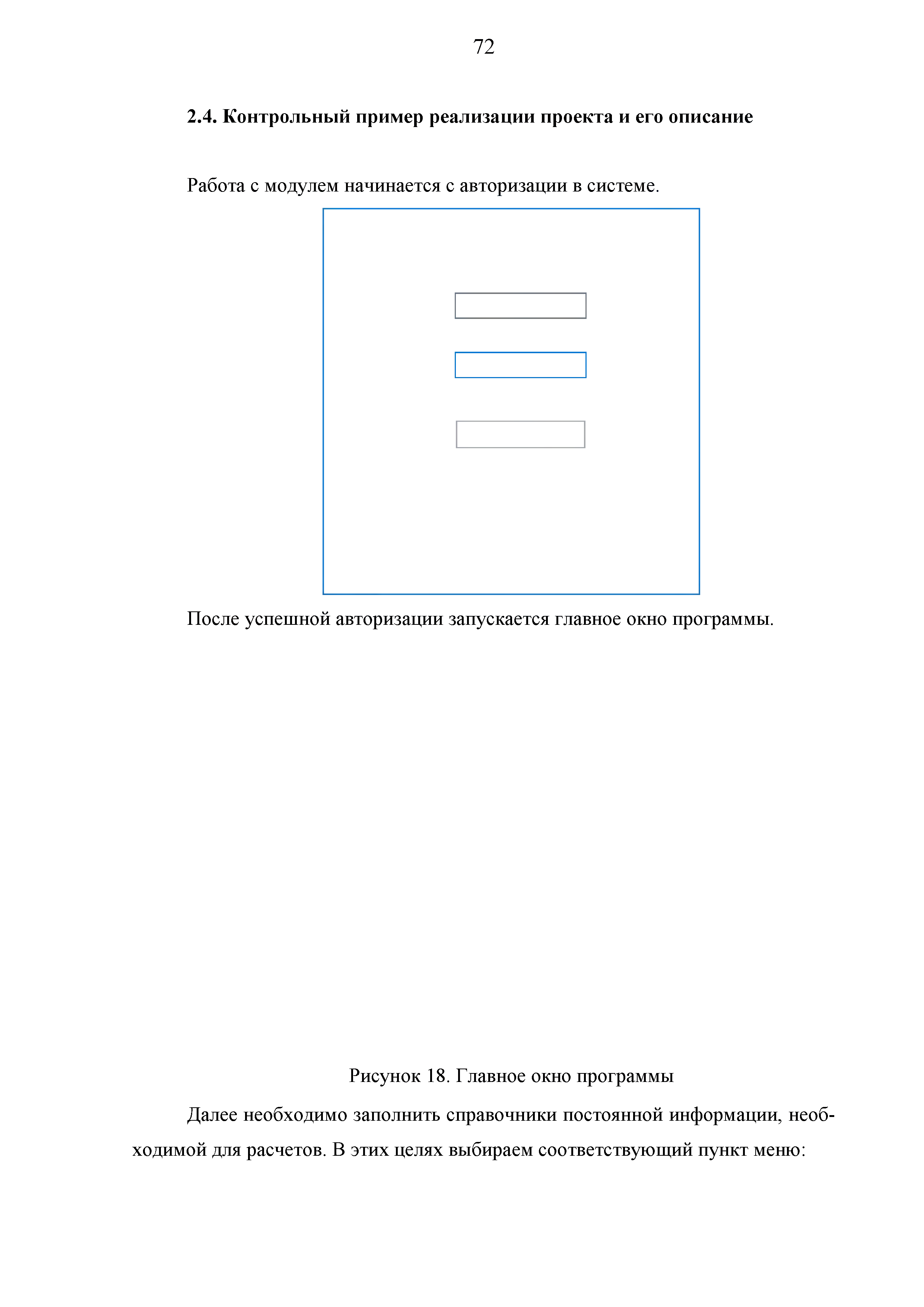
Оклад= ТарифнаяСТавка \* ТарифныйКоэффици ент / ЧасыНорма \*

**U r-1,.,,1th k'Тl,tUP..fk'll1P.**

Загруз ка таблицы «ставки»

Рисунок 17 Блок схема работы обработчика кнопки «Рассчитать»

72



* 1. **Контрольный пример реализации проекта и его описание**

Работа с модулем начинается с авторизации в системе.

После успешной авторизации запускается главное окно программы.

Рисунок 18. Главное окно программы

Далее необходимо заполнить справочники постоянной информации, необ­ ходимой для расчетов. В этих целях выбираем соответствующий пункт меню:

73

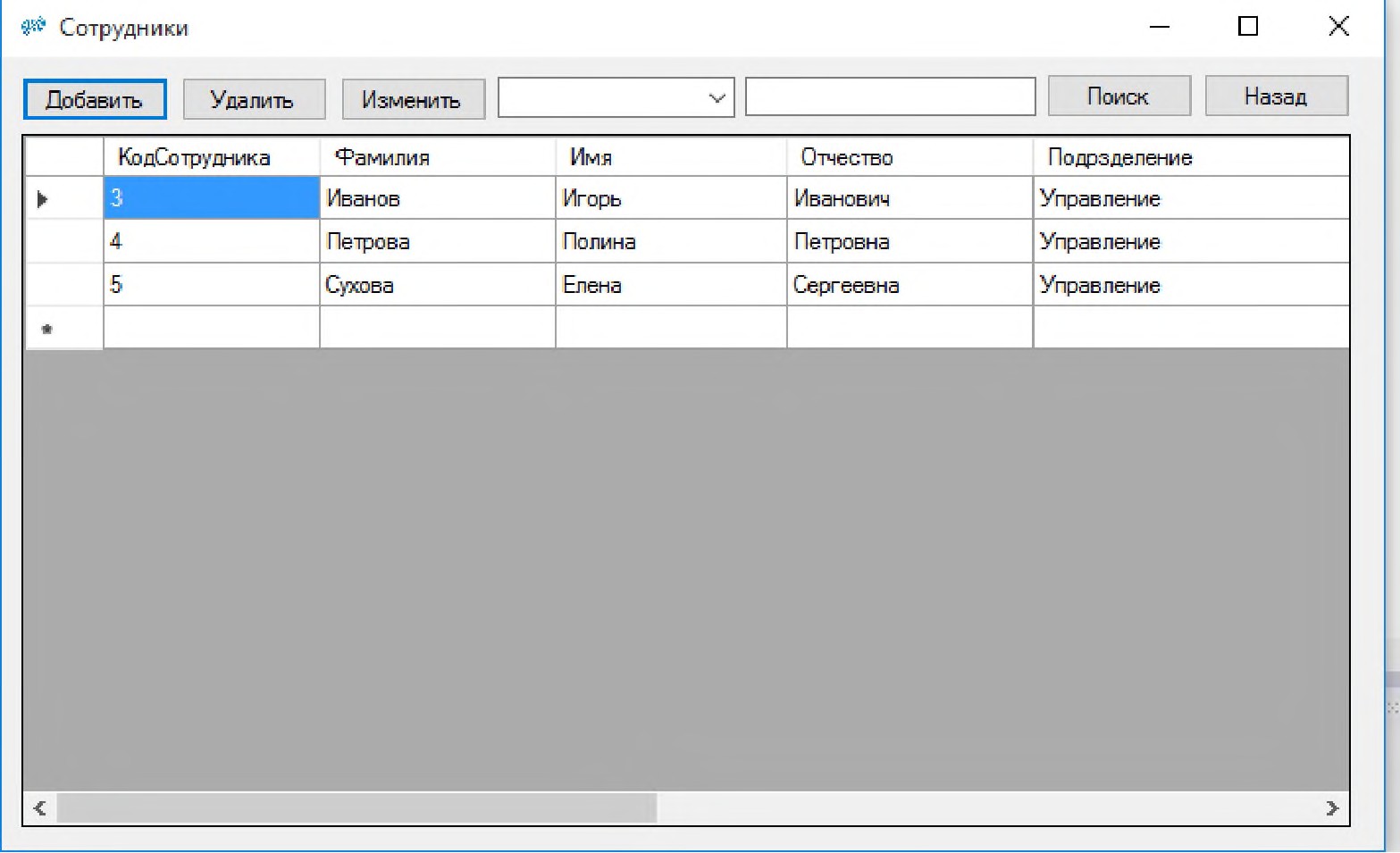
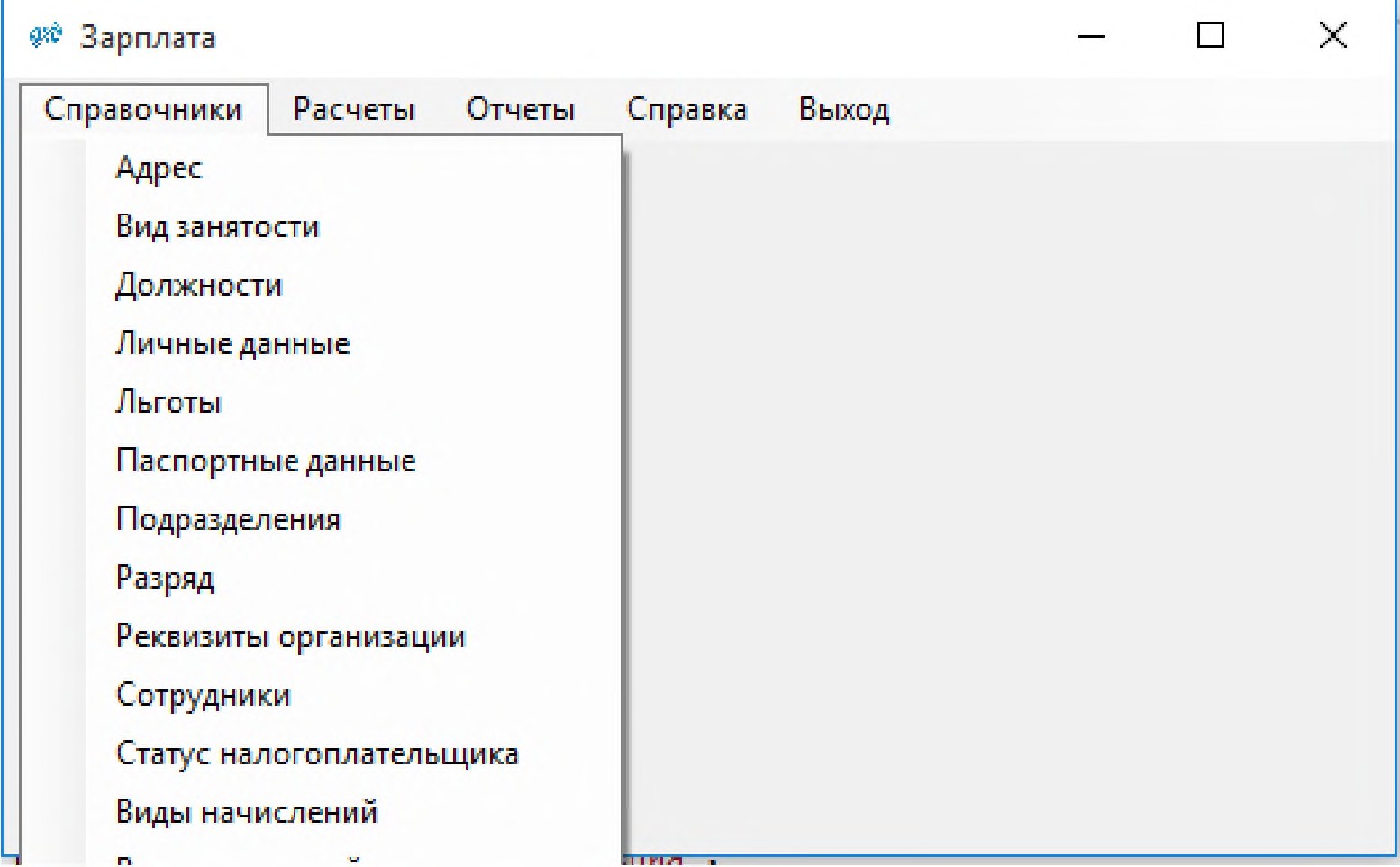
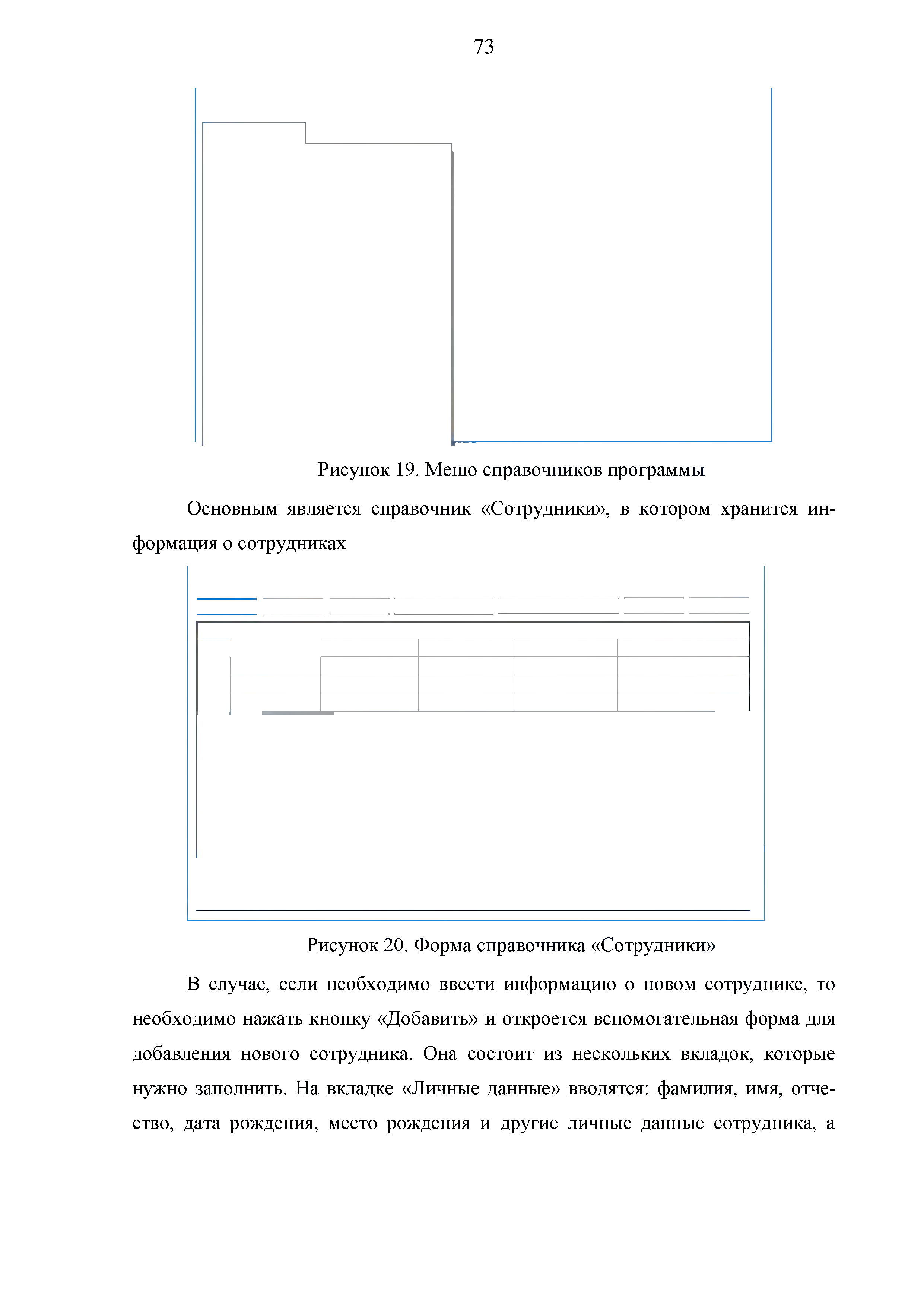


