

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»  
Заведующий кафедрой \_\_\_\_\_

(Ф.И.О., ученая степень, звание)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Автоматизированное рабочее место для  
сотрудника больницы города Алматы

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Каримбай М. М. БТ 10-4  
(Фамилия и инициалы) группа

Научный руководитель Бишургаев С. Б. профессор г.ф.-м.н.  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бреснева З.Д., с.н.и. преподаватель  
(Фамилия и инициалы, ученая степень, звание)  
Бреснева « 13 » 05 2014 г.  
(подпись)

по безопасности жизнедеятельности:

Тришодовская Н.Г., Д.Х.Н., профессор  
(Фамилия и инициалы, ученая степень, звание)  
Тришод « 04 » 05 2014 г.  
(подпись)

по применению вычислительной техники:

Бишургаев С. Б. г.ф.-м.н. профессор  
(Фамилия и инициалы, ученая степень, звание)  
Бишургаев « 29 » мая 2014 г.  
(подпись)

\_\_\_\_\_  
(Фамилия и инициалы, ученая степень, звание)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

(подпись)

Нормоконтролер: Тусупов Д.М.  
(Фамилия и инициалы, ученая степень, звание)

Тусупов « 28 » мая 2014 г.  
(подпись)

Рецензент: \_\_\_\_\_  
(Фамилия и инициалы, ученая степень, звание)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Факультет информационных технологий  
Специальность Вычислительная техника и программное обеспечение  
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Сарсенбай Максат Шакибайұлы  
(фамилия, имя, отчество)  
Тема проекта Автоматизированное рабочее место  
сотрудника больницы города Алматы

утверждена приказом ректора № 115 от «24» сентября 2013 г.

Срок сдачи законченной работы «30» сеп 2014 г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

АРМ сотрудника больницы города Алматы с  
использованием Oracle database и MS visual studio 2010

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

При разработке данного дипломного проекта были  
учтены следующие требования: простота в использовании,  
привлекательный внешний вид, надежность

Перечень графического материала (с точным указанием обязательных чертежей)

Смоделирование Form и MS Visual Studio 2010 и хранения функций процедур, пакетов, выполняющих взаимодействие данных, приложения, взаимодействие с Team Foundation для управления работами

Рекомендуемая основная литература

1. Адам Ф. iQuery для профессионалов
2. Таммек Д. MS SQL 2008, карманной справочник.
3. Ред П., Корона К. Системы баз данных
4. Ютерев Д., Костарев В. PHP
5. Кузнецов М., Сидяков И. PHP. Практика создания Web-сайтов

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Эксперимент	Зрацкая З.Д.	11.04-13.08.14	Зрацкая
БЖД	Бухаров И.Г.	11.04-04.05.14	Бухаров
Нормоконтроль Научное руководство	Трунов Д.М., Башурдаев С.Б.	28.05.14 29.05.14	Трунов Башурдаев

**Г Р А Ф И К**  
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	Сбор материала и обработка информации	1.04.14	
2	Формирование задач и разработка автоматизированной рабочей среды	10.04.14	
3	Разработка автоматизированной среды	13.04.14	
4	Корректировка и отладка автоматизированной рабочей среды для сотрудника больницы города Великого Новгорода	30.04.14	
5	Расчет экономической части	10.09.14	
6	Работа над разделом БМД	16.05.14	
7	подготовка графической части	18.05.14	
8	Работа над пояснительной запиской	20.05.14	

Дата выдачи задания « 10 » марта 20 14 г.

Заведующий кафедрой \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

Руководитель Билимурев С.Б. \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент Васильев \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

## **Аңдатпа**

Осы дипломдық жұмыста, барлық қойылған мақсаттар орындалған. Дипломның тақырыбы Алматы қаласының автоматтандырылған жұмыс орны. Oracle Database Express Edition 11g және ООП Microsoft visual studio C# технологиялар қолданылған.

Аурухананың қызметкерінің жұмысын тездейтін жаса жоба құрылды, ол емделушінің үлкен санының қаратуын көбейтеді.

Жұмыста тағыда қара-тіршілік әрекетімнің қауіпсіздігінің шаралары есептелінген, және техникo-экономикалық қисын табылды.

Сайып келгенде жиынтықта еңсер жұмыс, және экономикалық тиімділік енгізу айтылды жүйе.

## **Аннотация**

В данной дипломной работе согласно всем поставленным задачам было разработано и создано автоматизированное рабочее место сотрудника больницы города Алматы, на базе продуктов Oracle Database Express Edition 11g Release 2, и языка объектно-ориентированного программирования Microsoft visual studio C#.

В работе была создана система, ускоряющая работу сотрудников больниц, что позволяет увеличить осмотр большего количества пациентов.

В работе также были рассмотрены меры безопасности жизнедеятельности, а также технико-экономическое обоснование.

В заключении подводятся итоги проделанной работы, и экономическая эффективность внедрения данной системы.

## **Annotation**

According to all problems to solve, in this graduated work the computer workstation of hospital staff in Almaty region was prepared and produced. The program is based on Oracle Database Express Edition 11g Release 2 and object-oriented language program called Java.

In this work the new system was created in order to speed up the working process among the hospital staff which allows them to increase the inspection of more patients.

Also, the paper introduced the life safety measures and project feasibility study. In conclusion, the paper shows summary of the work and financial viability of the project implementation.

## Оглавление

Введение.....	14
1 Автоматизированное рабочее место .....	16
1.1 Понятие АРМ.....	16
1.2 Классификация АРМ .....	17
1.3 Зачем необходим АРМ .....	19
1.4 Задачи автоматизированного рабочего места .....	22
2 Теоритическая часть .....	24
2.1 Постановка задачи.....	24
2.2 Этап начальной разработки БД.....	24
2.3 Проектирование БД.....	28
2.3.1 Концептуальное проектирование .....	28
2.3.2 Бизнес правила .....	29
2.3.3 Разработка и построение подробной ER–диаграммы на основании бизнес правил.....	34
2.3.4 Анализ информационных задач и круга пользователей системы.....	35
2.3.5 Определение атрибутов каждой сущности .....	36
2.4 Составление реляционных отношений .....	39
2.5 Физическое проектирование .....	47
2.6 Логическое проектирование .....	54
2.6.1 Обоснование выбора СУБД .....	55
2.6.2 Меры по обеспечению безопасности .....	55
2.6.3 Создание пользователей, логинов и задание им паролей .....	56
2.7 Установка и настройка Microsoft visual studio 2010 .....	57
2.7.1 Установка Microsoft visual studio 2010.....	57
2.7.2 Настройка MS visual studio 2010 .....	59
3 Техничко–экономическое обоснование проекта .....	63
3.1 Описание работы и обоснование необходимости .....	63
3.2 Расчет затрат на разработку информационных технологий.....	63
3.3 Расчет цены программного продукта.....	71
Вывод.....	72
4 Безопасность жизнедеятельности.....	73
4.1 Анализ потенциально опасных и вредных факторов, воздействующих на обслуживающий персонал при эксплуатации технического оборудования.....	73
4.2 Планировка рабочего места .....	74
4.3 Расчет вентиляции помещения .....	76
4.4 Анализ пожарной безопасности.....	78
4.5 Вентиляция помещения.....	80
Вывод.....	80
Заключение .....	82
Список литературы .....	83
Приложение А .....	84
Приложение Б.....	88
Приложение В.....	90

Приложение Г.....	92
Приложение Д.....	92
Приложение Е.....	92
Приложение Ж.....	93

## Введение

Ускоряющиеся темпы развития информатизации человечества ставят во главе необходимость освоения вычислительной техники в управленческих процессах. При автоматизации процесса обработки информации использование современных технологии позволяют ускорить, облегчить и упростить производительность выполняемого труда, повысить эффективность и скорость работы с данными и информацией, а также ускорить и обезопасить обмен управленческими данными.

Постепенно все большее применение находит новая концепция разделенных автоматизированных систем управления, которые используются для местной обработки данных, из чего следует, что это позволяет переконструировать разделение труда управленческого персонала и упростить выполнение ими своих автоматизированных функций.

Основной задачей данного проекта является автоматизация рабочего места сотрудника больницы для обслуживания клиентов на базе продукта Oracle Database Express Edition 11g Release 2 и Microsoft visual studio C#.

Автоматизированное рабочее место (далее АРМ) – система средств, которые нужны для автоматизации работы и дающие необходимую подготовку, изменение, добавление, поиск и удаление на экран и распечатку нужных для работы документов или данных используемых при профессиональной деятельности. АРМ дает специалисту все то необходимое, что поможет выполнять ту или иную поставленную задачу.

Необходимость накопления, сбора, переработки и распространения (обмена) данных стояла перед пользователем на всех этапах его развития. Очень долгое время главными инструментами для ее решения являлись человеческий мозг, язык и слух.

Значительное и полное изменение началось с приходом письменности, а затем с появлением книгопечатания. Поскольку в эту эпоху главным носителем информации была книга, то такой метод накопления, хранения, использования и распространения информации принято говорить “бумажной информатикой”.