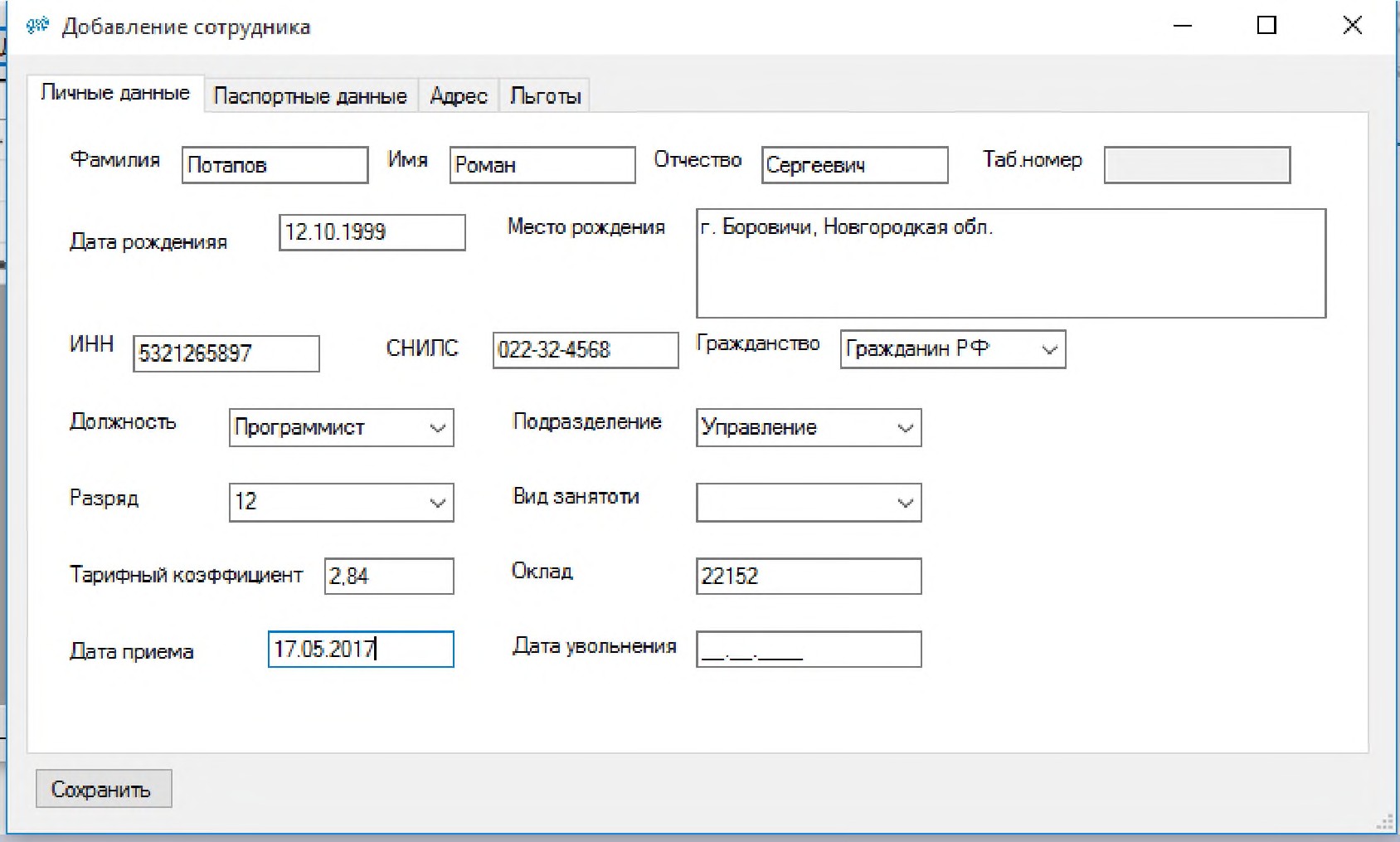
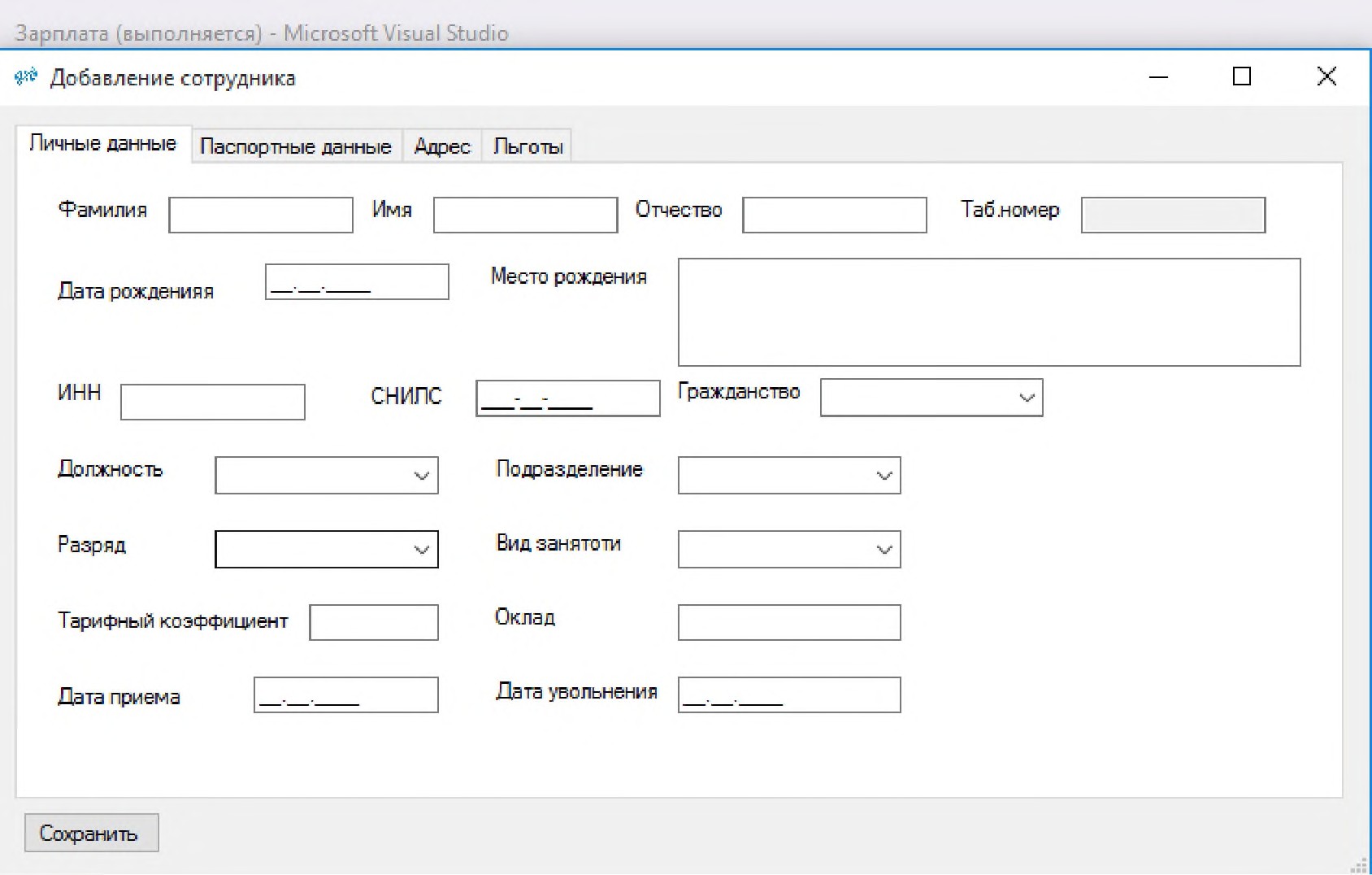
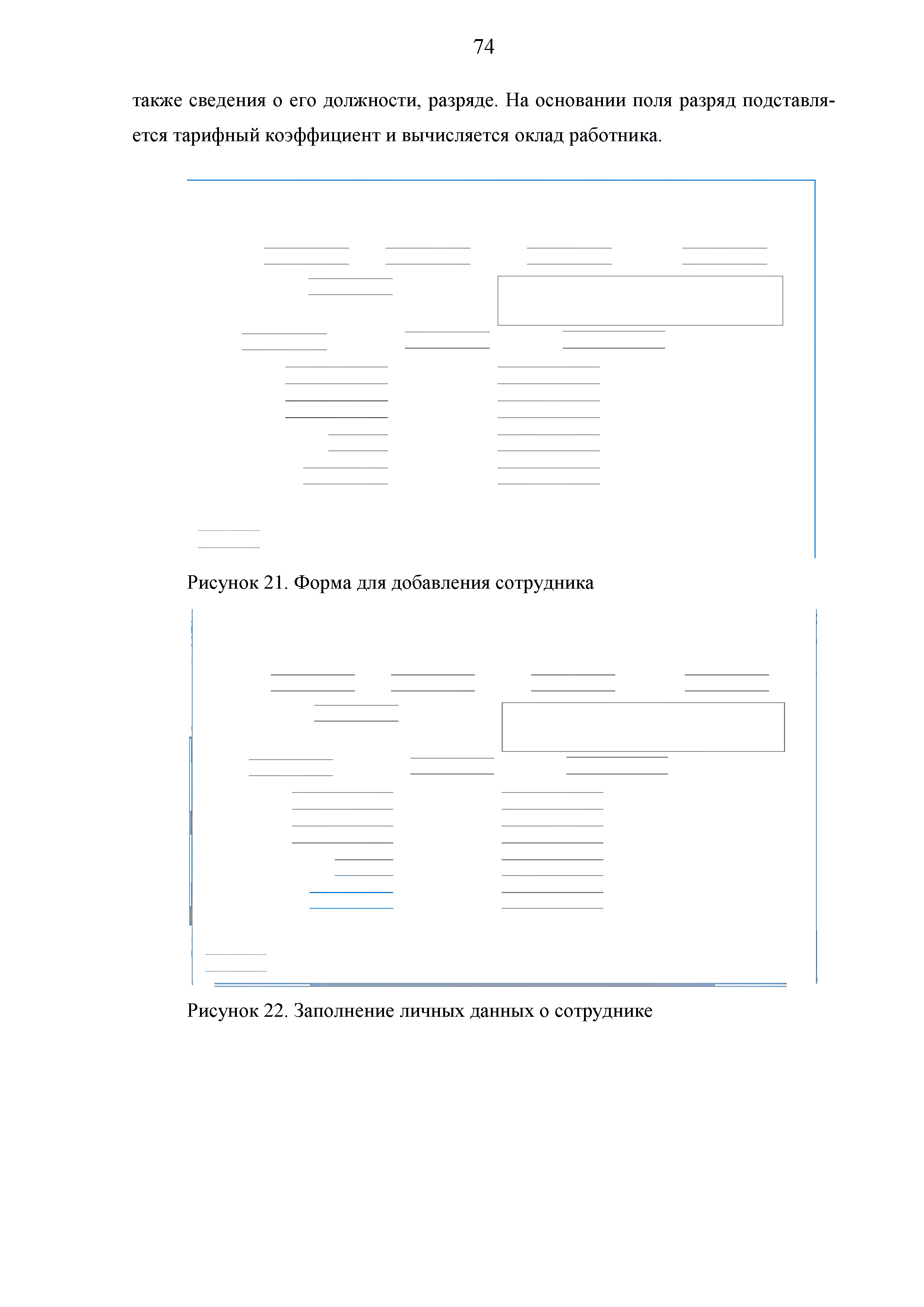
Рисунок 19. Меню справочников программы

Основным является справочник «Сотрудники», в котором хранится ин­ формация о сотрудниках

Рисунок 20. Форма справочника «Сотрудники»

В случае, если необходимо ввести информацию о новом сотруднике, то необходимо нажать кнопку «Добавить» и откроется вспомогательная форма для добавления нового сотрудника. Она состоит из нескольких вкладок, которые нужно заполнить. На вкладке «Личные данные» вводятся: фамилия, имя, отче­ ство, дата рождения, место рождения и другие личные данные сотрудника, а

74



также сведения о его должности, разряде. На основании поля разряд подставля­ ется тарифный коэффициент и вычисляется оклад работника.

Рисунок 21. Форма для добавления сотрудника

Рисунок 22. Заполнение личных данных о сотруднике



Дабавление сотрудника

□

х

Личные данные Паспортные данные Адрес Льготы

Серия Номер

Коn подразделени:Я

Кем выдан

Датавыдачи 1-·-·­

Датасведений 1-·-·-

Срок деЙС1ВИЯ 1-·-·-

1 СохраниТh

Рисунок 23. В8к;,лад4ка «Паспорт3н<ые д1анные»? ? 6

|  |  |
| --- | --- |
| Дата выдачи | 112102005 |
| Дата сведений | 117052017 |
| Срок дейстеия | 112102025 |

Рисунок 24. З8аAпо, лнLен1ие вкладки «Паспортные да3н<ные»1 ? ? 6

Добавление сотрудника

□

х

Личные данные Паспортные данные Аорес Льго-ть1

Серия

14901

Номер

1325415

Код подразделения

Кем выдан

1 У'!' МС по Новгородской облати

1 СохраниТh

Добавление сотрудника

□

х

Личные данные Паспортные данные Дцрес Пьго-ть1

Индекс Регион Район

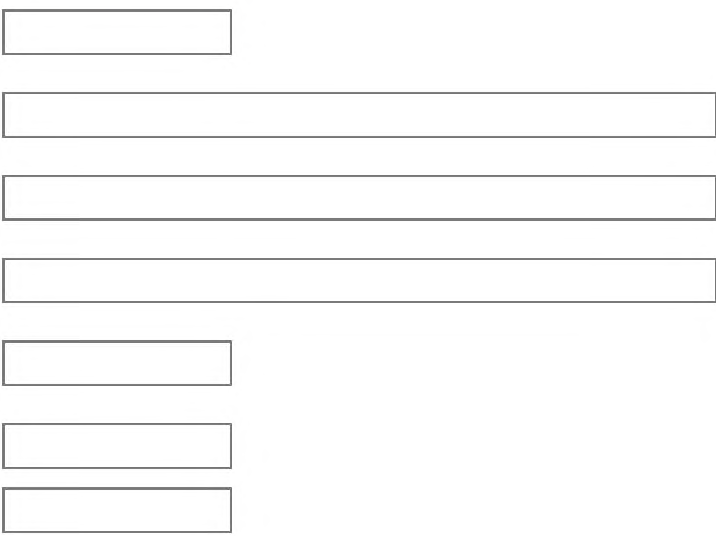
Населенный пункт

Улица

Дом

Квартира

1 Со,:раниТh 1

Рисунок 25. В8к@ла, дк4а «Адрес»

3! 6

Добавление со,трудника

□

х

Пичные данные Паспортные данные Адрес ПьгоТhl

Индекс

1174411

Регион

1 Новгородская область

Район

1 Боровичский

Населенный п rнкт

1 Боровичи

Улица

lсельскаR

Дом

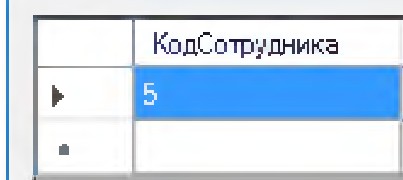
Квартира

СохраниТh

Рисунок 26. З8аCп,олнLен1ие вкладки «Адрес» 3! 6

Для того чтобы найти? нужного сотрудDника можно воспользоватьс0я кDноп­ 1

кой «По3и<сю> 6



"' Сотрудники

1 Добавить 11 Удалить 11 Изменить I .\_'fI'\_ам\_и\_л\_ия

'f'амилия Имя

Сухова Елена

\_,1 !Сухова

От,,ество Сергеевна

□

11 Поиск 11 Назад Подрзделение

Управление

х

)

(

Рисунок 27. Результат поиска по справочнику «Сотрудники»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Разряд ТаифныйКоэффициент "' | | | | |
| 1 | | | 1 | ,-  ..,, |
|  | 2 | 2 | 1.16 |
| з | з | 1,35 |
| **4** | **4** | 1.57 |
| 5 | 5 | 1.73 |
| 6 | 6 | 1,9 |
| 7 | 7 | 2.ОЗ |
| р, | р, | 2,17 |
| 9 | 9 | 2.32 |
| 1{1 | 1{1 | 2,4Р, |
| 11 | 11 | 2,65 |
| 12 | 12 | 2,84 |
| 13 | 13 | З,[14 |
| 14 | **14** | З.25 |
| 15 | 15 | нз |
| 16 | 16 | З.72 |
| 17 | 17 | З,ЭЗ |
| а" |  | .,,. |

Рисунок 28. Справочник «Разряды»

Ра:.ряды

□

х

Изменить

"

После того, как заполнятся все справочники, можно выполнять расчеты по начислению оплаты труда.

'.1-W Опл;з,та по окладам □ х

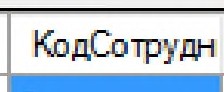
Дата НаЧИСЛеНИЯ 130 ИIОНЯ 2{)17 Г. □..1,Месячная норма времени

1l\_60

Наччисление по окладу Подразделение !Управление

Сотрудник IИваН(l)В

Отработано за месяц l.\_1 60 ,

Начислено 1 016 Рассчита,ъ Сrо:рани,ъ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 'Рамилия   * Иван(l)В | | | Имя  Игорь | | ... | Отчество Наимен(l)ваt  Иванович ... Оклад ... | | ДатаНачисr 3\_1}5\_2{)17 | Сумма  **2{Ю[Ю** |
| \* | **4**  3 | Петрова  Иван(l)В | Полина  Игорь | | ...  ... | Петровна ... Оклад  Иванович ... Оклад | ...  ... | 3\_11}5.2{)17  30.06.2017 | **15000**  2:9016 |
|  | | |  | | | | | |
|  | | | | | | | | | |

Рисунок 29. Расчет оплаты труда по окладам

Листинг программных модулей приведен в Приложении 1.

## 111 ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА

# Выбор и обоснование методики расчёта экономической эф­ фективности

Оценка экономической эффективности информационной системы являет­ ся процессом, заключающимся в определении, понимании и измерении полезно­ сти внедрения информационной системы для предприятия. Как правило, под экономической полезностью понимается денежный эквивалент изменения дохо­ дов организации по итогам инвестирования в информационную систему.

Метод оценки эффективности ИС - это набор средств или способ проведе­ ния полной оценки ИС. Методы могут состоять как из формальных, так и из не­ формальных процедур. При этом под формальными понимаются более объек­ тивные, рациональные, базирующиеся на недвусмысленных данных механизмы оценки, а под неформальными понимаются не основанные на цифровых данных, быстрые, преимущественно субъективные процедуры оценки.

Внедрение современных информационных технологий высоко затратно, однако функционирование компаний в рыночной среде требует, как минимум анализа экономических последствий, а еще больше - оценки экономической эф­ фективности того или иного шага преобразования системы управления компани­ ей.

Оценка экономической эффективности ИС является сложной и трудоем­ кой работой, требующей не только технических, но и экономических навыков. Именно сочетание данных двух составляющих может дать в итоге достоверный результат проводимого анализа.

Продвижение ИС на рынке в условиях современной конкуренции невоз­ можно без предоставления результатов оценки ожидаемой эффективности си­ стемы. Более того, актуальная статистическая оценка успешности внедрения си­ стем управления предприятием характеризуется неудачей внедрения от сорока до семидесяти процентов случаев.

Специалистам в области разработки, внедрения и сопровождения ИС необходимо обладать навыками проведения предварительной экспертизы проек­ та. Данным специалистам следует уметь вести постоянный мониторинг системы на соответствие внедряемых технологий стратегии развития предприятия. Про­ цесс соизмерения затрат и достигаемого за их счет эффекта должен являться именно «процессом», он должен быть итерационной процедурой, которая про­ водится на протяжении всего этапа разработки и внедрения проекта, и результат которой способен повлиять на дальнейшее продолжение проекта.

Оценка экономической эффективности информационной системы может проводиться несколькими способами, отраженными на рисунке ниже.

традиционная оценка эффективности как соотношение затрат и результатов

расчет совокупной стоимости владения информационной системой

оценка внедрения информационной системы как инвестиционного проекта

эффекта

разработка сбалансированной системы показателей для оценки экономического

Вне зависимости от технических, технологических, финансовых, отрасле­ вых или региональных особенностей, оценка эффективности проектов осу­ ществляется на основе единых принципов. К ним относятся:

многоэтапность оценки;

учет всех наиболее существенных последствий проекта; учет фактора времени;

сопоставимость условий сравнения различных проектов;

учет только предстоящих в ходе осуществления проекта затрат и поступлений;

положительность и максимум эффекта; моделирование денежных потоков; учет влияния инфляции;

сравнение «с проектом» и «без проекта»;

учет наличия разных участников проекта;

рассмотрение проекта на протяжении всего жизненного цикла; учет влияния неопределенностей и рисков.

В целом показатели коммерческой эффективности проекта отражают фи­

нансовые последствия внедрения ИС. Для расчета коммерческой эффективности проекта в качестве основных показателей рекомендуется использовать следую­ щие:

чистый доход;

чистый дисконтированный доход; индексы доходности затрат и инвестиций; внутренняя норма доходности;

срок окупаемости.

Резюмируя вышеизложенное, можно отметить, что процесс оценки эконо­ мической эффективности информационных систем достаточно сложен и неодно­ значен. В каждом конкретном случае необходимо подходить индивидуально, при этом опираясь на определенные методики, которые позволят исключить

«человеческий фактор» и уменьшить погрешности вследствие отсутствия каких либо данных.

В рассматриваемом случае, в связи с тем, что система не планируется к продаже, а будет внедрена только на одном предприятии, необходимо рассчи­ тать экономическую эффективность на основе снижения издержек на производ­ ственную деятельность.

При расчете экономической эффективности будет проведено сравнение результатов обработки информации при существующем бизнес-процессе и после внедрения разрабатываемой системы. Прямая эффективность машинной обра­ ботки информации представлена в показателе снижения экономических стои­ мостных затрат на обработку информации. При оценке прямой эффективности в стоимостных единицах измерения рассчитываются две группы показателей - показатель снижения трудовых затрат и показатель снижения стоимостных за­ трат.

При расчете изменения трудовых затрат на обработку информации ис­ пользуется следующая система показателей:

1. Абсолютный показатель снижения трудовых затрат на обработку информации

DT=To-T1 (3.1.1)

анте;

анте.

где То - годовая трудоемкость обработки информации при базисном вари-

Т1 - годовая стоимость обработки информации при проектируемом вари-

1. Коэффициент снижения трудовых затрат

Кт=(DТ/То)\* 100 (%) (3.1.2)

1. Индекс снижения трудовых затрат, отражающий рост производительно­

сти труда при обработке информации.

Ут=То/Т lв (3**.1.3)**

К стоимостным показателям относят абсолютное снижение стоимостных затрат (DC) , коэффициент относительного снижения стоимостных затрат (Кс) индекс снижения стоимостных затрат (Ус):

* 1. Показатель снижения стоимостных затрат

DC=Co-C1 (3.1.4)

где Со - годовая стоимость обработки информации при базисном вариан-

те;

анте.

С1 - годовая стоимость обработки информации при проектируемом вари-

* 1. Коэффициент эффективности по затратам:

Kc=(DC/Co)\* 100 (%) (3.1.5)

* 1. Индекс изменения стоимостных затрат

Ус=Со/С1 **(3.1.6)**

Более того, кроме рассмотренных показателей необходимо также рассчи­

тать срок окупаемости затрат на внедрение проекта машинной обработки ин­ формации (Ток), рассчитываемые в годах, долях года или в месяцах года:

Ток= Кп /DC (3.1.7),

где Кп - затраты на создание проекта (проектирование и внедрение). А также расчетный коэффициент эффективности капитальных затрат: Ер= 1/Ток (3.1.8)

## Расчёт показателей экономической эффективности проекта

В целях обоснования эффективности проекта неообходимо рассчитать за­ траты на выполнение работ в базовом варианте и на выполнение работы после внедрения задачи автоматизации.

Рассчитаем трудовые затраты по базовому и проектируемому варианту. А) Базовый вариант.

Для начала отметим, что объём выполняемой работы измеряется количе­ ством времени, которое затрачивает бухгалтер на выполнение расчетов начисле­ ний по оплате труда и удержаний из заработной платы. Необходимо, однако, помнить, что учет времени ведется на основании табеля учета рабочего времени. Бухгалтер анализирует имеющуюся информацию без использования средств ав­ томатизации.

Бухгалтер по учету оплаты труда работает неравномерно в течение меся­ ца, так как в на протяжение месяца бухгалтер вносит справочную информацию и отражает отработанное время или произведенные работы, а в конце отчетного периода бухгалтер должен произвести весь объем начислений и удержаний.

Норма выработки в час при базовом варианте насчитывает 1,5 операций. Трудоёмкость (Тб) определяется делением объёма работы на норму выра-

ботки.

Тб =180 /1,5 = 120 час.

Б) Проектный вариант.

Норма выработки в проектном варианте рассчитана из числа операций,

которые выполняет специалист за один час работы с использованием разрабо­ танной системы.

Норма выработки в час при проектном варианте составляет 12,0 операций. Трудоёмкость (Тп) считается как отношение объёма работ к норме выра­

ботки:

Тп = 180 /12,0 = 15 час.

Абсолютный показатель снижения трудовых затрат(ЛТ): ЛТ=Тп-Тб,

ЛТ= 120-15=105 час.

Коэффициент относительного снижения трудовых затрат (Кт): Кт=ЛТ/Тб

Кт=105/120\* 100%=87,5%

Индекс изменения трудовых затрат (индекс повышение производительно- сти труда Ут):

Ут=Тб/Тп.

Ут =120/15=8

Выполним расчет стоимостных показателей по базовому и проектируемо- му варианту:

А) Базовый вариант

1. Материальные затраты:

К материальным затратам на обработку информации относят все затраты на расходные материалы (канцтовары): ручки, карандаши, скоросшиватели, ла­ стики, линейки, корректор, бумагу белую формата А4. ТОО «Вершина» ежеме­ сячно расходует на это 350 рублей.

Картриджи: 1 шт. \* 380 руб.= 380 руб.; Итого: 730 руб. в месяц

Итого на год потребуется:

730 \* 12 = 8760 руб.

1. Трудовые затраты.

Для расчета заработной платы необходим один сотрудник. Фонд оплаты труда сотрудника - 200руб./час. \* 180 час.= 36 ООО рублей в месяц.

Фонд заработной платы за год составляет:

36 ООО \* 12 = 432 ООО руб.

Всего затраты по базовому варианту обработки информации составляет: Сб =432 000+8 760=440 760 руб.

Б) Проектный вариант.

1. Материальные затраты:

К материальным затратам на обработку информации в проектируемой си- стеме относятся следующие расходные материалы:

картридж **1** шт. \* 380 руб.= 380 руб.;

бумага **1** шт.\* 210 руб.= 210 руб.; Итого: 590 руб. в месяц Соответственно на год потребуется:

590 \* 12 = 7080 руб.

1. Трудовые затраты.

Количество работников, необходимых для расчета заработной платы, ана­ логично базовому варианту - один человек. Фонд оплаты труда одного сотруд­ ника - 200руб./час. \* 15 = 3000 рублей в месяц.

Фонд заработной платы за год составляет:

3000 х 12= 36000 руб.

1. Накладные расходы состоят из стоимости потреблённой электроэнер­ гии, расходов по сопровождению программного продукта, амортизации про­ граммного продукта:

2500Кв/ч\*2,40руб.=6 ООО руб. Сопровождение - 30 ООО руб. в год.

Срок действия проекта - 5 лет, тогда амортизационные отчисления соста- вят 20% (100% / 5) от капитальных вложений.

Капитальные вложения (инвестиции) в создание проекта. Суммарная стоимость проекта 407 750,00 руб.

Сумма амортизационных начислений составит 20 % от капитальных вло­ жений:

407 750,00 \* О, 2 = 81550,00 руб.

Всего затраты на автоматизированную обработку информации составля-

ют:

Сп= 7080+36000+6000+30000+81550= 166630 руб.

Теперь рассчитаем значение стоимостных показателей: Абсолютное снижение стоимостных затрат (ЛС):

ЛС=Сб-Сп,

ЛС =440760- 166630= 274130 руб.

Коэффициента относительного снижения стоимостных затрат (Кс): Кс=ЛС/Сб\*100%

Кс=274130/440760\*100%=62,2 %

Индекс изменения стоимостных затрат (Ус):

Ус=Сб/Сп. Ус=440760/166630=2,65

Рассчитаем экономическую эффективность от внедрения разработанной системы. Экономическая эффективность является отношением экономического эффекта, который был получен за определённый период времени от вложения капитала, к объёму вложенного капитала, который обеспечил внедрение меро­ приятия и получение экономического эффекта.

Экономический эффект является чистым доходом (прибылью), то есть- стоимость проекта - (минус) себестоимость.

Формула экономической эффективности имеет вид:

Е=ЛС/Кз=l/Т

где: ЛС - прибыль, полученная в течение года от эксплуатации внедрённо- го проекта;

Кз - объём инвестиций, за счёт которого обеспечен доход. Е =274130/407750=0,67

Срок окупаемости (Т) дополнительных капитальных затрат на модерниза­ цию информационной системы характеризует эффективность капитальных за­ трат:

Выведем срок окупаемости проекта. Он определяется, как обратная вели- чина экономической эффективности:

Т = 1 /Е

Т = 1 /0,67 = 1,5

Экономические показатели, которые были получены в результате расчета экономического обоснования дипломного проектирования, представлены в таб­ лице 18.

87

Для наглядности представим полученные результаты графически с по­ мощью гистограмм. Графические диаграммы, представлены на рисунке 30 (сравнительная диаграмма трудовых затрат) и 31 - сравнительная диаграмма стоимостных затрат.

Таблица 5

Экономические показатели проекта

|  |  |  |
| --- | --- | --- |
| Наименование показателей эффективности | Ед. измерения | Результат |
| Необходимые инвестиции на разработку си­ стемы | руб- | 407750,00 |
| Абсолютное снижение стоимостных затрат | руб. | 274130,00 |
| Индекс изменения стоимостных затрат | руб. | 2,65 |
| Годовой экономический эффект | руб. | 168855,00 |
| Экономическая эффективность проекта |  | 0,67 |
| Срок окупаемости проекта | лет | 1,5 |

500000

400000

300000

200000

100000

0

1

* Базовый
* Проектный

Рисунок 30. Сравнительная диаграмма трудовых затрат

500000

400000

300000

200000

100000

0

1

Базовый Проектный

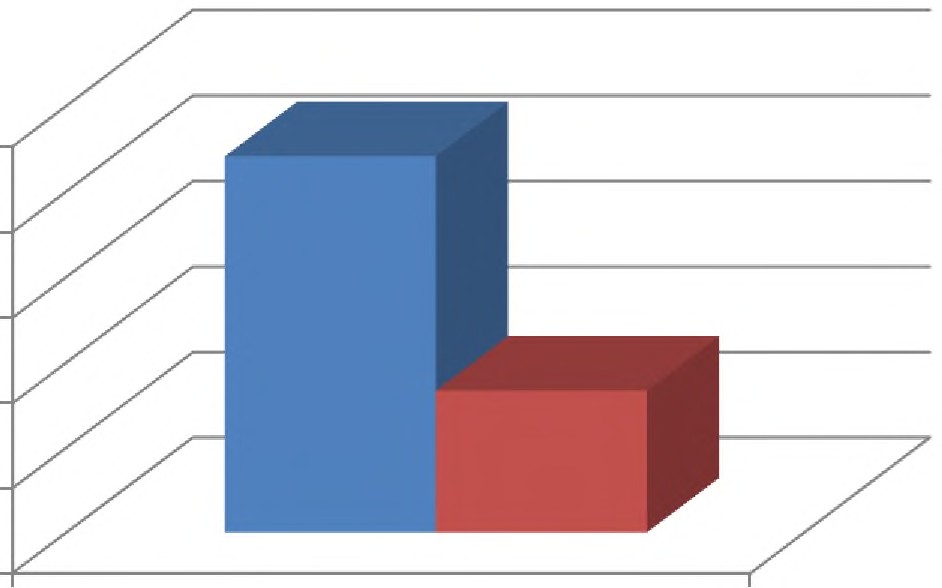
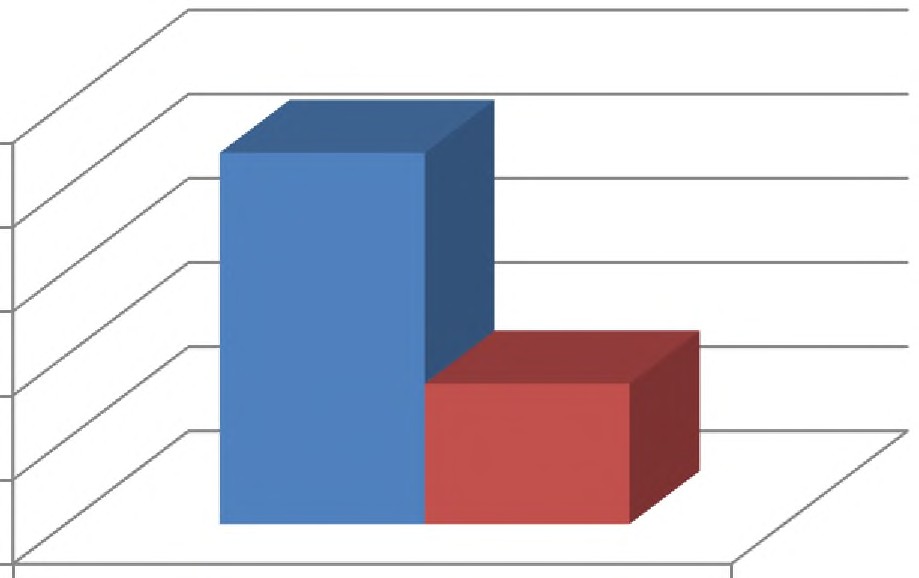
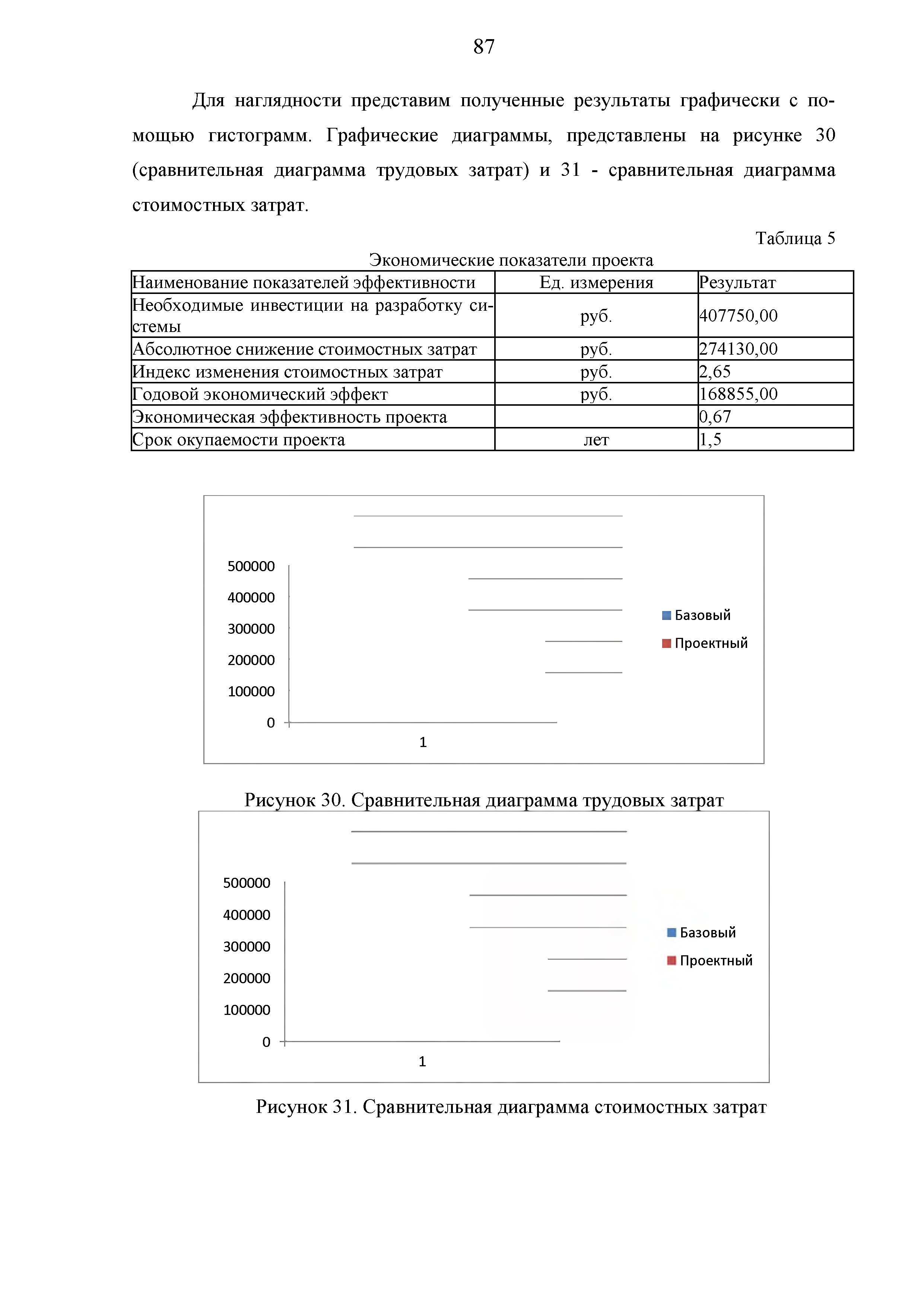


Рисунок 31. Сравнительная диаграмма стоимостных затрат

Вместе с тем, имеются также и косвенные, неизмеримые в денежном вы­ ражении, показатели экономической эффективности внедрения ИС, такие как: прозрачность бизнеса; улучшение имиджа; уменьшение количества ошибок; увеличение скорости оформления сделок и т. д.