Один из самых первых примеров комплексного системного использования ЭВМ стали так называемые административные системы для переработки информации: автоматизация бухгалтерских операций, банковского учета, бронирования и заказ билетов и т.п. Одним из главных значений для эффективной работы таких систем подобного рода является то, что эти системы опираются на автоматизированные информационные базы. При решении очередной задачи система нуждается во вводе только небольшой порции дополнительной информации, – остальное берется из информационной базы. Каждая порция вновь вводимой информации изменяет информационную базу системы. Эта база (информационная, или база данных) находится, таким образом, в состоянии непрерывного обновления, отражая все изменения, происходящие в реальном объекте, с которым имеет дело система.



Хранение информации в памяти персонального компьютера дает этим данным отличительно новое свойство динамичности, способности к мгновенной перестройке и мгновенному применению в поставленных на ПУ задачах. Автоматическая печать, которым снабжено любое современное оборудование, дают в крайнем случае способность мгновенной отсортировке нужных данных из этой информации в нужной форме представления.

По мере своего постепенного развития административные системы обработки информации начали переходить в автоматизированные системы управления, с такими же классами и объектами, в которых, не ограничиваются одним персональным компьютером.

Автоматизированная система управления – это такая система, в которой с помощью необходимых технических средств пользователь получает сбор, хранение, обработку данных, формулируемые оптимальные стратегии управления определенными компонентами и плагинами, а также выдача итоговых результатов пользователю или группе пользователей, берущих управление на себя.

Целью написания данного дипломного проекта являлись:

- Создать систему для облегчения прохождения медицинского осмотра.
- Обеспечить врачей средствами проверки состояния пациентов.
- Система должна быть динамической и легко расширяемой.

Структурно дипломная работа состоит из введения, четырех разделов и заключения.

В первом разделе определены требования, которым должен удовлетворять программный продукт. Разработана модель базы данных, рассмотрены технологии программирования, используемые в работе.

Во втором разделе приведено описание работы АРМ.

В третьем разделе приведено технико-экономическое обоснование реализации проекта и вычислена его себестоимость.

В четвертом разделе проведен анализ потенциально опасных и вредных производственных факторов проектируемого объекта, воздействующих на персонал, расчет кондиционирования офиса и пожарной безопасности проектируемого объекта.

# 1 Автоматизированное рабочее место

## 1.1 Понятие АРМ

Автоматизированное рабочее место (АРМ) можно определить, как совокупность информационно–программно–технических ресурсов, обеспечивающую конечному пользователю обработку данных и автоматизацию управленческих функций в конкретной предметной области.

Создание АРМ предполагает, что основные операции по накоплению, хранению и переработке информации возлагаются на вычислительную технику, а экономист выполняет часть ручных операций и операций, требующих творческого подхода при подготовке управленческих решений. Персональная техника применяется пользователем для контроля производственно–хозяйственной деятельности, изменения, значений отдельных параметров в ходе решения задачи, а также ввода исходных данных в АИС для решения текущих задач и анализа функций управления.

Анализируя сущность АРМ, специалисты определяют их чаще всего как профессионально–ориентированные малые вычислительные системы, расположенные непосредственно на рабочих местах специалистов и предназначенные для автоматизации их работ.

Для каждого объекта управления нужно предусмотреть автоматизированные рабочие места, соответствующие их функциональному назначению. Однако принципы создания АРМ должны быть:

- системность;
- гибкость;
- устойчивость;
- эффективность;
- максимальная ориентация на конечного пользователя;
- проблемная ориентация на решение определенного класса задач;
- эргономичность;
- принцип соответствия информационных потребностей пользователя используемым техническим средствам;
- принцип творческого контакта АРМ и их потенциальных пользователей.

Согласно принципу системности АРМ следует рассматривать как системы, структура которых определяется функциональным назначением.

Принцип гибкости означает приспособляемость системы к возможным перестройкам благодаря модульности построения всех подсистем и стандартизации их элементов.

Принцип устойчивости заключается в том, что система АРМ должна выполнять основные функции независимо от воздействия на нее внутренних и внешних возможных факторов. Это значит, что неполадки в отдельных ее

частях должны быть легко устранимы, а работоспособность системы – быстро восстанавливаема.

Наиболее простой функцией АРМ является информационно–справочное обслуживание, присущее любому АРМ. В зависимости от назначения АРМ могут включать экранные формы документов, расчетные алгоритмы, обеспечивающие обработку информации и отображение результатов, текстовые системы и т.д.

АРМ, созданные на базе ПК – наиболее простой и распространенный вариант, обеспечивающий:

- простоту и удобство пользования;
- простоту адаптации к конкретным функциям пользователя;
- компактность размещения и невысокие требования к условиям эксплуатации;
- простая организация технического обслуживания.

Эффективным режимом работы АРМ является его функционирование в рамках локальной вычислительной сети, когда необходимо распределить информационно–вычислительные ресурсы между несколькими пользователями.

Более сложные системы АРМ предполагают подключение нескольких ПК по каналам связи к главной ЭВМ или через специальное оборудование – к различным информационным службам и системам общего назначения (библиотечным системам, базам данных, информационно–поисковым системам и т.д.).

Работа пользователя с программным обеспечением АРМ реализуется, как правило, через меню. Минимальная аппаратная конфигурация для АРМ – ПК и принтер.

## **1.2 Классификация АРМ**

Множество известных АРМ может быть классифицировано на основе следующих обобщенных признаков:

- функциональная сфера использования (научная деятельность, проектирование, производственно–технологические процессы, организационное управление);
- тип используемой ЭВМ (микро–, мини–, макро–ЭВМ);
- режим эксплуатации (индивидуальный, групповой, сетевой);
- квалификация пользователей (профессиональные и непрофессиональные).

Внутри каждой из выделенных групп АРМ может быть проведена более детальная классификация.

Например, АРМ организационного управления могут быть разделены на АРМ руководителей организаций и подразделений, плановых работников, работников материально–технического снабжения, бухгалтеров и др. Условно все эти АРМы можно назвать АРМ экономиста.

Концептуальное отличие АРМ на базе ПЭВМ состоит в том, что АРМ открытая архитектура ПЭВМ функционально, физически и эргономически настраивается на конкретного пользователя (персональное АРМ) или группу пользователей (групповое АРМ).

Деловые АРМ сближают пользователя с возможностями современной информатики и Вт и создают условия для работы без посредника – профессионального программиста. При этом обеспечивается как автономная работа, так и возможность связи с другими пользователями в пределах организационных структур (с учетом особенностей этих структур).

Параметрический ряд деловых АРМ позволяет создать единую техническую, организационную и методологическую базу компьютеризации управления. Первоначально информационная технология локализуется в пределах персонального или группового АРМ, а в последующем (при объединении АРМ средствами коммуникации) создаются АРМ сектора, отдела, учреждения и формируется коллективная технология. тем самым достигается гибкость всей структуры и возможность наращивания информационной мощности.

Можно выделить три класса типовых АРМ:

- АРМ руководителя;
- АРМ специалиста;
- АРМ технического и вспомогательного персонала.

В основу классификации АРМ также может быть положен ряд классификационных признаков. С учетом областей применения возможна классификация АРМ по функциональному признаку:

- 1 АРМ административно–управленческого персонала.
- 2 АРМ проектировщика радиоэлектронной аппаратуры, автоматизированных систем управления и т.д.
- 3 АРМ специалиста в области экономики, математики, физики, и т.д.
- 4 АРМ производственно–технологического назначения.

Важным классификационным признаком АРМ является режим его эксплуатации, по которому выделяются одиночный, групповой и сетевой режимы эксплуатации.

Одним из подходов к классификации АРМ является их систематизация по видам решаемых задач. Возможны следующие группы АРМ:

- 1 Для решения информационно–вычислительных задач.
- 2 Для решения задач подготовки и ввода данных.
- 3 Для решения информационно–справочных задач.
- 4 Для решения задач бухгалтерского учета.
- 5 Для решения задач статистической обработки данных.
- 6 Для решения задач аналитических расчетов.

### 1.3 Зачем необходим АРМ

АРМ специалиста может включать следующие возможности:

- АРМ организация зависит от ее вида деятельности, а также может отличаться у сотрудников разных специальностей.

- Программное обеспечение АРМ может охватывать основные аспекты работы компании, а также может включать некие особенности, свойственные только отдельному предприятию.

- Автоматизированное рабочее место специалиста помогает контролировать состояние текущих дел и планировать на будущий период.

- АРМ заказчика может поставляться в базовой версии или разработанной на заказ.

- Разработка автоматизированного рабочего места начинается с выезда нашего технолога к заказчику.

- Организация автоматизированного рабочего места должна начинаться с персонала, расположенного на первой ступени должностной иерархии предприятия, и потом подниматься вверх до руководителя.

- Создание автоматизированного рабочего места сначала должно наладить повседневную работу сотрудников разных специальностей, а затем уже формировать сводные отчеты для руководителей разных уровней.

- Назначение автоматизированного рабочего места – навести порядок, наладить контроль, ускорить процесс работы. Автоматизированное рабочее место специалиста АРМ не должно игнорироваться самим сотрудником, поэтому руководители должны контролировать, что весь персонал работает в программе.

- Функции автоматизированного рабочего места заключаются в регистрации данных, контроле текущих дел и в формировании отчетности.

Согласно принципу системности, АРМ следует рассматривать как системы, структура которых определяется функциональным назначением. Принцип гибкости означает приспособленность системы к возможным перестройкам, благодаря модульности построения всех подсистем и стандартизации их элементов.