## ЗАКЛЮЧЕНИЕ

В процессе работы над данным проектом была всесторонне изучена дея­ тельность ТОО «Вершина», рассмотрена организационная структура предприя­ тия, порядок его функционирования и особенности построения системы автома­ тизации данной организации. Также была тщательно изучена задача автоматиза­ ции процесса расчета начисления заработной платы, исчисления индивидуаль­ ного налога на доходы физических лиц и социального налога. В процессе работы установлено место данной задачи в комплексе задач автоматизации, аргументи­ рована необходимость автоматизации учетных операций и необходимость ис­ пользования вычислительной техники для решения этой задачи.

В аналитической части проекта произведен комплекс работ, направленных на доказательство необходимости автоматизации: установлена сущность задачи, описаны основные требования к системе, дано описание всехныне существую­ щих на предприятии бизнес-процессов, рассмотрены особенности разработок, существующих в этой области. Также в первой главе были подготовлены про­ ектные решения по информационному, программному, техническому и техноло­ гическому обеспечению.

Далее был проведен анализ существующих разработок в этой области, сделан вывод о необходимости разработки собственного программного продук­ та. Выбрана стратегия автоматизации и определены необходимые шаги автома­ тизации и используемые информационные системы.

Основной целью реализации проекта является автоматизация процесса начисления заработной платы, хранения и обработки данных о ранее произве­ денных начислениях, вывод отчетов.

На предприятии применяются две основные формы оплаты труда: повре­ менная и сдельная. Повременная оплата труда ставит размер заработка в зависи­ мость от отработанного времени. Сдельная оплата труда ставит заработок в за­ висимость от количества выполненной работы. В ТОО «Вершина применяются повременно-премиальная и сдельно-премиальная системы оплаты труда.

В процессе разработки была выбрана стратегия автоматизации по участ­ кам, так как предполагается модификация имеющейся на предприятии автомати­ зированной системы расчета и начисления заработной платы для автоматизиро­ ванного выполнения учетных операций.

Информационная система в ТОО «Вершина» разрабатывается собствен­ ными силами, так как организация располагает квалифицированными сотрудни­ ками для разработки и внедрения данной информационной системы.

Для разработки приложения использованы следующие инструменты: Операционная система Windows предназначенная для функциони­ рования на клиентской машине;

Операционная система Windows серверного типа, предназначенная для функционирования на сервере и обеспечивающая многопользо­ вательское подключение к клиент-серверной СУБД

СУБД MS SQL Server для обеспечения хранения базы данных и многопользовательской работы в клиент-серверном варианте функ­ ционирования системы;

Учитывая все преимущества реляционных моделей данных для представ­ ления информации, обрабатываемой при решении задачи, была выбрана реляци­ онная модель базы данных.

Разрабатываемая система предназначена для работы в операционной си­ стеме Windows. Программа должна работать в любой операционной системе се­ мейства Windows: Windows7/8/8.l/10, а также на серверных вариантах этих опе­ рационных систем.

Проектная часть посвящена рассмотрению этапов жизненного цикла про­ екта. Также дана характеристика информационной архитектуры разрабатывае­ мого проекта, построена информационная модель задачи, проведено моделиро­ вание «сущность-связь» (ЕR-модель), описана структура полей таблиц базы дан­ ных, проанализированы все информационные потоки входной, оперативной, нормативно-справочной и результатной информации.

Для жизненного цикла текущего проекта была выбрана каскадная модель, так как она больше подходит для разрабатываемой системы. Переход к каждому

следующему этапу происходит только после завершения всех работ на преды­ дущем этапе, включая подготовку полного пакета документации, достаточной для того, чтобы разработка могла быть продолжена другой группой разработчи­ ков. При данной модели есть возможность планирования сроков завершения ра­ бот и затрат на их выполнение.

На этапе концептуального проектирования были выделены следующие типы, относящиеся к данной предметной области: «Реквизиты фирмы», «Спра­ вочная информация о сотрудниках», «Справочная информация по должностям»,

«Справочная информация об уровне МРП», «Справочная информация об уровне МРОТ» и «Начисления заработной платы».

На этапе логического проектирования разработанная модель преобразова­ на в более «пригодную» структуру для реализации ее в среде существующих СУБД, причем с ориентацией на выбранную СУБД. Разрабатываемое приложе­ ние представляет собой локальную БД, т.е. хранимые данные находятся на том же компьютере, что и приложение для работы с БД.

Входные данные: информация о реквизитах фирмы; информация о долж­ ностях и подразделениях; уровне МРП и МРОТ; информация, о сотрудниках включая ФИО, дату приема, дату увольнения, размер оклада, количество человек находящихся на иждивении, наличие льгот.

Выходные данные: расчет начисленных и удержанных сумм, расчет сумм подлежащих выплате, суммы начисленного налога на доходы физических лиц, суммы социального налога.

Далее в среде разработки была разработана и реализована программа, вы­ полняющая поставленную задачу. Были созданы справочники, документы, отче­ ты; определена их структура, реквизиты, формы, написаны программные моду­ ли, реализующие заданную функциональность этих объектов. В процессе реали­ зации проектных решений по программному обеспечению построены: дерево диалога (сценарий работы с системой), структурная схема пакета и ряд других компонент проекта, подробно раскрывающих сущность машинной реализации задачи.

Проведенное тестирование разработанной автоматизированной системы показывает, что она полностью удовлетворяет поставленной задаче, работает стабильно и без ошибок, обладает большой функциональностью и удобным ин­ терфейсом. Разработанную систему автоматизации можно внедрять в ТОО

«Вершина».

Кроме этого, во второй главе описаны все используемые программные модули, построена схема технологического процесса сбора, передачи, обработки и выдачи информации, описан контрольный пример реализации проекта.

Третья часть дипломного проекта посвящена выбору методики расчета экономической эффективности и собственно проведению всех расчетов, направ­ ленных на обоснование экономической эффективности проекта. Полученные ре­ зультаты расчетов свидетельствуют об экономической эффективности проекта. Исходя из выше сказанного, можно сделать вывод о том, что в результате ди­ пломного проектирования цель достигнута и поставленные задачи решены.

Годовой экономический эффект от реализации данного проекта составля­ ет 168855,00 руб. Срок окупаемости проекта составляет 1,5 года.

В конце мы хотели бы отметить, что разработанная ИС может использо­ вана через облако, что дает еще большие преимущества для организации и мо­ жет ускорить окупаемость.

## Литература

1. Трутнев Д. Р. Архитектуры информационных систем. Основы проектиро­ вания: Учебное пособие. - СПб.: НИУ ИТМО, 2012. -66 с.
2. Автоматизированные информационные системы, базы и банки данных. Вводный курс: Учебное пособие, М.: Гелиос АРВ, 2002. - 368 с., ил
3. Алистер Коберн, Современные методы описания функциональных тре­ бований к системам, М.: издательство «Лори», 2002. - 263 с
4. Алистер Коберн, Современные методы описания функциональных тре­ бований к системам
5. Астелс, Дэвид; Миллер Гранвилл; Новак, Мирослав, Практическое ру­ ководство по экстремальному программированию, Пер. с англ. - М.: Издательский дом «Вильяме», 2002. - 320 с.: ил. - Парал. тит. Англ
6. Гайфуллин Б.Н., Обухов И.А. Автоматизированные системы управле­ ния предприятиями стандарта ERP/МRPII. Производственное издание. М. «Богородский печатник», 2001, 104 с
7. Бек К Экстремальное программирование, СПб.: Питер, 2002. - 224 с.
8. Брауде Э. Технологии разработки программного обеспечения.СПб: Пи­ тер, 2004. - 655 с.: ил
9. Введение в системы баз данных - СПб: Издательский дом «Вильяме», 2000. - 848 с.;
10. Вендров А.М., САSЕ-технологии. Современные методы и средства проектирования информационных систем - М.: Финансы и статистика, 2006.
11. Вигерс Карл, Разработка требований к программному обеспечению, Пер, с англ. - М.:Издательско-торговый дом «Русская Редакция», 2004.

-576с.: ил

1. Гаджинский А.М. Основы логистики: Учеб.пособие/ Информ.­ внедрен.центр «Маркетинг».- М., 2005.- 121, с.: ил., табл.
2. ГОСТ 34.601-90. Информационная технология. Автоматизированные системы. Стадии создания
3. ГОСТ Р ИСО/МЭК 12207/99. Государственный стандарт РФ. Информа­ ционная технология. Процессы жизненного цикла информационных систем. Издание официальное. - М., 1999
4. Дейв Крейн, Эрик Паскарелло, Даррен Джеймс. AJAX в действии: Учебник - М.: Вильяме, 2006. 450 - 490 с.
5. Джексон Г. Проектирование реляционных баз данных для использова­ ния с микро-ЭВМ М.: Финансы и статистика, 1991.
6. Диго С.М. Базы данных: проектирование и использование: Учебник. - М.: Финансы и статистика, 2005. - 592 с.
7. Дэвид Флэнаган. JavaScript. Подробное руководство: Учебник - М.: Символ Плюс, 2008. 243 - 249 с.
8. Дэниел О'Лири. ERP системы. Современное планирование и управле­ ние ресурсами предприятия. Выбор, внедрение, эксплуатация. М.: ООО

«Вершина», 2004. - 272 с, [Пер. с англ. Ю.И.Водопьяновой 20.Зеленков Ю.А. Введение в базы данных. Центр Интернет ЯрГУ, 1997.

1. Зелковиц М., Шоу А., Гэннон Дж. Принципы разработки программного обеспечения/ Пер. с англ. -М.: Мир, 1982. -386 с., ил.
2. Ивлиев М.К., Порошина Л.А. Автоматизация оперативного и бухгал­ терского учета товаров, 1997.
3. Информационные системы: Учебник для вузов. 2-е изд. СПб: «Питер», 2005 г - 656 стр.
4. Керри Н. Праг, Майкл Р. Ирвин, Access 2000 - Библия пользователя, Диалектика, 2000.
5. Крис Дейт. Введение в базы данных, 6-е изд. Киев, Диалектика, 1998.
6. Кристиан Дари, Богдан Бринзаре, Филип Черчез-Тоза, Михай Бусика. AJAX и РНР. Разработка динамических веб-приложений: Учебник - М.: Символ Плюс, 2006.
7. Леффингуелл Д., Уидриг Д, Принципы работы с требованиями к про­ граммному обеспечению, М.: ИД «Вильяме», 2002
8. Макарова Н.В Информатика: Учебник, М.: Финансы и статистика, 2003. - 768 с
9. Марка Д., МакГоуэн К, Методология структурного анализа и проекти­ рования, М.: МетаТехнология, 1993
10. Марка ДА. Методология структурного анализа и проектирования, СПб.: Питер, 1995. - 235 с
11. Мацяшек Лешек, Анализ требований и проектирование систем. Разра­ ботка информационных, Пер. с англ. - М.: Издательский дом «Виль­ яме», 2002. - 432 с.: ил. - Парал. тит. Англ
12. Меняев М.Ф, Информационные технологии управления: Книга 3: Си­ стемы управления организацией, М.: Омега-Л, 2003. - 464 с
13. Орлик С., Булуй Ю, Введение в программную инженерию и управление жизненным циклом ПО Программная инженерия. Программные требо­ вания
14. Петров В. Н, Информационные системы, СПб.: Питер, 2002. - 688 с 37.Практическое руководство по программированию/ Пер. с англ. Б. Мик,

П. Хит, Н. Рашби и др.; под ред. Б. Мика, П. Хит, Н. Рашби. - М.: Ра­ дио и связь, 1986. - 168 с., ил.

1. Проектирование и использование баз данных: Учебник. М.:Финансы и статистика, 1995г. - 191 с.;
2. Разработка программного обеспечения - СПб «Питер», 2004 г - 592 стр.
3. Раскин А.Л., Руководство по применению стандарта ИСО 9001:2000 при разработке программного обеспечения. М.: РИА «Стандарты и ка­ чество», 2002. - 104 с. - («Дом качества», вып. 9 (18))
4. Реляционные базы данных: практические приемы оптимальных реше­ ний. -СПб.: БХВ-Петербург, 2005 -400с.:ил;
5. Симионов Ю.Ф., Боромотов В.В. Информационный менеджмент. - Ростов н.Д: Феникс, 2006, 250с., ил.;
6. Схемы алгоритмов, программ, данных и систем. Уславные обозначения и правила выполнения. ГОСТ 19.701-90 (ИСО 5807-85) / Государствен­ ный комитет СССР по управлению качеством продукции и стандартам, 01.01.1992.
7. Фаулер М, Скотт К, UML в кратком изложении. Применение стандарт­ ного языка объектного моделирования, Пер. с англ. - М.:Мир, 1999. - 191 с., ил
8. Фокс Дж. Программное обеспечение и его разработка/ Пер. с англ. - М.: Мир, 1985. - 368 с., ил.
9. Язык компьютера. Пер. с англ, под ред. и с предисл. В. М. Курочкина.

-М.: Мир, 1989. -240 с., ил.

1. Глушаков С.В., Ломотько Д.В. Базы данных, 2000.
2. Якобсон А., Г. Буч, Дж. Рамбо, Унифицированный процесс разработки программного обеспечения
3. Якобсон А., Г. Буч, Дж. Рамбо, Унифицированный процесс разработки программного обеспечения, СПб.: Питер 2002. 496 с

## ПРИЛОЖЕНИЕ

**Листинг программы Главная форма**

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms; using System.Data;

using System.Data.SqlClient;

namespaceЗapnлaтa

{

puЫicpartialclassfGlobal :Form

{

puЫic fGlobal()

{

InitializeComponent();

}

privatevoidвыxoдToolStripMenuitem\_Click(object sender, EventArgs е)

{

Application.Exit();

}

privatevoidaдpecToolStripMenultem\_Click(object sender, EventArgs е)

{

FormAdres fAdres = newFormAdres(); fAdres.Show();

}

privatevoidвидЗaнятocтиToolStripMenultem\_Click(object sender, EventArgs е)

{

FormVidSan fVidSan = newFormVidSan(); fVidSan.Show();

}

privatevoidдoлжнocтиToolStripMenultem\_Click(object sender, EventArgs е)

{

FormDolg fDolg = newFormDolg(); fDolg.Show();

}

privatevoidличныeДaнныeToolStripMenuitem\_Click(object sender, EventArgs е)

{

FormLD fLD = newFormLD(); fLD.Show();

}

privatevoidльгoтыToolStripMenultem\_Click(object sender, EventArgs е)

{

Formlgota fl = newFormlgota();

fL. Show();

}

**98**

privatevoidnacnopтныeДaнныeToolStripMenuitem\_Click(object sender, EventArgs е)

{

FormPasport fPas = newFormPasport(); fPas.Show();

}

privatevoidnoдpaздeлeнияToolStripMenuitem\_Click(object sender, EventArgs е)

{

FormPodrazd fPodrazd = newFormPodrazd(); fPodrazd.Show();

}

privatevoidpaзpядToolStripMenuitem\_Click(object sender, EventArgs е)

{

FormRazrjad fRazrjad = newFormRazrjad(); fRazrjad.Show();

}

privatevoidpe,квизить()pгaнизaцииToolStripМenuitem\_Click(object sender, EventArgs е)

{

FormOrganiz fOrg = newFormOrganiz(); fOrg.Show();

}

privatevoidcoтpyдникиToolStripMenuitem\_Click(object sender, EventArgs е)

{

**FormSotr fSotr = newFormSotr(); fSotr.Show();**

}

privatevoidвид HaчиcлeнийToolStripМenuitem\_Click(object sender, EventArgs е)

{

FormVidNachislenia fVNach = newFormVidNachislenia(); fVNach.Show();

}

privatevoidвидыYдepжaнийToolStripMenuitem\_Click(object sender, EventArgs е)

{

FormVidUderganij fVUdreg = newFormVidUderganij(); fVUdreg.Show();

}

privatevoidнaчиcлeнияToolStripМenuitem\_Click(object sender, EventArgs е)

{

FormTabel fТаЫе = newFormTabel(); fTaЫe.Show();

}

privatevoidнaчиcлeнияПoOклaдaмToolStripMenuitem\_Click(object sender, EventArgs е)

{

FormOklad fOklad = newFormOklad(); fOklad.Show();

}

privatevoidaвтopизaцияToolStripJVlenuitem\_Click(object sender, EventArgs е)

{

FormAuth fa = newFormAuth(); fa.Show();

}

privatevoidнaчиcлeниeПoвpeмeннoйOnлaтыToolStripMenuitem\_Click(object sender, EventArgs

е)

{

FormPovremenno fPovr fPovr.Show();

}

newFormPovremenno();

privatevoidнaчиcлeниeПoвpeмeннoйOnлaтыToolStripMenuitem\_Click\_l(object sender, Even­ tArgs е)

{

FormPovremenno fPovr fPovr.Show();

}

newFormPovremenno();

privatevoidнaчиcлeнияЗaMecяцToolStripMenultem\_Click(object sender, EventArgs е)

{

FormOtchet fOtchet = newFormOtchet(); fOtchet.Show();

}

privatevoid fGlobal\_Load(object sender, EventArgs е)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу «Зарnла­ таDаtаSеt.НачислениеЗарnлатыSеlесt». При необходимости она может быть перемещена или удалена.

// this.HaчиcлeниeЗapnлaтыSelectTaЬleAdapter.Fill(this.ЗapnлaтaDataSet.HaчиcлeниeЗapnлaты Select);

// this.reportViewerl.RefreshReport();

}

}

}

**Форма авторизации**

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms;

using System.Security.Cryptography;

namespaceЗapnлaтa

{

puЫicpartialclassFormAuth :Form

{

puЫicstring CalculateMD5Hash(string input)

{

// шаг **1,** вычислитьхэш MD5

MD5 md5 = System.Security.Cryptography.MD5.Create();

byte[] inputBytes = System.Text.Encoding.ASCII.GetBytes(input); byte[] hash = md5.ComputeHash(inputBytes);

// Шаг 2, преобразование байтового массива в хэш-строку StringBuilder sb = newStringBuilder();

for (int i = 0; i < hash.Length; i++)

{

sb.Append(hash[i].ToString(«X2»));

}

return sb.ToString();

}

internalfGlobal fl; puЫic FormAuth()

{

InitializeComponent();

//fGlobal owner

// this.Owner = owner;

// fl = owner;

}

privatevoid btAuth\_Click(object sender, EventArgs е)

{

string username = tbName.Text; if (username == ««)

{

МеssаgеВох.Shоw(«Введитеимяnользователя», «Ошибка», MessageBoxButtons.OK, MessageBoxI­ con.Information);

return;

}

string pwd = tbPass.Text; if (pwd == ««)

{

MessageBox.Show(«Bвeдитenapoль», «Ошибка», MessageBoxButtons.OK, MessageBoxI­ con.Information);

return;

}

}

}

}

Форма регистрации нового пользователя using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms; using System.Data.SqlClient;

using System.Security.Cryptography;

namespaceЗapnлaтa

{

puЫicpartialclassFormRegistr :Form

{

puЫicstring CalculateMD5Hash(string input)

{

// шаг 1, вычислитьхэш MD5

MD5 md5 = System.Security.Cryptography.MD5.Create();

byte[] inputBytes = System.Text.Encoding.ASCII.GetBytes(input); byte[] hash = md5.ComputeHash(inputBytes);

// Шаг 2, преобразование байтового массива в хэш-строку StringBuilder sb = newStringBuilder();

for (int i = 0; i < hash.Length; i++)

{

sb.Append(hash[i].ToString(«X2»));

}

return sb.ToString();

}

puЫic FormRegistr()

{

InitializeComponent();

}

privatevoid buttonl\_Click(object sender, EventArgs е)

{

string username = tbName.Text; if (username == ««)

{

МеssаgеВох.Shоw(«Введитеимяпользователя», «Ошибка», MessageBoxButtons.OK, MessageBoxI­ con.Information);

return;

}

string pwd = tbPass.Text;

if (pwd == ««)

{

МеssаgеВох.Shоw(«Введитепароль», «Ошибка», MessageBoxButtons.OK, MessageBoxI­ con.Information);

return;

}

string connectionString = Properties.Settings.Default.ЗapnлaтaConnectionString; SqlConnection сопп = newSqlConnection(connectionString);

SqlDataAdapter adapter = newSqlDataAdapter(); DataSetЗapnлaтaDataSet = newDataSet(); ЗарплатаDаtаSеt.ТаЫеs.Аdd(«Аутентификация»);

DataTaЬle tаЫе ЗарплатаDаtаSеt.ТаЫеs[«Аутентификация»]; сопп.Ореп();

SqlCommand comm newSqlCommand(); comm.CommandText = «AuthSelect»;

comm.CommandType = CommandType.StoredProcedure; comm.Connection = сопп;

adapter.SelectCommand = comm; taЫe.Rows.Clear(); adapter.Fill(taЫe);

dgTaЫe.DataSource = tаЫе;

//string sqlCmd = «insert into Аутентификация(UsеrNаmе, PasswordHash) values (@login, @pass)»;

SqlCommand command = newSqlCommand(); command.CommandText = «Auth»; command.Connection = сопп;

command.CommandType = CommandType.StoredProcedure; adapter.InsertCommand = command;

command.Parameters.Add(«@login», SqlDЫype.VarChar); command.Parameters[«@login»].Value = username;

command.Parameters.Add(«@pass», SqlDЫype.VarChar); command.Parameters[«@pass»].Value = CalculateMD5Hash(pwd);

taЫe.Rows.Clear(); adapter.Fill(taЫe);

int rowCount try

0·,

{

rowCount = command.ExecuteNonQuery();

}

catch (Exception е2)

{

MessageBox.Show(e2.Message, «Ошибкасозданияnользователя»);

}

finally

{

conn.Close();

}

if (rowCount != 0)

МеssаgеВох.Shоw(«Пользователь успешно создан», «Зарплата»);

}

}

}

Справочники using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms;

namespaceЗapnлaтa

{

puЫicpartialclassFormAdres :Form

{

puЫic FormAdres()

{

InitializeComponent();

}

privatevoidaдpecBindingNavigatorSaveltem\_Click(object sender, EventArgs е)

{

this.Validate(); this.aдpecBindingSource.EndEdit();

this.taЫeAdapterManager.UpdateAll(this.зapnлaтaDataSet);

}

privatevoid FormAdres\_Load(object sender, EventArgs е)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу «зapnлa­ тaDataSet.Aдpec». При необходимости она может быть перемещена или удалена. this.aдpecTaЬleAdapter.Fill(this.зapnлaтaDataSet.Aдpec); aдpecDataGridView.Columns[0].VisiЫe = false;

}

}

}

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms; using System.Data.Sql;

using System.Data.SqlClient;

namespaceЗapnлaтa

{

puЫicpartialclassFormDolg :Form

{

privateconststringДoлжнocтиEntity privateDataSet dataset; privateSqlDataAdapter adapter; privateSqlConnection connection; privateSqlCommand command; puЫicDataTaЬle tаЫе; privatestring entityName;

puЫic FormDolg()

{

InitializeComponent(); dataset = newDataSet();

dataset.TaЬles.Add(ДoлжнocтиEntity); adapter = newSqlDataAdapter(); tаЫе = dataset.TaЬles[0];

«Должности»;

connection = newSqlConnection(Properties.Settings.Default.ЗapnлaтaConnectionString); command = newSqlCommand();

command.Connection = connection; command.CommandType = CommandType.StoredProcedure;

}

privatevoid FormDolg\_Load(object sender, EventArgs е)

{

slct();

}

puЫicvoid slct() //отображениетаблиц

{

// adapter = new SqlDataAdapter(); tаЫе = dataset.TaЬles[0];

command.CommandText = ДолжностиЕntitу + «Select»; command.Parameters.Clear();

adapter.SelectCommand = command; taЫe.Rows.Clear(); adapter.Fill(taЫe);

dgTaЫe.DataSource tаЫе;

}

privatevoid btlnsert\_Click(object sender, EventArgs е)

{

tаЫе = dataset.TaЬles[0];

command.CommandText = ДолжностиЕntitу + «Insert»; adapter.InsertCommand = command;

command.Parameters.Add(newSqlParameter(«@Name», SqlDbType.NChar)); command.Parameters[«@Name»].Value =

dgTaЫe.CurrentRow.Cells[l].Value.ToString();

taЫe.Rows.Clear(); adapter.Fill(taЫe); slct();

}

privatevoid btDelete\_Click(object sender, EventArgs е)

{

command.CommandText = ДолжностиЕntitу + «Delete»; command.Parameters.Clear(); command.Parameters.Add(newSqlParameter(«@IdDolgnost», SqlDbType.Int));

command.Parameters[«@IdDolgnost»].Value =

Con­ vert.Tolnt32(taЫe.Rows[dgTaЫe.CurrentRow.Index].ItemArray.ElementAt(0).ToString());

adapter.DeleteCommand = command; taЫe.Rows.Clear(); adapter.Fill(taЫe);

slct();

}

}

}

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms; using System.Data.Sql;

using System.Data.SqlClient;

namespaceЗapnлaтa

{

puЫic partial class FormlnsertSotr Form

{

private const string СотрудникиЕntitу = «Сотрудники»; private const string ЛичныеДанныеЕntitу = «ЛичныеДанные»; private const string ПодразделенияЕntitу = «Подразделения»; private const string PaзpядEntity = «Разряд»;

private const string ДолжностиЕntitу = «Должности»;

private const string ПасnортныеДанныеЕntitу = «ПасnортныеДанные»; private const string AдpecEntity = «Адрес»;

private const string СтавкиЕntitу = «Ставки»;

private const string ВидЗанятостиЕntitу = «ВидЗанятости»; private DataSet ЗapnлaтaDataSet;

private SqlDataAdapter adapter;

private SqlConnection connection; private SqlCommand command; puЫic DataTaЬle tаЫе;

private string EntityName;

puЫic FormlnsertSotr()

{

InitializeComponent(); ЗapnлaтaDataSet = new DataSet();

ЗapnлaтaDataSet.TaЬles.Add(PaзpядEntity); ЗapnлaтaDataSet.TaЬles.Add(CoтpyдникиEntity); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ПодразделенияЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ЛичныеДанныеЕntitу); ЗapnлaтaDataSet.TaЬles.Add(ДoлжнocтиEntity); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ВидЗанятостиЕntitу); ЗapnлaтaDataSet.TaЬles.Add(ПacnopтныeДaнныeEntity); ЗapnлaтaDataSet.TaЬles.Add(AдpecEntity); ЗapnлaтaDataSet.TaЬles.Add(CтaвкиEntity);

adapter = new SqlDataAdapter(); tаЫе = ЗapnлaтaDataSet.TaЬles[l];

connection new SqlConnec- tion(Properties.Settings.Default.ЗapnлaтaConnectionString);

command = new SqlCommand(); command.Connection = connection;

command.CommandType = CommandType.StoredProcedure;

}

puЫic void slct() //отображениетаблиц

{

// adapter = new SqlDataAdapter(); tаЫе = ЗapnлaтaDataSet.TaЬles[EntityName];

command.CommandText = EntityName + «Select»; command.Parameters.Clear();

adapter.SelectCommand taЫe.Rows.Clear(); adapter.Fill(taЫe);

}

command;

private void btlnsert\_Click(object sender, EventArgs е)

{

EntityName «Адрес»;

tаЫе ЗapnлaтaDataSet.TaЬles[AдpecEntity];

command.CommandText = EntityName + «Insert»; adapter.InsertCommand = command;

command.Parameters.Add(new SqlParameter(«@Index», SqlDЫype.NChar)); command.Parameters[«@Index»].Value = textBox10.Text;

command.Parameters.Add(new SqlParameter(«@Region», SqlDЫype.NChar)); command.Parameters[«@Region»].Value = textBox11.Text;

command.Parameters.Add(new SqlParameter(«@Rajon», SqlDЫype.NChar)); command.Parameters[«@Rajon»].Value = textBox12.Text;

command.Parameters.Add(new SqlParameter(«@NasPunct», SqlDЫype.NChar)); command.Parameters[«@NasPunct»].Value = textBox13.Text;

command.Parameters.Add(new SqlParameter(«@Ulica», SqlDЫype.NChar)); command.Parameters[«@Ulica»].Value = textBox14.Text;

command.Parameters.Add(new SqlParameter(«@Dom», SqlDЫype.NChar)); command.Parameters[«@Dom»].Value = textBox15.Text;

command.Parameters.Add(new SqlParameter(«@Kvartira», SqlDЫype.NChar)); command.Parameters[«@Kvartira»].Value = textBox16.Text;

taЫe.Rows.Clear();

adapter.Fill(taЫe); slct О;

dgAdres.DataSource tаЫе;

EntityName = «ПасnортныеДанные»; tаЫе = ЗapnлaтaDataSet.TaЬles[EntityName];

command.CommandText = EntityName + «Insert»; adapter.InsertCommand = command;

command.Parameters.Add(new SqlParameter(«@Seria», SqlDЫype.Int)); command.Parameters[«@Seria»].Value = textBox7.Text;

command.Parameters.Add(new SqlParameter(«@Nomer», SqlDЫype.Int)); command.Parameters[«@Nomer»].Value = textBox8.Text;

command.Parameters.Add(new SqlParameter(«@KemVydan», SqlDЫype.NChar)); command.Parameters[«@KemVydan»].Value = textBox9.Text;

command.Parameters.Add(new SqlParameter(«@DataVyd», SqlDЫype.Date)); command.Parameters[«@DataVyd»].Value

vert.ToDateTime(maskedTextBoxЗ.Text);

command.Parameters.Add(new SqlParameter(«@KodPodr», SqlDЫype.NChar)); command.Parameters[«@KodPodr»].Value = textBox17.Text;

command.Parameters.Add(new SqlParameter(«@DataSved», SqlDЫype.Date)); command.Parameters[«@DataSved»].Value

vert.ToDateTime(maskedTextBox4.Text);

if (maskedTextBox7.Selectionlength != 0)

{

command.Parameters.Add(new SqlParameter(«@Srok», SqlDЫype.Date));

Con-

Con-

command.Parameters[«@Srok»].Value vert.ToDateTime(maskedTextBox7.Text);

}

Con-

else

{

command.Parameters.Add(new SqlParameter(«@Srok», SqlDbType.Date)); command.Parameters[«@Srok»].Value = null;

}

taЫe.Rows.Clear(); adapter.Fill(taЫe); slct О;

dgPass.DataSource

int 1].Cells[0].Value.ToString());

int Adr

1].Cells[0].Value.ToString());

tаЫе;

Pass=Convert.Tolnt32(dgPass.Rows[dgPass.RowCount- Convert.Tolnt32(dgAdres.Rows[dgAdres.RowCount

EntityName = «ЛичныеДанные»;

tаЫе = ЗapnлaтaDataSet.TaЬles[EntityName];

command.CommandText = EntityName + «Insert»; adapter.InsertCommand = command;

command.Parameters.Add(new SqlParameter(«@Familia», SqlDbType.Int)); command.Parameters[«@Familia»].Value = textBoxl.Text;

command.Parameters.Add(new SqlParameter(«@Imja», SqlDbType.Int)); command.Parameters[«@Imja»].Value = textBox2.Text;

command.Parameters.Add(new SqlParameter(«@Otchestvo», SqlDbType.NChar)); command.Parameters[«@Otchestvo»].Value = textBoxЗ.Text;

command.Parameters.Add(new SqlParameter(«@Datarog», SqlDbType.Date)); command.Parameters[«@Datarog»].Value

vert.ToDateTime(maskedTextBoxl.Text);

command.Parameters.Add(new SqlParameter(«@KodPodr», SqlDbType.Date));