Эффективность АРМ следует рассматривать как интегральный показатель уровня реализации приведенных выше принципов, отнесенного к затратам на создание и эксплуатацию системы.

Структура АРМ – это совокупность его подсистем и элементов Рисунок 1.1. К обеспечивающим системам в первую очередь следует отнести: техническое, информационное, программное и организационное. Кроме того, существует целый ряд подсистем.

техническое обеспечение представляет собой комплекс технических средств, основой которого служит профессиональный персональный компьютер, предусматривающий работу специалиста без посредников (программистов, операторов и др.).

К комплексу технических средств следует отнести и средства коммуникаций для связи различных АРМ в сетях, а также средства телефонной связи.

Информационное обеспечение – это массивы информации, хранящиеся в локальных базах, данных. Информация организуется и хранится, в основном, на магнитных дисках. Управление ею осуществляется с помощью программной системы управления базами данных, которая производит запись информации, поиск, считывание, корректировку и решение информационных задач. В АРМ может быть несколько баз данных.

Организационное обеспечение включает средства и методы организации функционирования, совершенствования и развития АРМ, а также подготовки и повышения квалификации кадров.

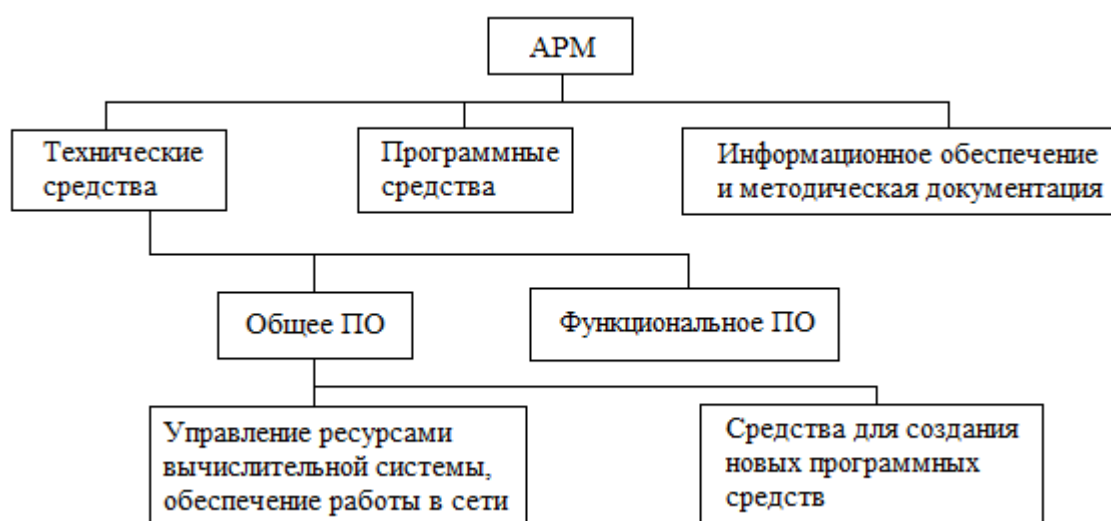


Рисунок 1.1 – Схема автоматизированного рабочего места

Организационное обеспечение АРМ включает комплекс документов, регламентирующих деятельность специалистов при использовании компьютера или терминала другого вида на рабочем месте и определяющих функции и задачи каждого специалиста.

Специалистом выполняются на АРМ следующие операции:

- ввод информации с документов при помощи клавиатуры (с визуальным контролем по экрану дисплея);
- ввод данных в ПЭВМ с магнитных носителей с других АРМ;
- прием данных в виде сообщений по каналам связи с других АРМ в условиях функционирования локальных вычислительных сетей;
- редактирование данных и манипулирование ими;
- накопление и хранение данных;
- поиск, обновление и защита данных;

- вывод на экран, печать, магнитный носитель результатной информации, а также различных справочных и инструктивных сообщений пользователю;

- формирование и передача данных на другие АРМ в виде файлов на магнитных носителях или по каналам связи в вычислительных сетях;

- получение оперативных справок по запросам.

Методическое обеспечение АРМ состоит из методических указаний, рекомендаций и положений по внедрению, эксплуатации и оценке эффективности их функционирования. Оно включает в себя также организованную машинным способом справочную информацию об АРМ в целом и отдельных его функциях, средства обучения работе на АРМ, демонстрационные примеры.

Использование АРМ в современном архиве максимально облегчает работу специалиста, высвобождая время и усилия, которые ранее расходовались на выполнение рутинных операций сбора данных и сложных расчетов, для творческой научно–обоснованной деятельности в решении профессиональных задач.

АРМ обладает следующими свойствами:

- Максимальная ориентация на конечного пользователя, достигаемая созданием инструментальных средств адаптации АРМ к уровню подготовки пользователя, возможностей его обучения и самообучения.

- Формализация профессиональных знаний, то есть возможность предоставления с помощью АРМ самостоятельно автоматизировать новые функции и решать новые задачи в процессе накопления опыта работы с системой.

- Проблемная ориентация автоматизированного рабочего места на решение определенного класса задач, объединенных общей технологией обработки информации, единством режимов работы и эксплуатации, что характерно для специалистов экономических служб.

- Модульность построения, обеспечивающая сопряжение АРМ с другими элементами системы обработки информации, а также модификацию и наращивание возможностей АРМ без прерывания его функционирования.

- Эргономичность, то есть создание для пользователя комфортных условий труда и дружественного интерфейса общения с системой.

- АРМ должен отвечать следующим требованиям:

- Своевременное удовлетворение информационной и вычислительной потребности специалиста.

- Минимальное время ответа на запросы пользователя.

- Адаптация к уровню подготовки пользователя и его профессиональным запросам.

- Простота освоения приемов работы на АРМ и легкость общения, надежность и простота обслуживания.

- терпимость по отношению к пользователю.

- Возможность быстрого обучения пользователя.

## 1.4 Задачи автоматизированного рабочего места

Помимо этого, такое рабочее место имеет профессиональную ориентацию на конкретную, определенную предметную сферу и выступает в роли средства общения специалиста с автоматизированной информационной системой компании.

Если в качестве критерия брать организационную структуру управления компанией, то можно умозрительно выделить АРМ руководителя, АРМ управленческих работников среднего и АРМ работников оперативных уровней. В условиях выборочного распределения информации все перечисленные сотрудники имеют совершенно разные потребности в информационной поддержке.

Для каждого конкретного специалиста автоматизированное рабочее место обеспечивает:

- Удобство, простоту использования и дружеское отношение к пользователю.
- Легкость адаптации к конкретным требующимся пользователю функциям.
- Компактность размещения, а также доступные требования по условиям эксплуатации.
- Высокую живучесть и гарантированную надежность.
- Несложную организацию технического обслуживания.

Наиболее эффективный режим функционирования автоматизированного рабочего места, есть его функционирование интегрировано по отношению к локальной вычислительной сети компании.

Созданные на предприятии АРМ специалистов позволяют их пользователям трудиться в диалоговом режиме, быстро и эффективно решать текущие задачи, вести необходимый контроль, удобно вводить нужные данные и осуществлять обработку текущей информации, а также определять достоверность полученной информации, выводить ее для последующего анализа и передавать по каналами связи.

Эффективное функционирование автоматизированной информационной системы и АРМ базируется на комплексном использовании современных технических и программных средств обработки информации в совокупности с современными организационными формами размещения техники.

Выбор организационных форм использования программно–технических средств целесообразно осуществлять с учетом их рассредоточения по уровням иерархии управления в соответствии с организационной структурой автоматизируемого объекта. При этом основным принципом выбора является коллективное обслуживание пользователей, отвечающее структуре автоматизируемого объекта.

С учетом современной функциональной структуры территориальных органов управления совокупность программно–технических средств должна образовывать по меньшей мере трехуровневую глобальную систему обработки



данных с развитым набором периферийных средств каждого уровня Рисунок 1.2.

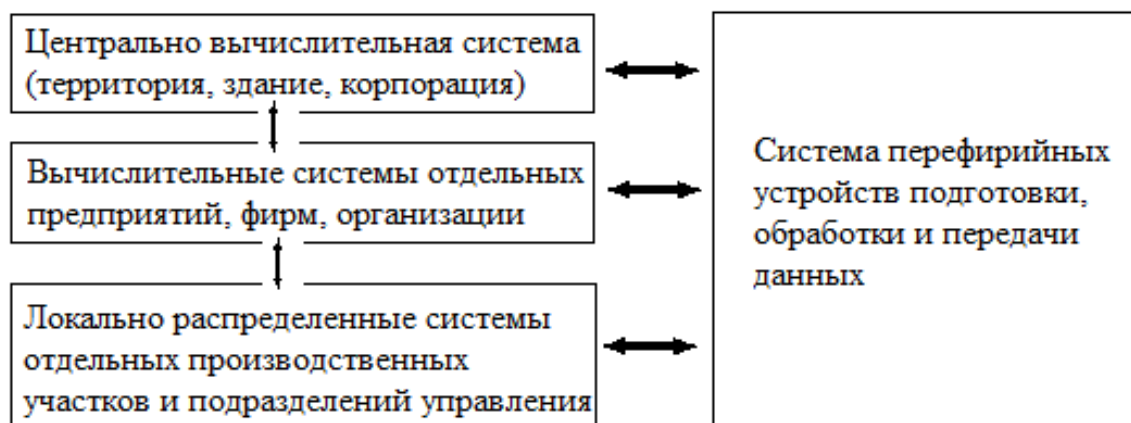


Рисунок 1.2 – Принципиальная схема многоуровневой организации программно-технических средств

Первый уровень – центральная вычислительная система территориального или корпоративного органа, включающая одну или несколько мощных ПК или мэйнфреймов. Ее главная функция – общий, экономический и финансовый контроль, информационное обслуживание работников управления.