Con-

command.Parameters[«@KodPodr»].Value = Convert.ToDateTime(textBox17.Text); command.Parameters.Add(new SqlParameter(«@MestoRog», SqlDbType.NChar));

command.Parameters[«@MestoRog»].Value = textBox5.Text; command.Parameters.Add(new SqlParameter(«@INN», SqlDbType.Int));

command.Parameters[«@INN»].Value = textBoxб.Text; command.Parameters.Add(new SqlParameter(«@SNILS», SqlDbType.Int));

command.Parameters[«@SNILS»].Value = maskedTextBox2.Text; command.Parameters.Add(new SqlParameter(«@Pass», SqlDbType.Int));

command.Parameters[«@Pass»].Value = Pass;

command.Parameters.Add(new SqlParameter(«@Adres», SqlDbType.Int)); command.Parameters[«@Adres»].Value = Adr;

command.Parameters.Add(new SqlParameter(«@Grag», SqlDbType.Int)); command.Parameters[«@Grag»].Value = cbGrag.Text;

taЫe.Rows.Clear();

adapter.Fill(taЫe); slct О;

dgTaЬleLD.DataSource = tаЫе;

int Dolg=0;

for(int i=0; i<dgDolg.RowCount-l;i++)

{

if (dgDolg.Rows[i].Cells[l].Value.ToString() == cbDolg.Text)

Dolg = Convert.Tolnt32(dgDolg.Rows[i].Cells[0].Value.ToString());

}

int Podr 0·,

for (int i = 0; i < dgPodr.RowCount - **1;** i++)

{

if (dgPodr.Rows[i].Cells[l].Value.ToString() == cbPodr.Text)

Podr = Convert.Tolnt32(dgPodr.Rows[i].Cells[0].Value.ToString());

}

int LD = 0;

for (int i = 0; i <dgTaЬleLD.RowCount - **1;** i++)

{

if (dgTaЬleLD.Rows[i].Cells[б].Value.ToString() == textBoxб.Text)

LD = Convert.Tolnt32(dgTaЬleLD.Rows[i].Cells[0].Value.ToString());

}

int Razrjad = 0;

for (int i = 0; i < dgRazr.RowCount - **1;** i++)

{

if (dgRazr.Rows[i].Cells[l].Value.ToString() == cbRazrjad.Text)

Razrjad = Convert.Tolnt32(dgRazr.Rows[i].Cells[0].Value.ToString());

}

int VidZan for (int i

{

0·,

0; i < dgVidZan.RowCount - **1;** i++)

if (dgVidZan.Rows[i].Cells[l].Value.ToString() == cbRazrjad.Text)

VidZan = Convert.Tolnt32(dgVidZan.Rows[i].Cells[0].Value.ToString());

}

EntityName = «Сотрудники»;

tаЫе = ЗapnлaтaDataSet.TaЬles[EntityName];

command.CommandText = EntityName + «Insert»; adapter.InsertCommand = command;

command.Parameters.Add(new SqlParameter(«@LD», SqlDЫype.Int)); command.Parameters[«@LD»].Value = LD;

command.Parameters.Add(new SqlParameter(«@Dolg», SqlDЫype.Int)); command.Parameters[«@Dolg»].Value = Dolg;

command.Parameters.Add(new SqlParameter(«@Podr», SqlDЫype.Int)); command.Parameters[«@Podr»].Value = Podr;

command.Parameters.Add(new SqlParameter(«@Datapriema», SqlDЫype.Date)); command.Parameters[«@Datapriema»].Value

vert.ToDateTime(maskedTextBox5.Text);

command.Parameters.Add(new SqlParameter(«@Datauv», SqlDЫype.Date)); command.Parameters[«@Datauv»].Value

vert.ToDateTime(maskedTextBoxб.Text);

command.Parameters.Add(new SqlParameter(«@Razrjad», SqlDЫype.Int)); command.Parameters[«@Razrjad»].Value = Razrjad;

command.Parameters.Add(new SqlParameter(«@VidZan», SqlDЫype.Int)); command.Parameters[«@VidZan»].Value = VidZan;

Con-

Con-

taЫe.Rows.Clear(); adapter.Fill(taЫe); slct О;

dgTaЫe.DataSource tаЫе;

}

private void FormlnsertSotr\_Load(object sender, EventArgs е)

{

slct О;

EntityName

СотрудникиЕntitу;

dgTaЫe.DataSource = tаЫе; EntityName = PaзpядEntity;

slct О;

dgRazr.DataSource = tаЫе;

for (int i = 0; i < dgRazr.RowCount-1; i++)

{

cbRazrjad.Items.Add(dgRazr.Rows[i].Cells[l].Value.ToString().Trim());

}

EntityName ПодразделенияЕntitу; slct();

dgPodr.DataSource = tаЫе;

for (int i = 0; i < dgPodr.RowCount-1; i++)

{

cbPodr.Items.Add(dgPodr.Rows[i].Cells[l].Value.ToString().Trim());

}

EntityName ЛичныеДанныеЕntitу; slct();

dgTaЬleLD.DataSource = tаЫе; EntityName = ДолжностиЕntitу;

slct();

dgDolg.DataSource = tаЫе;

for (int i = 0; i < dgDolg.RowCount-l;i ++)

{

cbDolg.Items.Add(dgDolg.Rows[i].Cells[l].Value.ToString().Trim());

}

EntityName slct();

ВидЗанятостиЕntitу;

dgVidZan.DataSource = tаЫе;

for (int i = 0; i < dgVidZan.RowCount-1; i++)

{

cbVidZan.Items.Add(dgVidZan .Rows[i].Cells[l].Value.ToString().Trim());

}

EntityName ПасnортныеДанныеЕntitу; slct();

dgPass.DataSource tаЫе; EntityName = AдpecEntity; slct();

dgAdres.DataSource = tаЫе; EntityName = СтавкиЕntitу;

slct();

dgRekvizit.DataSource tаЫе;

}

private void cbRazrjad\_SelectedindexChanged(object sender, EventArgs е)

{

for (int i 0; i < dgRazr.RowCount - 1; i++)

{

if (cbRazrjad.Text

{

dgRazr.Rows[i].Cells[l].Value.ToString().Trim())

tbTK.Text dgRazr.Rows[i].Cells[2]. Value.ToString();

tbOklad.Text Convert.ToString(Convert.ToDouЫe(tbTK.Text) \* Con- vert.ToDouЫe(dgRekvizit.Rows[0].Cells[2].Value.ToString().Trim()));

}

}

}

}

}

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms; using System.Data.Sql;

using System.Data.SqlClient;

namespaceЗapnлaтa

{

puЫicpartialclassFormOklad :Form

{

privateconststringPaзpядEntity = «Разряд»; privateconststringCoтpyдникиEntity = «Сотрудники»; privateconststringПoдpaздeлeнияEntity = «Подразделения»; privateconststringBидHaчиcлeнияEntity = «ВидНачисления»; privateconststringHaчиcлeниeЗapnлaтыEntity = «НачислениеЗарnлаты»; privateconststringЛичныeДaнныeEntity = «ЛичныеДанные»; privateconststringCтaвкиEntity = «Ставки»; privateDataSetЗapnлaтaDataSet;

privateSqlDataAdapter adapter; privateSqlConnection connection; privateSqlCommand command; puЫicDataTaЬle tаЫе; privatestring EntityName;

puЫic FormOklad()

{

InitializeComponent(); ЗapnлaтaDataSet = newDataSet();

ЗapnлaтaDataSet.TaЬles.Add(PaзpядEntity); ЗapnлaтaDataSet.TaЬles.Add(CoтpyдникиEntity); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ПодразделенияЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ВидНачисленияЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(НачислениеЗарnлатыЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ЛичныеДанныеЕntitу); ЗapnлaтaDataSet.TaЬles.Add(CтaвкиEntity);

adapter = newSqlDataAdapter(); tаЫе = ЗapnлaтaDataSet.TaЬles[4];

connection = newSqlConnection(Properties.Settings.Default.ЗapnлaтaConnectionString); command = newSqlCommand();

command.Connection = connection; command.CommandType = CommandType.StoredProcedure;

}

puЫicvoid slct() //отображениетаблиц

{

// adapter = new SqlDataAdapter();

tаЫе = ЗapnлaтaDataSet.TaЬles[EntityName]; command.CommandText = EntityName + «Select»;

command.Parameters.Clear(); adapter.SelectCommand = command;

taЫe.Rows.Clear(); adapter.Fill(taЫe);

}

privatevoid FormOklad\_Load(object sender, EventArgs е)

{

EntityName = «НачислениеЗарnлаты»; slct();

dgTaЫe.DataSource = tаЫе; dgTaЫe.Columns[4].VisiЫe = false;

SqlDataAdapter tip = newSqlDataAdapter(«SELECT \* FROM Подразделения», connection); DataTaЬle tЫl = newDataTaЬle();

tip.Fill(tЬll);

comboBoxl.DataSource = tЫl; comboBoxl.DisplayMember = «Наименование»; comboBoxl.ValueMember = «КодПодразделения»; comboBoxl.Selectedindex = **-1;**

SqlDataAdapter sotr = newSqlDataAdapter(«SELECT \* FROM ЛичныеДанные», connection); DataTaЬle tЫ2 = newDataTaЬle();

sotr.Fill(tЫ2);

comboBox2.DataSource = tЫ2; comboBox2.DisplayMember = «Фамилия» comboBox2.ValueMember = «Код»; comboBox2.Selectedindex = **-1;**

}

privatevoid btSchet\_Click(object sender, EventArgs е)

{

intKoд = Convert.Tolnt32(comboBox2.SelectedValue); intPaзpяд=l;

douЬleTC=0; dоuЫеОклад;

EntityName = «Сотрудники»; slct();

dgTaЬle2.DataSource = tаЫе; for(int i=0; i<dgTaЬle2.RowCount-1; i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[l].Value.ToString().Trim())==Koд) Paзpяд=Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[6].Value.ToString().Trim());

}

EntityName = «Разряд»; slct();

dgTaЬle2.DataSource = tаЫе;

for (int i = 0; i < dgTaЬle2.RowCount - **1;** i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[l].Value.ToString().Trim()) Разряд)

те Convert.ToDouЫe(dgTaЬle2.Rows[i].Cells[2].Value.ToString().Trim());

}

EntityName = «Ставки»; slct();

dgTaЬle2.DataSource = tаЫе;

if (Convert.Tolnt32(tbFakt.Text) >= Convert.Tolnt32(tbNorma.Text))

Оклад Convert.ToDouЫe(dgTaЬle2.Rows[0].Cells[2].Value.ToString().Trim()) \* ТС; else

Оклад (Convert.ToDouЫe(dgTaЬle2.Rows[0].Cells[2].Value.ToString().Trim()) \* ТС) /

Convert.ToDouЫe(tbNorma.Text) \* Convert.ToDouЫe(tbFakt.Text); tbSumma.Text = Convert.ToString(Convert.Tolnt32(Oклaд));

}

privatevoid btlnsert\_Click(object sender, EventArgs е)

{

int КодСотрудника=l;

EntityName = «Сотрудники»; slct();

dgTaЬle2.DataSource = tаЫе;

for (int i = 0; i < dgTaЬle2.RowCount - 1; i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[l].Value.ToString().Trim()) == Con­ vert.Tolnt32(comboBox2.SelectedValue))

КодСотрудника = Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[0].Value.ToString().Trim());

}

tаЫе = ЗарnлатаDаtаSеt.ТаЫеs[НачислениеЗарnлатыЕntitу];

command.CommandText = НачислениеЗарnлатыЕntitу + «Insert»; adapter.InsertCommand = command;

command.Parameters.Add(newSqlParameter(«@IdSotr», SqlDbType.Int)); command.Parameters[«@IdSotr»].Value = КодСотрудника; command.Parameters.Add(newSqlParameter(«@IdVidNach», SqlDbType.Int));

command.Parameters[«@IdVidNach»].Value = 5; command.Parameters.Add(newSqlParameter(«@Data», SqlDbType.Date));

command.Parameters[«@Data»].Value = dateTimePickerl.Value; command.Parameters.Add(newSqlParameter(«@Summa», SqlDbType.Float));

command.Parameters[«@Summa»].Value = Convert.ToDouЫe(tbSumma.Text); taЫe.Rows.Clear();

adapter.Fill(taЫe);

EntityName = «НачислениеЗарnлаты»; slct();

dgTaЫe.DataSource = tаЫе; dgTaЫe.Columns[4].VisiЫe = false;

}

}

}

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms;

namespaceЗapnлaтa

{

puЫicpartialclassFormOtchet :Form

{

puЫic FormOtchet()

{

InitializeComponent();

}

privatevoid FormOtchet\_Load(object sender, EventArgs е)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу «Зарпла­ таDаtаSеt.НачислениеЗарплатыSеlесt». При необходимости она может быть перемещена или удалена. this.НачислениеЗарплатыSеlесtТаЫеАdарtеr.Fill(this.ЗарплатаDаtаSеt.НачислениеЗарплаты Select);

this.reportViewerl.RefreshReport();

}

}

}

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms; using System.Data.Sql;

using System.Data.SqlClient;

паmеsрасеЗарплата

{

puЫicpartialclassFormPovremenno :Form

{

privateconststringPaзpядEntity = «Разряд»; privateconststringCoтpyдникиEntity = «Сотрудники»; privateconststringПoдpaздeлeнияEntity = «Подразделения»; privateconststringBидHaчиcлeнияEntity = «ВидНачисления»; privateconststringHaчиcлeниeЗapnлaтыEntity = «НачислениеЗарплаты»; privateconststringЛичныeДaнныeEntity = «ЛичныеДанные»; privateconststringCтaвкиEntity = «Ставки»; privateDataSetЗapnлaтaDataSet;

privateSqlDataAdapter adapter; privateSqlConnection connection; privateSqlCommand command; puЫicDataTaЬle tаЫе; privatestring EntityName;

puЫic FormPovremenno()

{

InitializeComponent(); ЗapnлaтaDataSet = newDataSet();

ЗapnлaтaDataSet.TaЬles.Add(PaзpядEntity); ЗарплатаDаtаSеt.ТаЫеs.Аdd(СотрудникиЕпtitу); ЗарплатаDаtаSеt.ТаЫеs.Аdd(ПодразделенияЕпtitу); ЗарплатаDаtаSеt.ТаЫеs.Аdd(ВидНачисленияЕпtitу); ЗарплатаDаtаSеt.ТаЫеs.Аdd(НачислениеЗарплатыЕпtitу); ЗарплатаDаtаSеt.ТаЫеs.Аdd(ЛичныеДанныеЕпtitу); ЗapnлaтaDataSet.TaЬles.Add(CтaвкиEntity);

adapter = newSqlDataAdapter(); tаЫе = ЗapnлaтaDataSet.TaЬles[4];

connection = newSqlConnection(Properties.Settings.Default.ЗapnлaтaConnectionString); command = newSqlCommand();

command.Connection = connection; command.CommandType = CommandType.StoredProcedure;

}

puЫicvoid slct() //отображениетаблиц

{

// adapter = new SqlDataAdapter();

tаЫе = ЗapnлaтaDataSet.TaЬles[EntityName];

command.CommandText = EntityName + «Select»; command.Parameters.Clear();

adapter.SelectCommand = command; taЫe.Rows.Clear(); adapter.Fill(taЫe);

}

privatevoid FormPovremenno\_Load(object sender, EventArgs е)

{

EntityName = «НачислениеЗарnлаты»; slct();

dgTaЫe.DataSource = tаЫе; dgTaЫe.Columns[4].VisiЫe = false;

SqlDataAdapter tip = newSqlDataAdapter(«SELECT \* FROM Подразделения», connection); DataTaЬle tЫl = newDataTaЬle();

tip.Fill(tЬll);

comboBoxl.DataSource = tЫl; comboBoxl.DisplayMember = «Наименование»; comboBoxl.ValueMember = «КодПодразделения»; comboBoxl.Selectedindex = **-1;**

SqlDataAdapter sotr = newSqlDataAdapter(«SELECT \* FROM ЛичныеДанные», connection); DataTaЬle tЫ2 = newDataTaЬle();

sotr.Fill(tЫ2);

comboBox2.DataSource = tЫ2; comboBox2.DisplayMember = «Фамилия»; comboBox2.ValueMember = «Код»; comboBox2.Selectedindex = **-1;**

}

privatevoid btSchet\_Click(object sender, EventArgs е)

{

intKoд = Convert.Tolnt32(comboBox2.SelectedValue); intРазряд = **1;**

douЬleTC = 0; dоuЫеТариф;

EntityName = «Сотрудники»; slct();

dgTaЬle2.DataSource = tаЫе;

for (int i = 0; i < dgTaЬle2.RowCount - **1;** i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[0].Value.ToString().Trim()) == Код) Разряд= Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[6].Value.ToString().Trim());

}

EntityName = «Разряд»; slct();

dgTaЬle2.DataSource = tаЫе;

for (int i = 0; i < dgTaЬle2.RowCount - **1;** i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[l].Value.ToString().Trim()) Разряд)

те Convert.ToDouЫe(dgTaЬle2.Rows[i].Cells[2].Value.ToString().Trim());

}

EntityName = «Ставки»; slct();

dgTaЬle2.DataSource = tаЫе;

Тариф= Convert.ToDouЫe(dgTaЬle2.Rows[l].Cells[2].Value.ToString().Trim()) \* TC\*Convert.ToDouЫe(textBox2.Text);

tbSumma.Text = Convert.ToString(Convert.Tolnt32(Tapиф));

}

privatevoid btlnsert\_Click(object sender, EventArgs е)