Второй уровень – вычислительные системы предприятий (объединений), организаций и фирм, которые включают мэйнфреймы, мощные ПК, обеспечивают обработку данных и управление в рамках структурной единицы.

Третий уровень – локально распределенные вычислительные сети на базе ПК, обслуживающие производственные участки нижнего уровня. Каждый участок оснащен собственным ПК, который обеспечивает комплекс работ по первичному учету, учету потребности и распределения ресурсов. В принципе это может быть АРМ, выполняющее функциональные вычислительные процедуры в рамках определенной предметной области.

В то же время на каждом уровне иерархии управления имеют место три способа организации технических средств: централизованный, децентрализованный и иерархически распределенный. Первый способ предполагает выполнение всех работ по обработке данных, начиная со сбора и регистрации данных, в одном центре обработки; второй предусматривает предварительную обработку информации, которая не требует создания очень крупных массивов данных, на периферийном оборудовании удаленного пользователя низовых звеньев автоматизируемого объекта; при третьем способе технология обработки оптимально распределена по уровням управления системы.

## **2 теоритическая часть**

### **2.1 Постановка задачи**

Необходимо разработать автоматизированное рабочее место сотрудника больницы города Алматы.

Данное автоматизированное рабочее место позволит посетителям, пациентам и сотрудникам больницы получить им необходимую информацию за короткое время, записаться на прием не становясь в живую очередь и узнать всю информацию о лечащем враче, курсе лечения и прописанных больным препаратах. Пропадает необходимость ведения амбулаторной карты, т.е. любая больница может запросить информацию о пациенте из последней больницы где находился больной пациент. Данная система позволит пациенту на дому узнавать результаты анализов, теряя необходимость посещения больничных учреждений. Пациент заполняет свой профиль в котором ведется все обследования которые он проходил, все препараты которые были ему прописаны, врачи которые принимали пациента и место где проходил курс лечения пациент.

таким образом, пользователь–пациент и пользователь–врач после регистрации могут получать всю необходимую им информацию за короткое время, получать результаты всех проведенных анализов, курсов лечения, и препаратов которые принимал пациент, в свою очередь пациент может знать какой лечащий врач у него, его статистику, должность, а также когда и какая операция будет проводиться и кем.

Врач может вести записи о пациенте создавать заметки, отправлять сообщения своим пациентам в режиме реального времени, например, о консультации или изменениях в ходе лечения.

### **2.2 Этап начальной разработки БД**

Курс «Проектирование базы данных» – это предмет, изучающийся в АУЭС, на котором проходятся основы проектирования и реализации баз данных.

Базовой СУБД для изучения данного курса является Oracle Database Express Edition 11g Release 2 от компании Oracle.

Студенты должны изучать теоритические материалы, после каждой главы на лабораторных занятиях им предлагается выполнить несколько практических заданий для более глубокого освоения материала.

Сами задания выполняются либо в программах для проектирования ER–диаграмм, например, ERWIN или Microsoft Visio, а также в среде SQL Server Management Studio. После выполнения задания студенты должны показать решение преподавателю.

также необходимо проводить тестирование студентов на предмет освоения ими теоритических материалов.

Данная дипломная работа упростит процесс лечения пациента врачом, уход за больными, регистрации на необходимые процедуры за счет уменьшения времени.

Вся информация о пациентах, врачах и леченях будет храниться в БД.

Базовые сущности данной базы данных:

1 Пользователи. Эта сущность служит для хранения информации о пользователях–пациентах системы, об их личных данных и уровне доступа.

2 Модули лечения. В данной сущности хранится вся информация о прохождений лечений пациентом и его наблюдающим врачом.

3 Здания. В этой сущности хранится информация о учреждениях в которых работает врач и наблюдался или наблюдается пациент.

4 Препараты. В данной сущности хранится информация о лекарствах и кому они прописаны.

5 Врачи. Здесь хранится информация о врачах, за кем они наблюдали или наблюдают.

6 Должности. Здесь хранятся данные занимаемых должностях врачом.

7 Курс лечения. В данную сущность будут записываться данные кто какое проходит лечение.

8 Сообщения. Здесь в зашифрованном виде хранятся все пересылаемые пациентом или врачом сообщения.

### *UML диаграммы*

Для моделирования статических объектов в объектно–ориентированной концепции UML существует 12 диаграмм, но так как большинство объектов БД «Интерактивной обучающей системы» являются статическими, используем самые основные:

- 1 Диаграмма компонентов.
- 2 Диаграмма прецедентов.
- 3 Диаграмма классов.
- 4 Диаграмма развертывания.

### *Диаграмма компонентов*

Диаграмма компонентов показывает набор компонентов и отношений между ними.

Компонент – это физически заменяемая часть системы, которая имеет набор интерфейсов и обеспечивает их реализацию.

Приложение обращается за получением или добавлением информации к базе данных Oracle. Но обращается не напрямую, а с помощью Java. таким образом, Java является интерфейсом между приложениями базой данных, обеспечивая их взаимодействие. Диаграмма компонентов представлена на рисунке 2.1.

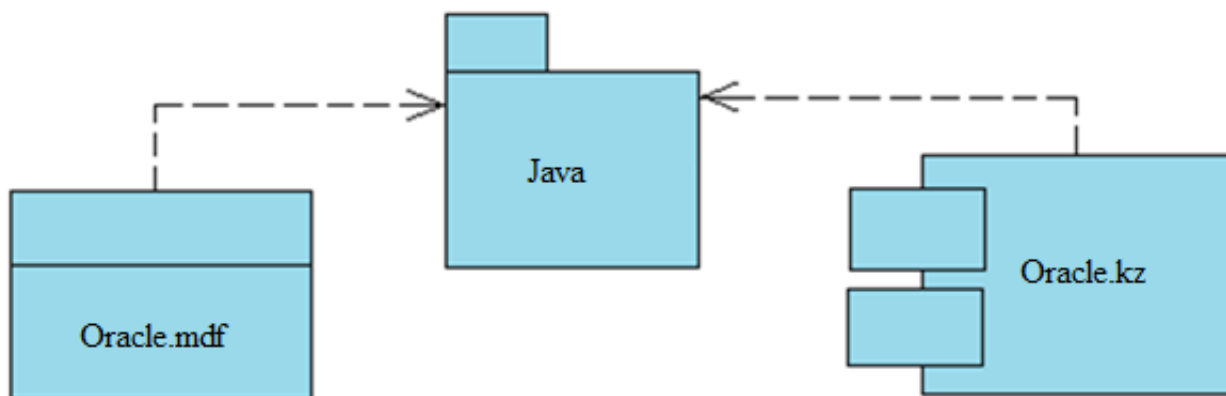


Рисунок 2.1 – Диаграмма компонентов

*Диаграмма прецедентов*

Для характеристики взаимодействия пользователей с базами данных применяется диаграмма прецедентов или вариантов использования. Диаграмма прецедентов представлена на рисунке 2.2.



Рисунок 2.2 – Диаграмма прецедентов

Из нее видно, что пользоваться базой могут 2 группы пользователей: операторы и пользователи.

### Диаграмма классов

Диаграмма классов, показанная на рисунке 2.3, в UML является частным случаем ER–диаграммы. ER–диаграммы используются для логического проектирования баз данных. Главное их отличие: в ER–диаграмме уделяется внимание структуре данных, а в диаграмме классов – поведению классов.

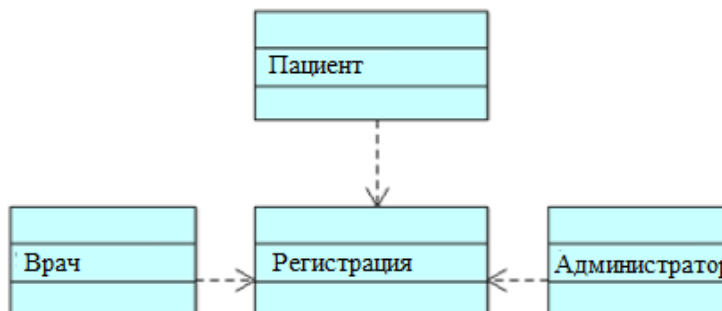


Рисунок 2.3 – Диаграмма классов

### Диаграмма развертывания

Диаграмма развертывания, представленная на рисунке 2.4, Deployment diagram в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать web–сайт диаграмма развертывания должна показывать, какие аппаратные компоненты («узлы») существуют (например, web–сервер, сервер базы данных, сервер приложения), какие программные компоненты («артефакты») работают на каждом узле (например, web–приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, OLEDB, RMI).