{

int КодСотрудника = 1; EntityName = «Сотрудники»; slct();

dgTaЬle2.DataSource = tаЫе;

for (int i = 0; i < dgTaЬle2.RowCount - 1; i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[l].Value.ToString().Trim()) == Con­ vert.Tolnt32(comboBox2.SelectedValue))

КодСотрудника = Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[0].Value.ToString().Trim());

}

tаЫе = ЗарnлатаDаtаSеt.ТаЫеs[НачислениеЗарnлатыЕntitу]; command.CommandText = НачислениеЗарnлатыЕntitу + «Insert»;

adapter.InsertCommand = command; command.Parameters.Add(newSqlParameter(«@IdSotr», SqlDbType.Int)); command.Parameters[«@IdSotr»].Value = КодСотрудника; command.Parameters.Add(newSqlParameter(«@IdVidNach», SqlDbType.Int));

command.Parameters[«@IdVidNach»].Value = З; command.Parameters.Add(newSqlParameter(«@Data», SqlDbType.Date));

command.Parameters[«@Data»].Value = dateTimePickerl.Value; command.Parameters.Add(newSqlParameter(«@Summa», SqlDbType.Float));

command.Parameters[«@Summa»].Value = Convert.ToDouЫe(tbSumma.Text); taЫe.Rows.Clear();

adapter.Fill(taЫe);

EntityName = «НачислениеЗарnлаты»; slct();

dgTaЫe.DataSource = tаЫе; dgTaЫe.Columns[4].VisiЫe = false;

}

}

}

puЫicpartialclassFormOklad :Form

{

privateconststringPaзpядEntity = «Разряд»; privateconststringCoтpyдникиEntity = «Сотрудники»; privateconststringПoдpaздeлeнияEntity = «Подразделения»; privateconststringBидHaчиcлeнияEntity = «ВидНачисления»; privateconststringHaчиcлeниeЗapnлaтыEntity = «НачислениеЗарnлаты»; privateconststringЛичныeДaнныeEntity = «ЛичныеДанные»; privateconststringCтaвкиEntity = «Ставки»; privateDataSetЗapnлaтaDataSet;

privateSqlDataAdapter adapter; privateSqlConnection connection; privateSqlCommand command; puЫicDataTaЬle tаЫе; privatestring EntityName;

puЫic FormOklad()

{

InitializeComponent(); ЗapnлaтaDataSet = newDataSet();

ЗapnлaтaDataSet.TaЬles.Add(PaзpядEntity); ЗapnлaтaDataSet.TaЬles.Add(CoтpyдникиEntity); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ПодразделенияЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ВидНачисленияЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(НачислениеЗарnлатыЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ЛичныеДанныеЕntitу); ЗapnлaтaDataSet.TaЬles.Add(CтaвкиEntity);

adapter = newSqlDataAdapter();

tаЫе = ЗapnлaтaDataSet.TaЬles[4];

connection = newSqlConnection(Properties.Settings.Default.ЗapnлaтaConnectionString); command = newSqlCommand();

command.Connection = connection; command.CommandType = CommandType.StoredProcedure;

}

puЫicvoid slct() //отображениетаблиц

{

// adapter = new SqlDataAdapter();

tаЫе = ЗapnлaтaDataSet.TaЬles[EntityName]; command.CommandText = EntityName + «Select»;

command.Parameters.Clear(); adapter.SelectCommand = command;

taЫe.Rows.Clear(); adapter.Fill(taЫe);

}

privatevoid FormOklad\_Load(object sender, EventArgs е)

{

EntityName = «НачислениеЗарплаты»; slct();

dgTaЫe.DataSource = tаЫе; dgTaЫe.Columns[4].VisiЫe = false;

SqlDataAdapter tip = newSqlDataAdapter(«SELECT \* FROM Подразделения», connection); DataTaЬle tЫl = newDataTaЬle();

tip.Fill(tЬll);

comboBoxl.DataSource = tЫl; comboBoxl.DisplayMember = «Наименование»; comboBoxl.ValueMember = «КодПодразделения»; comboBoxl.Selectedindex = **-1;**

SqlDataAdapter sotr = newSqlDataAdapter(«SELECT \* FROM ЛичныеДанные», connection); DataTaЬle tЫ2 = newDataTaЬle();

sotr.Fill(tЫ2);

comboBox2.DataSource = tЫ2; comboBox2.DisplayMember = «Фамилия» comboBox2.ValueMember = «Код»; comboBox2.Selectedindex = **-1;**

}

privatevoid btSchet\_Click(object sender, EventArgs е)

{

intKoд = Convert.Tolnt32(comboBox2.SelectedValue); iпtРазряд=l;

douЬleTC=0; dоuЫеОклад;

EntityName = «Сотрудники»; slct();

dgTaЬle2.DataSource = tаЫе; for(int i=0; i<dgTaЬle2.RowCount-1; i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[l].Value.ToString().Trim())==Koд) Paзpяд=Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[6].Value.ToString().Trim());

}

EntityName = «Разряд»; slct();

dgTaЬle2.DataSource = tаЫе;

for (int i = 0; i < dgTaЬle2.RowCount - **1;** i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[l].Value.ToString().Trim()) Разряд)

те Convert.ToDouЫe(dgTaЬle2.Rows[i].Cells[2].Value.ToString().Trim());

}

EntityName = «Ставки»; slct();

dgTaЬle2.DataSource = tаЫе;

if (Convert.Tolnt32(tbFakt.Text) >= Convert.Tolnt32(tbNorma.Text))

Оклад Convert.ToDouЫe(dgTaЬle2.Rows[0].Cells[2].Value.ToString().Trim()) \* ТС; else

Оклад (Convert.ToDouЫe(dgTaЬle2.Rows[0].Cells[2].Value.ToString().Trim()) \* ТС) / Convert.ToDouЫe(tbNorma.Text) \* Convert.ToDouЫe(tbFakt.Text);

tbSumma.Text = Convert.ToString(Convert.Tolnt32(Oклaд));

}

privatevoid btlnsert\_Click(object sender, EventArgs е)

{

int КодСотрудника=l;

EntityName = «Сотрудники»; slct();

dgTaЬle2.DataSource = tаЫе;

for (int i = 0; i < dgTaЬle2.RowCount - 1; i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[l].Value.ToString().Trim()) == Con­ vert.Tolnt32(comboBox2.SelectedValue))

КодСотрудника = Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[0].Value.ToString().Trim());

}

tаЫе = ЗарnлатаDаtаSеt.ТаЫеs[НачислениеЗарnлатыЕntitу];

command.CommandText = НачислениеЗарnлатыЕntitу + «Insert»; adapter.InsertCommand = command;

command.Parameters.Add(newSqlParameter(«@IdSotr», SqlDbType.Int)); command.Parameters[«@IdSotr»].Value = КодСотрудника; command.Parameters.Add(newSqlParameter(«@IdVidNach», SqlDbType.Int));

command.Parameters[«@IdVidNach»].Value = 5; command.Parameters.Add(newSqlParameter(«@Data», SqlDbType.Date));

command.Parameters[«@Data»].Value = dateTimePickerl.Value; command.Parameters.Add(newSqlParameter(«@Summa», SqlDbType.Float));

command.Parameters[«@Summa»].Value = Convert.ToDouЫe(tbSumma.Text); taЫe.Rows.Clear();

adapter.Fill(taЫe);

EntityName = «НачислениеЗарnлаты»; slct();

dgTaЫe.DataSource = tаЫе; dgTaЫe.Columns[4].VisiЫe = false;

}

}

}

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms;

namespaceЗapnлaтa

{

puЫicpartialclassFormOtchet :Form

{

puЫic FormOtchet()

{

InitializeComponent();

}

privatevoid FormOtchet\_Load(object sender, EventArgs е)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу «Зарnла­ таDаtаSеt.НачислениеЗарnлатыSеlесt». При необходимости она может быть перемещена или удалена. this.HaчиcлeниeЗapnлaтыSelectTaЬleAdapter.Fill(this.ЗapnлaтaDataSet.HaчиcлeниeЗapnлaты Select);

this.reportViewerl.RefreshReport();

}

}

}

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms; using System.Data.Sql;

using System.Data.SqlClient;

namespaceЗapnлaтa

{

puЫicpartialclassFormPovremenno :Form

{

privateconststringPaзpядEntity = «Разряд»; privateconststringCoтpyдникиEntity = «Сотрудники»; privateconststringПoдpaздeлeнияEntity = «Подразделения»; privateconststringBидHaчиcлeнияEntity = «ВидНачисления»; privateconststringHaчиcлeниeЗapnлaтыEntity = «НачислениеЗарnлаты»; privateconststringЛичныeДaнныeEntity = «ЛичныеДанные»; privateconststringCтaвкиEntity = «Ставки»; privateDataSetЗapnлaтaDataSet;

privateSqlDataAdapter adapter; privateSqlConnection connection; privateSqlCommand command; puЫicDataTaЬle tаЫе; privatestring EntityName;

puЫic FormPovremenno()

{

InitializeComponent(); ЗapnлaтaDataSet = newDataSet();

ЗapnлaтaDataSet.TaЬles.Add(PaзpядEntity); ЗapnлaтaDataSet.TaЬles.Add(CoтpyдникиEntity); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ПодразделенияЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ВидНачисленияЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(НачислениеЗарnлатыЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ЛичныеДанныеЕntitу); ЗapnлaтaDataSet.TaЬles.Add(CтaвкиEntity);

adapter = newSqlDataAdapter(); tаЫе = ЗapnлaтaDataSet.TaЬles[4];

connection = newSqlConnection(Properties.Settings.Default.ЗapnлaтaConnectionString); command = newSqlCommand();

command.Connection = connection; command.CommandType = CommandType.StoredProcedure;

}

puЫicvoid slct() //отображениетаблиц

{

// adapter = new SqlDataAdapter();

tаЫе = ЗapnлaтaDataSet.TaЬles[EntityName]; command.CommandText = EntityName + «Select»;

command.Parameters.Clear(); adapter.SelectCommand = command;

taЫe.Rows.Clear(); adapter.Fill(taЫe);

}

privatevoid FormPovremenno\_Load(object sender, EventArgs е)

{

EntityName = «НачислениеЗарnлаты»; slct();

dgTaЫe.DataSource = tаЫе; dgTaЫe.Columns[4].VisiЫe = false;

SqlDataAdapter tip = newSqlDataAdapter(«SELECT \* FROM Подразделения», connection); DataTaЬle tЫl = newDataTaЬle();

tip.Fill(tЬll);

comboBoxl.DataSource = tЫl; comboBoxl.DisplayMember = «Наименование»; comboBoxl.ValueMember = «КодПодразделения»; comboBoxl.Selectedindex = **-1;**

SqlDataAdapter sotr = newSqlDataAdapter(«SELECT \* FROM ЛичныеДанные», connection); DataTaЬle tЫ2 = newDataTaЬle();

sotr.Fill(tЫ2);

comboBox2.DataSource = tЫ2; comboBox2.DisplayMember = «Фамилия»; comboBox2.ValueMember = «Код»; comboBox2.Selectedindex = **-1;**

}

privatevoid btSchet\_Click(object sender, EventArgs е)

{

intKoд = Convert.Tolnt32(comboBox2.SelectedValue); intРазряд = **1;**

douЬleTC = 0; dоuЫеТариф;

EntityName = «Сотрудники»; slct();

dgTaЬle2.DataSource = tаЫе;

for (int i = 0; i < dgTaЬle2.RowCount - **1;** i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[0].Value.ToString().Trim()) == Код) Разряд= Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[6].Value.ToString().Trim());

}

EntityName = «Разряд»; slct();

dgTaЬle2.DataSource = tаЫе;

for (int i = 0; i < dgTaЬle2.RowCount - **1;** i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[l].Value.ToString().Trim()) Разряд)

те Convert.ToDouЫe(dgTaЬle2.Rows[i].Cells[2].Value.ToString().Trim());

}

EntityName = «Ставки»; slct();

dgTaЬle2.DataSource = tаЫе;

Тариф= Convert.ToDouЫe(dgTaЬle2.Rows[l].Cells[2].Value.ToString().Trim()) \* TC\*Convert.ToDouЫe(textBox2.Text);

tbSumma.Text = Convert.ToString(Convert.Tolnt32(Tapиф));

}

privatevoid btlnsert\_Click(object sender, EventArgs е)

{

int КодСотрудника = 1; EntityName = «Сотрудники»; slct();

dgTaЬle2.DataSource = tаЫе;

for (int i = 0; i < dgTaЬle2.RowCount - 1; i++)

{

if (Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[l].Value.ToString().Trim()) == Con­ vert.Tolnt32(comboBox2.SelectedValue))

КодСотрудника = Convert.Tolnt32(dgTaЬle2.Rows[i].Cells[0].Value.ToString().Trim());

}

tаЫе = ЗарnлатаDаtаSеt.ТаЫеs[НачислениеЗарnлатыЕntitу]; command.CommandText = НачислениеЗарnлатыЕntitу + «Insert»;

adapter.InsertCommand = command; command.Parameters.Add(newSqlParameter(«@IdSotr», SqlDbType.Int)); command.Parameters[«@IdSotr»].Value = КодСотрудника; command.Parameters.Add(newSqlParameter(«@IdVidNach», SqlDbType.Int));

command.Parameters[«@IdVidNach»].Value = З; command.Parameters.Add(newSqlParameter(«@Data», SqlDbType.Date));

command.Parameters[«@Data»].Value = dateTimePickerl.Value; command.Parameters.Add(newSqlParameter(«@Summa», SqlDbType.Float));

command.Parameters[«@Summa»].Value = Convert.ToDouЫe(tbSumma.Text); taЫe.Rows.Clear();

adapter.Fill(taЫe);

EntityName = «НачислениеЗарnлаты»; slct();

dgTaЫe.DataSource = tаЫе; dgTaЫe.Columns[4].VisiЫe = false;

}

}

}

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms; using System.Data.Sql;

using System.Data.SqlClient; namespaceЗapnлaтa

{

puЫicpartialclassFormRazrjad :Form

{

privateconststringPaзpядEntity = «Разряд»; privateDataSetЗapnлaтaDataSet; privateSqlDataAdapter adapter; privateSqlConnection connection; privateSqlCommand command;

puЫicDataTaЬle tаЫе; puЫic FormRazrjad()

{

InitializeComponent(); ЗapnлaтaDataSet = newDataSet();

ЗapnлaтaDataSet.TaЬles.Add(PaзpядEntity); adapter = newSqlDataAdapter();

tаЫе = ЗapnлaтaDataSet.TaЬles[0];

connection = newSqlConnection(Properties.Settings.Default.ЗapnлaтaConnectionString); command = newSqlCommand();

command.Connection = connection; command.CommandType = CommandType.StoredProcedure;

}

privatevoid FormRazrjad\_Load(object sender, EventArgs е)

{

slct();

}

puЫicvoid slct() //отображениетаблиц

{

// adapter = new SqlDataAdapter();

// tаЫе = ЗapnлaтaDataSet.TaЬles[0];

command.CommandText = PaзpядEntity + «Select»; command.Parameters.Clear();

adapter.SelectCommand = command; taЫe.Rows.Clear(); adapter.Fill(taЫe);

dgTaЫe.DataSource tаЫе;

}

privatevoid btUpdate\_Click(object sender, EventArgs е)

{

command.CommandText = «РазрядUрdаtе»; adapter.UpdateCommand = command;

command.Parameters.Add(newSqlParameter(«@IdRazrjad»,SqlDbType.Int)); command.Parameters[«@IdRazrjad»].Value = Con­

vert.Tolnt32(dgTaЫe.CurrentRow.Cells[0].Value.ToString()); command.Parameters.Add(newSqlParameter(«@Razrjad», SqlDbType.Int));

command.Parameters[«@Razrjad»].Value = Con- vert.Tolnt32(dgTaЫe.CurrentRow.Cells[l].Value.ToString()); command.Parameters.Add(newSqlParameter(«@TK», SqlDbType.Float));

command.Parameters[«@TK»].Value = Con­ vert.ToDouЫe(dgTaЫe.CurrentRow.Cells[2].Value.ToString());

taЫe.Rows.Clear(); adapter.Fill(taЫe); slct();

}

}

}

Форма «Сотрудники» usingSystem;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms;

using System.Data.Sql;

using System.Data.SqlClient;

namespaceЗapnлaтa

{

puЫicpartialclassFormSotr :Form

{

privateconststringCoтpyдникиEntity = «Сотрудники»; privateconststringЛичныeДaнныeEntity = «ЛичныеДанные»; privateconststringПoдpaздeлeнияEntity = «Подразделения»; privateconststringPaзpядEntity = «Разряд»; privateconststringДoлжнocтиEntity = «Должности»; privateconststringBидЗaнятocтиEntity = «ВидЗанятости»; privateDataSetЗapnлaтaDataSet;

privateSqlDataAdapter adapter; privateSqlConnection connection; privateSqlCommand command; puЫicDataTaЬle tаЫе; privatestring EntityName;

puЫic FormSotr()

{

InitializeComponent(); ЗapnлaтaDataSet = newDataSet();

ЗapnлaтaDataSet.TaЬles.Add(PaзpядEntity); ЗapnлaтaDataSet.TaЬles.Add(CoтpyдникиEntity); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ПодразделенияЕntitу); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ЛичныеДанныеЕntitу); ЗapnлaтaDataSet.TaЬles.Add(ДoлжнocтиEntity); ЗарnлатаDаtаSеt.ТаЫеs.Аdd(ВидЗанятостиЕntitу); adapter = newSqlDataAdapter();

tаЫе = ЗapnлaтaDataSet.TaЬles[l];

connection = newSqlConnection(Properties.Settings.Default.ЗapnлaтaConnectionString); command = newSqlCommand();

command.Connection = connection; command.CommandType = CommandType.StoredProcedure;

}

puЫicvoid slct() //отображениетаблиц

{

// adapter = new SqlDataAdapter();

tаЫе = ЗapnлaтaDataSet.TaЬles[EntityName]; command.CommandText = EntityName + «Select»;

command.Parameters.Clear(); adapter.SelectCommand = command;

taЫe.Rows.Clear(); adapter.Fill(taЫe);

}

privatevoid FormSotr\_Load(object sender, EventArgs е)

{

EntityName = СотрудникиЕntitу; slct();

dgTaЫe.DataSource = tаЫе; dgTaЫe.Columns[4].VisiЫe = false; dgTaЫe.Columns[б].VisiЫe = false; dgTaЫe.Columns[10].VisiЫe = false; dgTaЫe.Columns[12].VisiЫe = false; dgTaЫe.Columns[5].HeaderText = «Подрзделение»; dgTaЫe.Columns[lЗ].HeaderText = «Видзанятоти»; for(int i=0; i<dgTaЫe.ColumnCount;i++)

{

cbSearch.Items.Add(dgTaЫe.Columns[i].HeaderText);

}

}

privatevoid bu\_Click(object sender, EventArgs е)

{

BindingSource bs = newBindingSource(); bs.DataSource = dgTaЫe.DataSource;

bs.Filter = cbSearch.Text + « Like '« + tbSearch.Text + «%'«; dgTaЫe.DataSource = bs;

}

privatevoid button5\_Click(object sender, EventArgs е)

{

BindingSource bs = newBindingSource(); bs.DataSource = dgTaЫe.DataSource; bs.Filter = ««;

dgTaЫe.DataSource = bs;

}

privatevoid buttonl\_Click(object sender, EventArgs е)

{

}

privatevoid btlnsert\_Click(object sender, EventArgs е)

{

FormlnsertSotr fIS = newFormlnsertSotr(); fIS.ShowDialog();

EntityName = СотрудникиЕntitу; slct();

dgTaЫe.DataSource = tаЫе; dgTaЫe.Columns[4].VisiЫe = false; dgTaЫe.Columns[б].VisiЫe = false; dgTaЫe.Columns[10].VisiЫe = false; dgTaЫe.Columns[12].VisiЫe = false; dgTaЫe.Columns[5].HeaderText = «Подрзделение»; dgTaЫe.Columns[lЗ].HeaderText = «Видзанятоти»;

}

}

}

Хранимые процедуры для работы с базой данных CREATEPROCEDURE [dbo].[Auth]

(@login nvarchar(50)=NULL, @pass nvarchar(256)=NULL)

AS

BEGIN SETNOCOUNTON

INSЕRТINТОАутентификация( UserName,

PasswordHash) VALUES(

@login, @pass) RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[AuthSelect] @login nvarchar(50)=NULL

AS

BEGIN SETNOCOUNTON

SELECT

Аутентификация.UsеrNаmе, Аутентификация.РаsswоrdНаsh FRОМАутентификация

WHERE (@login isnullorAyтeнтификaция.UserName=@login) END

CREATEPROCEDURE [dbo].[AuthSelect] @login nvarchar(50)=NULL

AS

BEGIN SETNOCOUNTON SELECT

Аутентификация.UsеrNаmе, Аутентификация.РаsswоrdНаsh FRОМАутентификация

WHERE (@login isnullorAyтeнтификaция.UserName=@login) END

CREATEPROCEDURE [dbo].[AuthSelect] @login nvarchar(50)=NULL

AS

BEGIN SETNOCOUNTON SELECT

Аутентификация.UsеrNаmе, Аутентификация.РаsswоrdНаsh FRОМАутентификация

WHERE (@login isnullorAyтeнтификaция.UserName=@login) END

CREATEPROCEDURE [dbo].[AuthSelect] @login nvarchar(50)=NULL

AS

BEGIN SETNOCOUNTON SELECT

Аутентификация.UsеrNаmе, Аутентификация.РаsswоrdНаsh FRОМАутентификация

WHERE (@login isnullorAyтeнтификaция.UserName=@login) END

CREATEPROCEDURE [dbo].[ВидНачисленияDеlеtе] (@IdVidNach int=NULL)

AS BEGIN

SETNOCOUNTON

DЕLЕТЕFRОМВидНачисления WhereKoдBидaHaчиcлeния=@IdVidNach RETURN@@ERROR

END

REATEPROCEDURE [dЬо].[ВидНачисленияlnsеrt]

(

@Name nchar(50)=NULL)

AS

BEGIN SETNOCOUNTON

INSЕRПNТОВидНачисления,: Наименование)

VALUES(

@Name ) RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[ВидНачисленияSеlесt] (@IDVN int=NULL)

AS

BEGIN SETNOCOUNTON SELECT

КодВидаНачисления, Наименование

FROM ВидНачисления

WHERE (@IDVN isnullor КодВидаНачисления=@IDVN) END

CREATEPROCEDURE [dbo].[ВидНачисленияUрdаtе] (@IdVN int=NULL,

@Name nchar(50)=NULL) AS

BEGIN SETNOCOUNTON

UРDАТЕВидНачисления SET

Наименование=@Nаmе WhеrеКодВидаНачисления=@IdVN RETURN@@ERROR

END

CREATEPROCEDURE [dЬо].[ВидУдержанияDеlеtе] (@IdVidUd int=NULL)

AS BEGIN

SETNOCOUNTON

DЕLЕТЕFRОМВидУдержания WhereKoдBидaYдepжaния=@IdVidUd RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[ВидУдержанияlnsеrt] (@IdVU int=NULL,

@Name nchar(50)=NULL)

AS

BEGIN SETNOCOUNTON

INSЕRТINТОВидУдержания( ВидУдержания)

VALUES(

@Name ) RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[ВидУдержанияSеlесt] (@IDVU int=NULL)

AS

BEGIN SETNOCOUNTON SELECT

КодВидаУдержания, ВидУдержания

FROM ВидУдержания

WHERE (@IDVU isnullor КодВидаУдержания=@IDVU) END

CREATEPROCEDURE [dbo].[ВидУдержанияUрdаtе] (@IdVU int=NULL,

@Name nchar(50)=NULL) AS

BEGIN SETNOCOUNTON

UРDАТЕВидУдержания SET

ВидУдержания=@Nаmе WhеrеКодВидаУдержания=@IdVU RETURN@@ERROR

END

CREATEPROCEDURE [dЬо].[ДолжностиDеlеtе] (@IdDolgnost int=NULL)

AS BEGIN

SETNOCOUNTON

DЕLЕТЕFRОМДолжности WhereKoдДoлжнocти=@IdDolgnost RETURN@@ERROR

END

CREATEPROCEDURE [dЬо].[Должностиlnsеrt]

(

@Name nchar(30)=NULL)

AS

BEGIN SETNOCOUNTON

INSЕRТINТОДолжности( Должность)

VALUES(

@Name) RETURN@@ERROR

END

CREATEPROCEDURE [dЬо].[ДолжностиSеlесt] (@Koдint=NULL)

AS

BEGIN SETNOCOUNTON SELECT

Должности.КодЦолжности as КодЦолжности, Должности.Должность

FrоmДолжности Whеrе(@КодisnullоrДолжности.КодЦолжности=@Код)

END

CREATEPROCEDURE [dbo].[ДолжностиUрdаtе] (@IdDolg int=NULL,

@Name nchar(50)=NULL) AS

BEGIN SETNOCOUNTON

UРDАТЕДолжности SET

Должность=@Nаmе WhereKoдЦoлжнocти=@IdDolg RETURN@@ERROR

END

CREATEPROCEDURE [dЬо].[ЛичныеДанныеlnsеrt]

(

@Familia nchar(30)=NULL, @Imja nchar(30)=NULL, @Otchestvo nchar(30)=NULL, @Datarog date=NULL, @MestoRog nchar(50)=NULL, @INN int=NULL,

@SNILS nchar(15)=NULL,

@Pass int=NULL, @Adres int=NULL, @Grag nchar(50))

AS

BEGIN SETNOCOUNTON

INSЕRПNТОЛичныеДанные( Фамилия,

Имя, Отчество, ДатаРождения, МестоРождения, ИНН,

### снилс,

Паспорт, Адрес, Гражданство) VALUES(

@Familia, @Imja, @Otchestvo, @Datarog, @MestoRog, @INN, @SNILS,

@Pass, @Adres, @Grag) RETURN@@ERROR

END

CREATEPROCEDURE [dЬо].[ЛичныеДанныеSеlесt] (@IdLD int=NULL)

AS

BEGIN SETNOCOUNTON SELECT

Код, Фамилия, Отчество, ДатаРождения,

МестоРождения, ИНН,

### снилс,

Паспорт, Адрес, Гражданство

FROM ЛичныеДанные WHERE(@IdLDisnullorKoд=@IdLD)

END

CREATEPROCEDURE [dЬо].[НачислениеЗарnлатыlnsеrt]

(

@IdSotr int=NULL, @IdVidNach int=NULL, @Data date=NULL, @Summa float=NULL)

AS

BEGIN SETNOCOUNTON

INSЕRПNТОНачислениеЗарnлаты( КодСотрудника, КодВидаНачисления, ДатаНачисления,

Сумма) VALUES(

@IdSotr, @IdVidNach, @Data, @Summa

)

RETURN@@ERROR

END

CREATEPROCEDURE [dЬо].[НачислениеЗарnлатыSеlесt] @IdSotr int=NULL,

@IdVidNach int=NULL

AS BEGIN

SETNOCOUNTON

SELECT

Сотрудники.КодСотрудника, ЛичныеДанные.Фамилия, ЛичныеДанные.Имя, ЛичныеДанные.Отчество, ВидНачисления.КодВидаНачисления, ВидНачисления.Наименование, НачислениеЗарnлаты.ДатаНачисления, НачислениеЗарnлаты.Сумма

FROM НачислениеЗарплаты

LEFTOUTERJOIN Сотрудники ON Сотрудни­ ки.КодСотрудника=НачислениеЗарплаты.КодСотрудника

LEFTOUTERJOIN ВидНачисления ON ВидНачисле- ния.КодВидаНачисления=НачислениеЗарплаты.КодВидаНачисления

LEFТOUTERJOIN ЛичныеДанные ON ЛичныеДанные.Код=Сотрудники.ЛичныеДанные WHERE(@IdSotr ISNULLОRНачислениеЗарплаты.КодСотрудника=@IdSоtr AND @IdVidNachISNULLORHaчиcлeниeЗapnлaты.KoдBидaHaчиcлeния=@IdVidNach) END

CREATEPROCEDURE [dЬо].[ПапортныеДанныеlпsеrt]

(

@Seria int=NULL, @Nomer int=NULL,

@KemVydan nchar(50)=NULL, @DataVyd date=NULL, @KodPodr nchar(10)=NULL, @DataSved date=NULL, @Srok date=NULL)

AS

BEGIN SETNOCOUNTON

INSЕRПNТОПаспортныеДанные( Серия,

Нмер, КемВыдан, ДатаВыдачи,

КодПодразделения, ДатаСведений, СрокДействия) VALUES(

@Seria, @Nomer, @KemVydan, @DataVyd, @KodPodr, @DataSved, @Srok) RETURN@@ERROR

END

CREATEPROCEDURE [dЬо].[ПаспортныеДанныеSеlесt] (@IdPass int=NULL)

AS

BEGIN SETNOCOUNTON SELECT

КодПаспорта, Серия, Нмер, КемВыдан, ДатаВыдачи,

КодПодразделения, ДатаСведений, СрокДействия

FROM ПаспортныеДанные WHERE(@IdPassisnullorKoдПacnopтa=@IdPass)

END

CREATEPROCEDURE [dЬо].[ПодразделенияDеlеtе]

(@IdPodrazd int=NULL) AS

BEGIN SETNOCOUNTON

DЕLЕТЕFRОМПодразделения WhеrеКодПодразделения=@IdРоdrаzd RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[Подразделенияlnsеrt] (@IdPodr int=NULL,

@Name nchar(50)=NULL)

AS

BEGIN SETNOCOUNTON

INSЕRПNТОПодразделения,: Наименование)

VALUES(

@Name ) RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[ПодразделенияSеlесt] (@IDPodr int=NULL)

AS

BEGIN SETNOCOUNTON SELECT

КодПодразделения, Наименование

FROM Подразделения

WHERE (@IDPodr isnullor КодПодразделения=@IDРоdr) END

CREATEPROCEDURE [dbo].[ПодразделенияUрdаtе] (@IdPodrazd int=NULL,

@Name nchar(50)=NULL) AS

BEGIN SETNOCOUNTON

UРDАТЕПодразделения SET

Наименование=@Nаmе WhеrеКодПодразделения=@IdРоdrаzd RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[РазрядDеlеtе] (@IdRazrjad int=NULL)

AS BEGIN

SETNOCOUNTON

DELETEFROMPaзpяд WhereKoд=@IdRazrjad RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[Paзpядlnsert] (@IdRazrjad int=NULL, @Razrjad int=NULL,

@ТК float=NULL)

AS

BEGIN SETNOCOUNTON INSERTINTOPaзpяд(

Разряд,ТаифныйКоэффициент) VALUES(

@Razrjad,@TK ) RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[РазрядSеlесt] (@IdRazr int=NULL)

AS

BEGIN SETNOCOUNTON SELECT

Код, Разряд,

ТаифныйКоэффициент FRОМРазряд

WHERE (@IdRazr isnullorKoд=@IdRazr) END

CREATEPROCEDURE [dbo].[РазрядUрdаtе] (@IdRazrjad int=NULL, @Razrjad int=NULL,

@ТК float=NULL) AS

BEGIN SETNOCOUNTON

UРDАТЕРазряд SET

Paзpяд=@Razrjad, ТаифныйКоэффициент=@ТК WhereKoд=@IdRazrjad RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[Сотрудникиlnsеrt]

(

@LD int=NULL, @Dolg int=NULL, @Podr int=NULL,

@Datapriema date=NULL, @Datauv date=NULL, @Razrjad int=NULL, @VidZan int=NULL

)

AS

BEGIN SETNOCOUNTON

INSЕRТINТОСотрудники( ЛичныеДанные, Должность, Подразделение, ДатаПриема,

ДатаУвольнения, Разряд, ВидЗанятости) VALUES(

@LD,

@Dolg, @Podr, @Datapriema, @Datauv, @Razrjad, @VidZan) RETURN@@ERROR

END

CREATEPROCEDURE [dbo].[СотрудникиSеlесt] (@IdSotr int=NULL,

@IdLD int=NULL, @IdDolg int=NULL, @IdPodr int=NULL, @IdDataPriem date=NULL, @DataUv date=NULL, @IdRazr int=NULL, @IdVidZan int=NULL)

AS

BEGIN SETNOCOUNTON SELECT

Сотрудники.КодСотрудника, ЛичныеДанные.Фамилия, ЛичныеДанные.Имя, ЛичныеДанные.Отчество, Подразделения.КодПодразделения, Подразделения.Наименование, Должности.КодДолжности, Должности.Должность, Сотрудники.ДатаПриема, Сотрудники.ДатаУвольнения, Разряд.Код,

Разряд.Разряд, ВидЗанятости.КодЗанятости, ВидЗанятости.Наименование

From Сотрудники

LEFTOUTERJOIN Подразделения ON Подразделе-

ния.КодПодразделения=Сотрудники.Подразделение

LEFTOUTERJOIN Должности ON Сотрудники.Должность:Должности.КодДолжности LEFТOUTERJOIN ЛичныеДанные ON ЛичныеДанные.Код=Сотрудники.ЛичныеДанные LEFTOUTERJOIN Разряд ON Разряд.Код=Сотрудники.Разряд

LEFTOUTERJOIN ВидЗанятости ON ВидЗанятости.КодЗанятости=Сотрудники.ВидЗанятости WHERE(@IdSotr ISNULLОRСотрудники.КодСотрудника=@IdSоtr AND @IdLDISNULLORCoтpyдники.ЛичныeДaнныe=@IdLDAND

@IdDolg ISNULLOR Сотрудники.Должность= @IdDolg AND @IdPodr ISNULLOR Сотрудники.Подразделение= @IdPodr AND

@IdDataPriem ISNULLOR Сотрудники.ДатаПриема= @IdDataPriem AND @DataUv ISNULLOR Сотрудники.ДатаУвольнения= @DataUv AND @IdRazrISNULLORCoтpyдники.Paзpяд= @IdRazrAND

@IdVidZan ISNULLOR Сотрудники.ВидЗанятости= @IdVidZan)

END

CREATEPROCEDURE [dbo].[СтавкиSеlесt]

(@Koдint=NULL)

AS

BEGIN SETNOCOUNTON SELECT

Ставки.КодаsКод, Ставки.Наименование, Ставки.Размер

FrоmСтавки Where(@KoдisnullorCтaвки.Koд=@Koд)

END