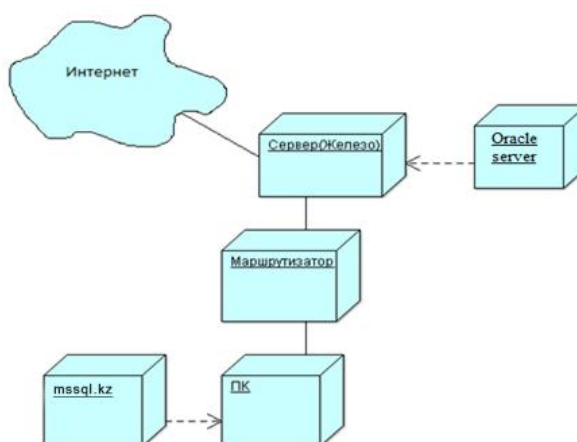


Рисунок 2.4 – Диаграмма развертывания

Узлы представляются как прямоугольные параллелепипеды с артефактами, расположенными в них, изображенными в виде прямоугольников. Узлы могут иметь подузлы, которые представляются как вложенные прямоугольные параллелепипеды. Один узел диаграммы развертывания может

концептуально представлять множество физических узлов, таких как кластер серверов баз данных.

Существует два типа узлов:

- узел устройства;
- узел среды выполнения.

Узлы устройств являются физически вычислительными ресурсами со своей памятью и сервисами для выполнения программного обеспечения, такие как обычные ПК, мобильные телефоны. Узел среды выполнения – это программный вычислительный ресурс, который работает внутри внешнего узла и который предоставляет собой сервис, выполняющий другие исполняемые программные элементы.

Диаграмма развертывания нашей системы очень проста, так как не требует специального оборудования и может устанавливаться на любой ПК в кабинете врача или лечащегося пациента [2].

## **2.3 Проектирование БД**

### **2.3.1 Концептуальное проектирование**

Учитывая параметры системы, описанных выше выберем основные сущности и связи между ними, изобразим их на предварительной ER–диаграмме, которая приведена на рисунке 2.5.

Для построения ER–диаграммы воспользуемся программным обеспечением от компании Microsoft под названием Visio. Данная программа позволяет не только составить диаграмму сущность–связь, но и в дальнейшем экспортировать данную схему в любую из множества поддерживаемых ей СУБД, в моем случае это будет экспорт в Oracle Database Express Edition 11g Release 2.

На основе данной диаграммы сформируем наши бизнес–правила. Бизнес правилами называются такие правила, которые устанавливают ограничения для связей между сущностями.

Далее установим связи между сущностями, опираясь на описание операций на предварительной ER–диаграмме. точнее связи наших сущностей можно установить на основе бизнес–правил, которые, в свою очередь, построены на основе подробного описания операций.

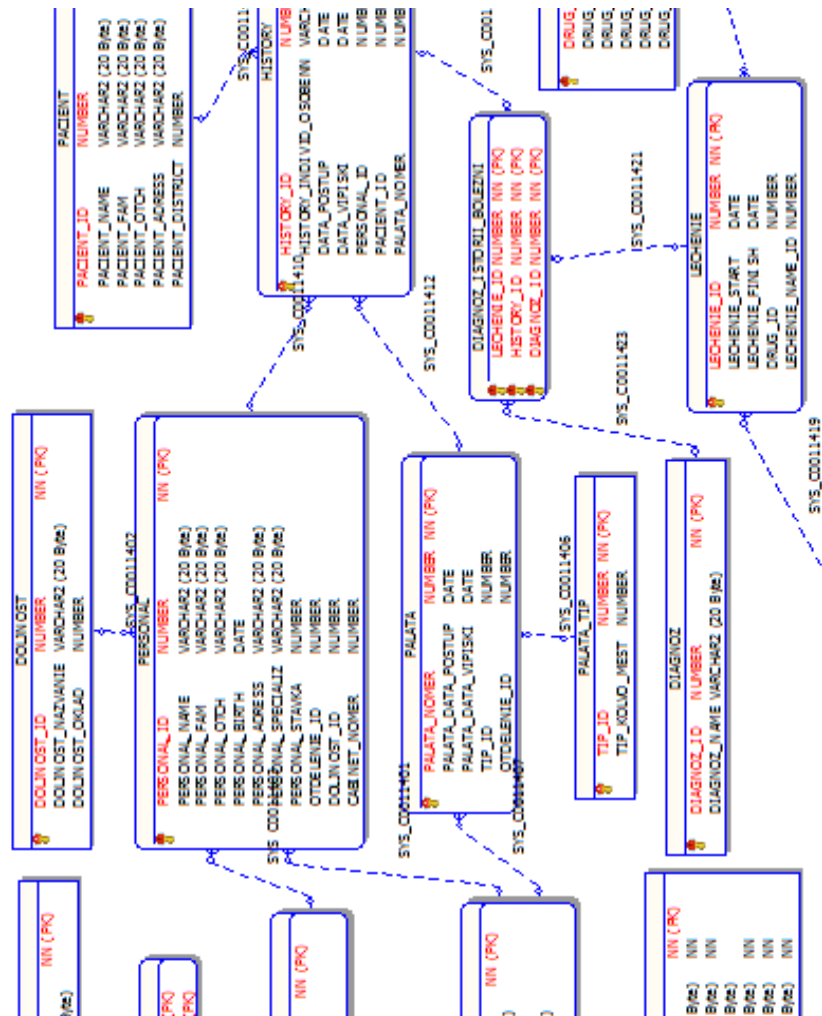


Рисунок 2.5 – Общая ER–диаграмма

### 2.3.2 Бизнес правила

1 Информация о пациенте заносится в две сущности. В одной из них находится история болезни, а во второй личные данные пользователей, такие как имя, фамилия, отчество, адрес, район где он прописан (Рисунок 2.6).

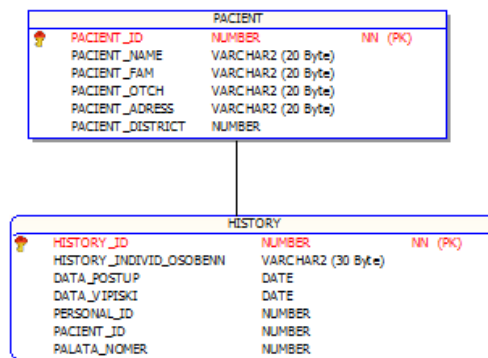


Рисунок 2.6 – Сегмент ER –диаграммы для бизнес–правила 1

2 Каждое отделение находится в определенной больнице. В каждой больнице может быть неограниченное количество отделений (Рисунок 2.7).

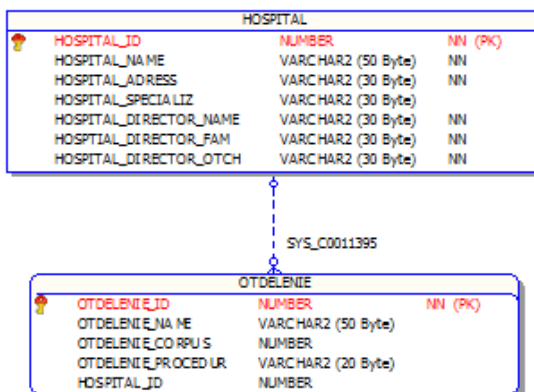


Рисунок 2.7 – Сегмент ER–диаграммы для бизнес–правила 2

3 В каждом отделений есть свой кабинет (Рисунок 2.8).

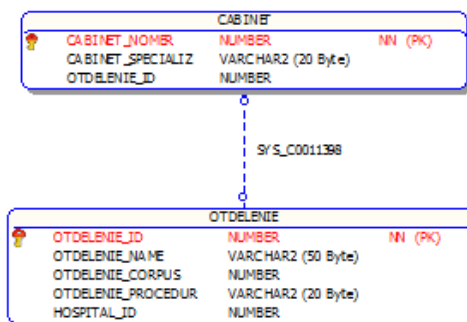


Рисунок 2.8 – Сегмент ER–диаграммы для бизнес–правила 3

4 В истории болезни указывается какой пациент смотрится у какого врача. У одного пациента может быть множество врачей. У одного врача может быть множество пациентов (Рисунок 2.9).

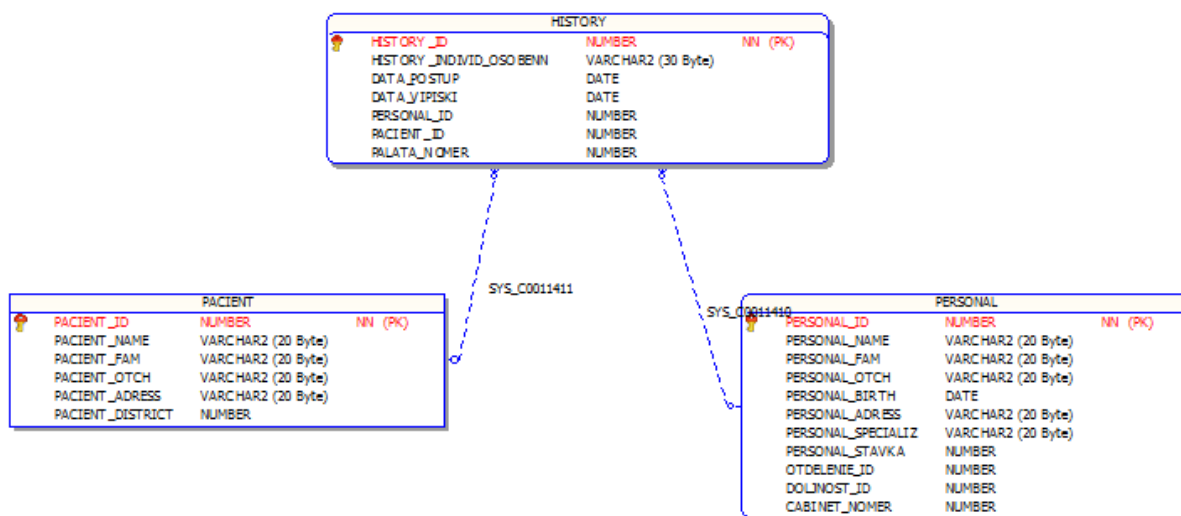




Рисунок 2.9 – Сегмент ER –диаграммы для бизнес–правила 4

5 В больнице кабинеты имеют множество типов. Один тип кабинета может быть у множества кабинетов (Рисунок 2.10).

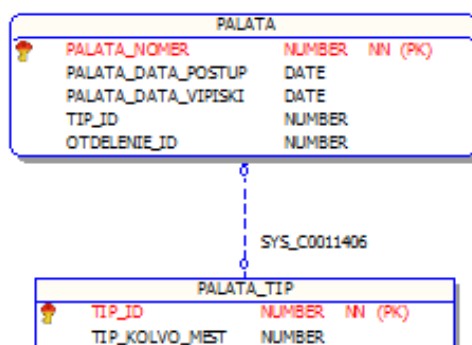


Рисунок 2.10 – Сегмент ER–диаграммы для бизнес–правила 5

6 К каждому лечению есть свой диагноз и у каждого диагноза есть свое лечение. У одного лечение может быть несколько диагнозов. И у одного диагноза может быть несколько лечений (Рисунок 2.11).

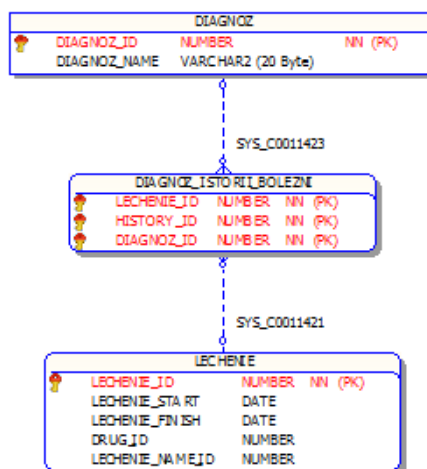


Рисунок 2.11 – Сегмент ER –диаграммы для бизнес–правила 6

7 Лекарство лечит болезни. Одно лекарство лечит несколько болезни. И у одного лечения может быть несколько лекарств (Рисунок 2.12).

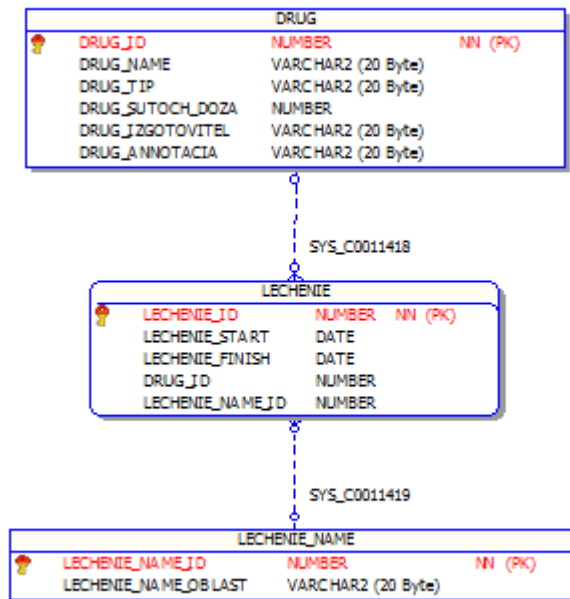


Рисунок 2.12 – Сегмент ER–диаграммы для бизнес–правила 7

8 В каждом кабинете есть свой сотрудник (Рисунок 2.13).

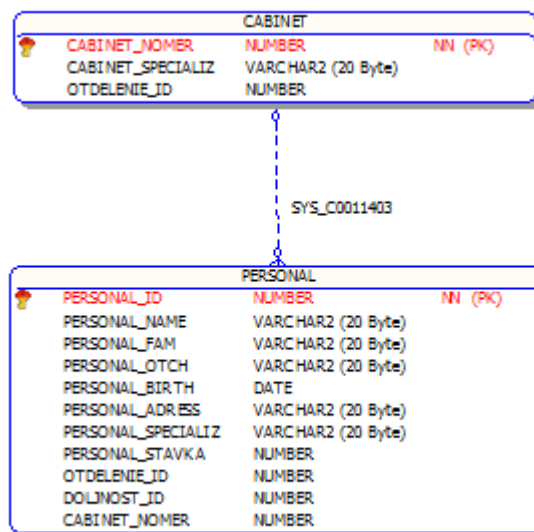


Рисунок 2.13 – Сегмент ER–диаграммы для бизнес–правила 8

9 В каждом кабинете есть оборудование. В одном кабинете может быть несколько оборудований. Одно оборудование может быть в нескольких кабинетах (Рисунок 2.14).

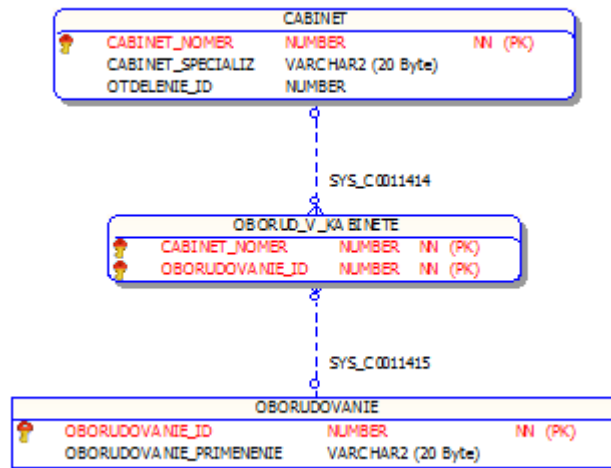


Рисунок 2.14 – Сегмент ER–диаграммы для бизнес–правила 9

10 Каждый сотрудник больницы занимает одну должность (Рисунок 2.15).

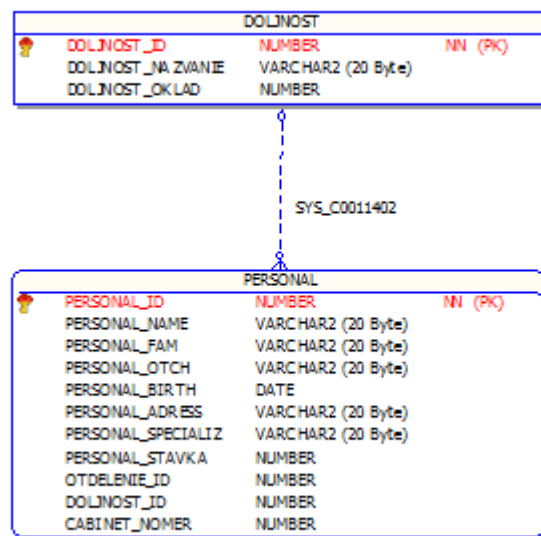


Рисунок 2.15 – Сегмент ER –диаграммы для бизнес–правила 10

11 Каждый сотрудник прикреплен к кабинету (Рисунок 2.16).

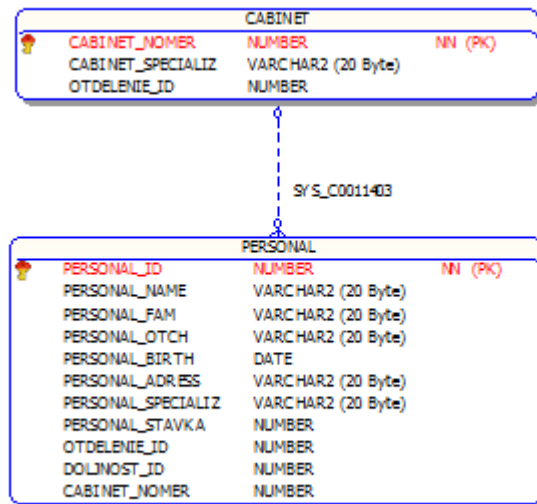


Рисунок 2.16 – Сегмент ER–диаграммы для бизнес–правила 11

12 Не может быть пользователей с одинаковыми логинами или электронными адресами.

13 Пользователи делятся на 3 уровня доступа к информации: пациент, врач, администратор.

### 2.3.3 Разработка и построение подробной ER–диаграммы на основании бизнес правил

Концептуальная модель интерактивной обучающей системы для курса «Проектирование базы данных» представлена на рисунке 2.17.

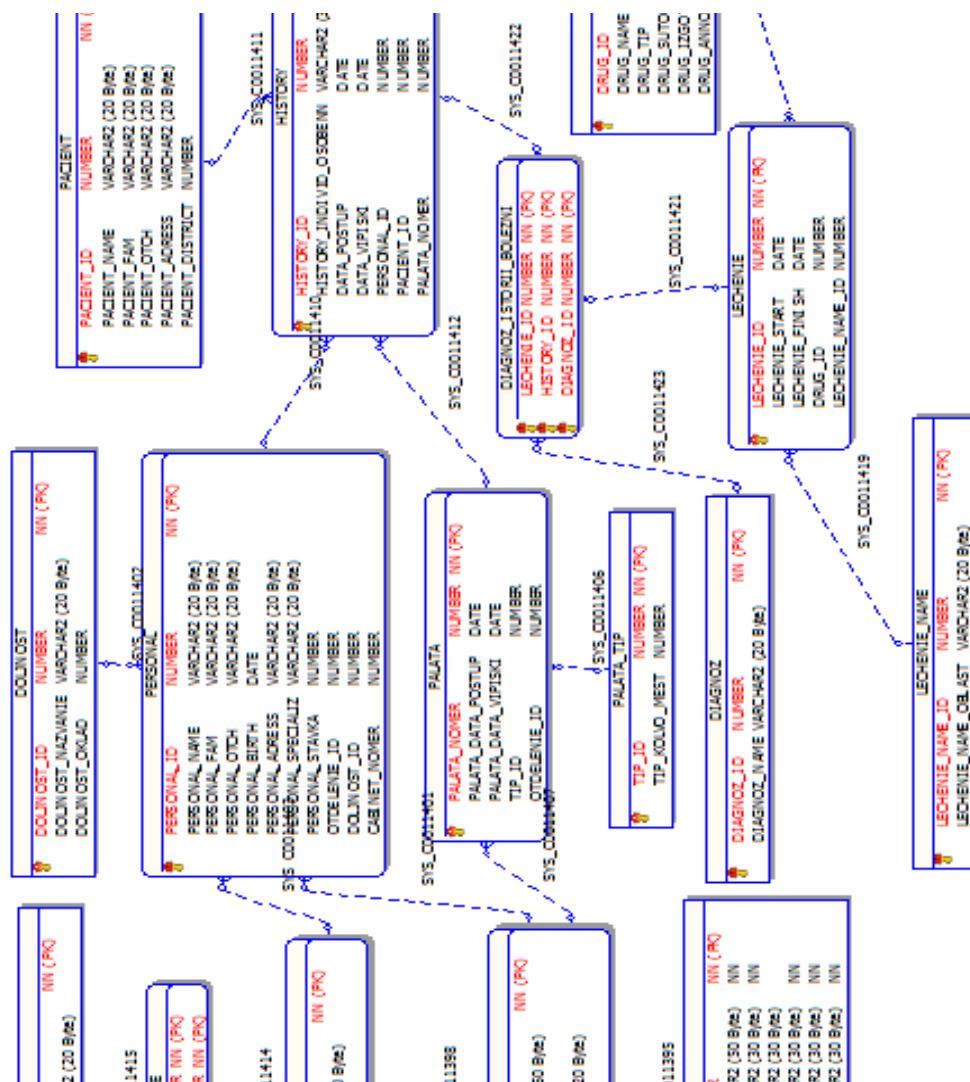


Рисунок 2.17 – Окончательный вариант концептуальной модели интерактивной обучающей системы

### 2.3.4 Анализ информационных задач и круга пользователей системы

С данной автоматизированной рабочей системой могут работать следующие типы пользователей:

- пациент;
- врач;
- администратор.

При работе с системой пациент должен иметь возможность совершать следующие действия:

- 1 Изменять свои личные данные.
- 2 Читать свою историю болезни.
- 3 Записываться на прием врачу.
- 4 Записываться на сдачу анализов.
- 5 Вести общение с лечащим врачом и администратором.

Врач должен иметь возможность совершать следующие действия:

- 1 Проверка состояний пациентов.

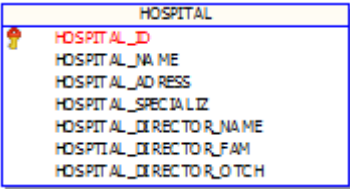
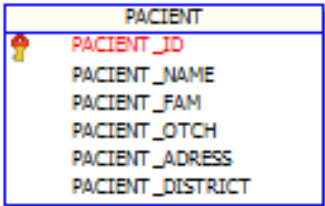
- 2 Просматривать информацию о пациентах.
  - 3 Должен иметь доступ ко всей информации о дате и времени, а также кто именно проходил обследование.
  - 4 Должна быть возможность о рассылке сообщений.
  - 5 Вступать в диалог с любым пользователем системы.
  - 6 Управлять базой пациентов, истории болезни и методе лечений.
  - 7 Управлять очередью записывающихся к нему пациентов.
- Администратор должен иметь возможность совершать следующие действия:

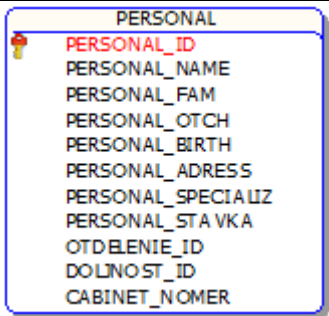
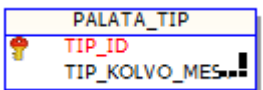
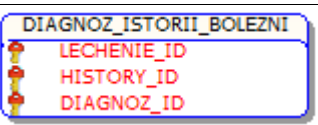
- 1 Выдавать пользователям привилегии пациентов и врачей.
- 2 Возможность блокировать, удалять, создавать дополнительные аккаунты.
- 3 Следить за состоянием всей системы.

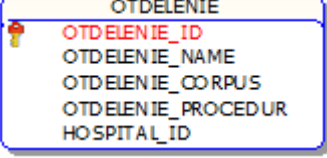
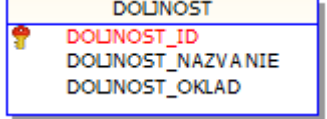
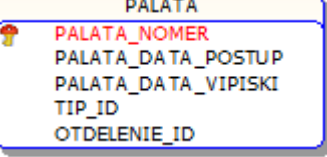
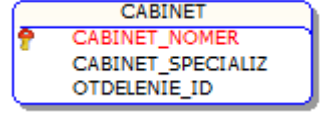
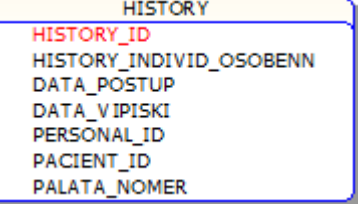
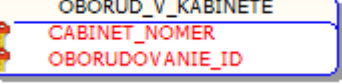
### 2.3.5 Определение атрибутов каждой сущности

В таблице 2.1 представлены атрибуты каждой из сущностей созданной базы данных. Данная таблица необходима нам для того, чтобы продолжить дальнейшее проектирование базы данных. В ней определяются все атрибуты, которые будут в таблицах, а также приводится их краткое описание. В каждом сегменте таблицы описывается сущность таблицы, краткое описание содержания таблицы, описывающее какое значение имеет оно в базе данных, определяющее атрибут каждой сущности.

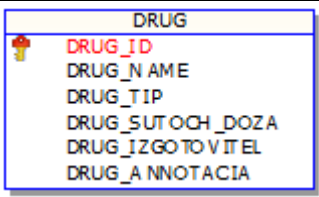
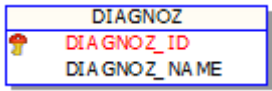
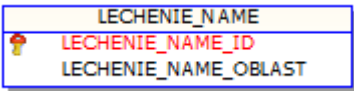
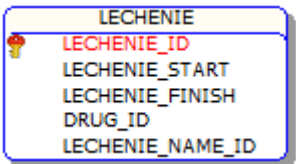
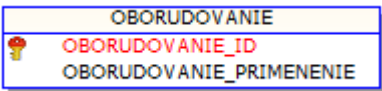
таблица 2.1 – Определение атрибутов каждой сущности

Сегмент ER модели	Описание
	<p>таблица Hospital (Больницы) содержит следующие данные:</p> <p>Hospital_id – идентификатор больницы</p> <p>Hospital_name – название больницы</p> <p>Jospital_adress – адрес больницы</p> <p>Hospital_specializ – специализация больницы,</p> <p>Hospital_director_name – имя директора</p> <p>Hospital_director_fam – фамилия директора</p> <p>Hospital_director_otch – отчество директора</p>
	<p>таблица Pacient (Пациенты) содержит следующие данные:</p> <p>Pacient_id – уникальный номер пациента</p> <p>Pacient_name – имя пациента</p> <p>Pacient_fam – фамилия пациента</p>

	<p>Pacient_otch – отчетсво пациента  Pacient_adress – адрес пациента  Pacient_district – район, где прописан пациент</p>
	<p>таблица Personal (Персонал) следующие данные:  Personal_id – уникальный номер сотрудника  Personal_name – имя сотрудника  Personal_fam – фамилия сотрудника  Personal_otch – отчество сотрудника  Personal_birth – дата рождения сотрудника  Personal_adress – адрес сотрудника  Personal_specializ – специализация сотрудника  Personal_stavka – заработная плата сотрудника  Otdelenie_id – отделение сотрудника  Cabinet_nomer – номер кабинета</p>
	<p>таблица Palata_tip (тип палаты) содержит следующие данные:  Tip_id– идентификатор типа палаты  Tip_kolvo_mest– количесвто мест для типа палаты</p>
	<p>таблица Diagnoz_istorii_bol (Диагноз истории болезни) содержит следующие данные:  Lechenie_id – идентификатор лечения  History_id – идентификатор истории  Diagnoz_id – идентификатор диагноза</p>

Сегмент ER модели	Описание
 <p>OTDELENIE  <b>OTDELENIE_ID</b>  OTDELENIE_NAME  OTDELENIE_CORPUS  OTDELENIE_PROCEDUR  HOSPITAL_ID</p>	<p>таблица Otdelenie (Отделение) содержит следующие данные:  Otdelenie_id – идентификатор отделения  Otdelenie_name – название отделения  Otdelenie_corpus – корпус отделния  Otdelenie_procedur – процедуры отделения  Hospital_id – идентификатор больницы? Где находится отделение</p>
 <p>DOLJNOST  <b>DOLJNOST_ID</b>  DOLJNOST_NAZVANIE  DOLJNOST_OKLAD</p>	<p>таблица Doljnost (Должность) содержит следующие данные:  Doljnost_id – уникальный номер должности  Doljnost_nazvanie – название должности  Doljnost_oklad – оклад для должности</p>
 <p>PALATA  <b>PALATA_NOMER</b>  PALATA_DATA_POSTUP  PALATA_DATA_VIPISKI  TIP_ID  OTDELENIE_ID</p>	<p>таблица Palata (Палата) содержит следующие данные:  Palata_nomer – идентификатор палаты  Palata_data_postup – дата поступления в палату  Palata_data_vipiski – дата выписки из палаты  Tip_id – тип палаты  Otdelenie_id – идентификатор отделения где находится палата</p>
 <p>CABINET  <b>CABINET_NOMER</b>  CABINET_SPECIALIZ  OTDELENIE_ID</p>	<p>таблица Cabinet (Кабинет) содержит следующие данные:  Cabinet_nomer – номер кабинета  Cabinet_spcializ – специализация кабинета  Otdelenie_id – идентификатор отделения, где находится кабинет</p>
 <p>HISTORY  <b>HISTORY_ID</b>  HISTORY_INDIVID_OSOBENN  DATA_POSTUP  DATA_VIPISKI  PERSONAL_ID  PACIENT_ID  PALATA_NOMER</p>	<p>таблица History (История болезни) содержит следующие данные:  Hist_id – уникальный номер истории болезни  Hist_ind_osb – индивидуальные особенности  Data_postup – дата поступления  Data_vipiski – дата выписки  Personal_id – идентификатор персонала  Pacient_id – идентификатор пациента  Palata_nomer – номер палаты</p>
 <p>OBOHUD_V_KABINETE  <b>OBOHUDOVANIE_ID</b>  CABINET_NOMER</p>	<p>таблица Oborud_v_kab (Оборудование в кабинете) содержит следующие данные:  Cabinet_nomer – номер кабинета  Oborudov_id – идентификатор оборудования</p>



Сегмент ER модели	Описание
	<p>таблица Drug (Лекарства) содержит следующие данные:</p> <p>Drug_id – уникальный номер лекарства</p> <p>Drug_name – название лекарства</p> <p>Drug_tip – тип лекарства</p> <p>Drug_sutoch_doza – суточная доза</p> <p>Drug_izgotovit – изготовитель лекарства</p> <p>Drug_annotacia – аннотация к лекарству</p>
	<p>таблица Diagnoz (Диагноз) содержит следующие данные:</p> <p>Diagnoz_id – уникальный номер диагноза</p> <p>Diagnoz_name – название диагноза</p>
	<p>таблица Lechenie_name (Название лечения) содержит следующие данные:</p> <p>Lechenie_name_id – идентификатор лечения</p> <p>Lechenie_name_oblast – область лечения</p>
	<p>таблица Lechenie (Лечение) содержит следующие данные:</p> <p>Lechenie_id – идентификатор лечения</p> <p>Lechenie_start – дата начала лечения</p> <p>Lechenie_finish – дата окончания лечения</p> <p>Drug_id – идентификатор лечения</p> <p>Lechenie_name_id – идентификатор названия лечения</p>
	<p>таблица Oborudovanie (Оборудование) содержит следующие данные:</p> <p>Oborudovanie_id – идентификатор оборудования</p> <p>Oborudovanie_prim – применение оборудования</p>

## 2.4 Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они есть).

Отношения приведены в таблицах 2.2 – 2.16. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Обязательное поле для краткости обозначено not null, необязательное – null. типы данных обозначаются так: number – числовой, varchar2 – символьный, date – дата [3].

таблица 2.2 – Otdelenie

Содержание поля	Имя поля	тип, длина	Примечание
ИД отделения	Otdelenie_id	int	Первичный ключ
Название	Otdelenie_name	varchar (255)	Not null
Корпус	Otdelenie_corpus	int	null
Процедуры	Otdelenie_procedur	varchar (255)	Not null
ИД больницы	Hospital_id	int	Not null(внешний ключ)

таблица 2.3 – Drug

Содержание поля	Имя поля	тип, длина	Примечание
ИД лекарства	Drug_id	int	Первичный ключ
Название	Drug_name	varchar (255)	Not null
тип	Drug_tip	varchar (255)	null
Суточная доза	Drug_sutoch_doza	int	Not null
Изготовитель	Drug_izgotovitel	varchar (255)	Not null
Аннотация	Drug_annotacia	varchar (255)	Not null

таблица 2.4 – Palata

Содержание поля	Имя поля	тип, длина	Примечание
ИД палаты	Palata_nomer	int	Первичный ключ
Дата поступления	Palata_data_postup	datetime	Not null
Дата выписки	Palata_data_vipiski	datetime	Not null
тип	Tip_id	int	Внешний ключ
Отделение	Otdelenie_id	int	Внешний ключ

таблица 2.5 – Palata\_tip

Содержание поля	Имя поля	тип, длина	Примечание
ИД тип	Tip_id	int	первичный ключ
Количество мест	Tip_kolvo_mest	int	Not null

таблица 2.6 – Oborud\_v\_kabinete

Содержание поля	Имя поля	тип, длина	Примечание
Номер кабинета	Cabinet_nomer	int	Внешний ключ
Оборудование	Oborudovanie_id	int	Внешний ключ

таблица 2.7 – History

Содержание поля	Имя поля	тип, длина	Примечание
ID истории	History_id	int	Первичный ключ
Индив. особенн.	History_ind_osob	varchar (255)	Not null
Дата поступления	Data_postup	datetime	Not null
Дата выписки	Data_vipiski	datetime	Not null
Персонал	Personal_id	int	Not null
Пациент	Pacient_id	int	Not null
Палата	Palata_nomer	int	Not null

таблица 2.8 – Personal

Содержание поля	Имя поля	тип, длина	Примечание
ID очереди	Personal_id	Int	Первичный ключ
Имя	Personal_name	varchar (255)	Not null
Фамилия	Personal_fam	varchar (255)	Not null
Отчество	Personal_otch	varchar (255)	Null
Дата рождения	Personal_birth	Datetime	Null
Адрес	Personal_adress	varchar (255)	Null
Специализация	Personal_specializ	varchar (255)	Null
Ставка	Persona_stavka	Int	Not null
Отделение	Otdelenie_id	int	Внешний ключ
Должность	Doljnost_id	int	Внешний ключ
Кабинет	Cabinet_nomer	int	Внешний ключ

таблица 2.9 – Cabinet

Содержание поля	Имя поля	тип, длина	Примечание
ID кабинета	Cabinet_nomer	Int	Первичный ключ
Специализация	Cabinet_specializ	varchar (255)	Not null
Отделение	Otdelenie_id	Int	Внешний ключ

таблица 2.10 – Hospital

Содержание поля	Имя поля	тип, длина	Примечание
ID сообщения	Hospital_id	int	Первичный ключ
Название	Hospital_name	Varchar(200)	null
Адрес	Hospital_adress	Varchar(200)	Not null
Специализация	Hospital_specializ	Varchar(200)	Null
Имя	Hospital_dir_name	Varchar(200)	Not null
Фамилия	Hospital_dir_fam	Varchar(200)	null
Отчество	Hospital_dir_otch	Varchar(200)	Not null

таблица 2.11 – Doljnost

Содержание поля	Имя поля	тип, длина	Примечание
ID должности	Doljnost_id	int	Первичный ключ
Название	Doljnost_nazvnaie	Varchar(200)	Not null
Оклад	Doljnost_oklad	int	null

таблица 2.12 – Pacient

Содержание поля	Имя поля	тип, длина	Примечание
ID пациента	Pacient_id	Int	Первичный ключ
Имя	Pacient_name	Varchar(200)	Not null
Фамилия	Pacient_fam	Varchar(200)	Not null
Отчество	Pacient_otch	Varchar(200)	Not null
Адресс	Pacient_adress	Varchar(200)	Not null
Район	Pacient_district	Int	Not null

таблица 2.13 – Diagnoz

Содержание поля	Имя поля	тип, длина	Примечание
ID диагноза	Diagnoz_id	int	Первичный ключ
Название	Diagnoz_name	Varchar(200)	Not null

таблица 2.14 – Lechenie\_name

Содержание поля	Имя поля	тип, длина	Примечание
ID лечения	Lech_name_id	int	Первичный ключ
Область	Lech_name_oblast	varchar(200)	Not null

таблица 2.15 – Lechenie

Содержание поля	Имя поля	тип, длина	Примечание
ID лечения	Lechenie_id	Int	Первичный ключ
Начало лечения	Lechenie_start	datetime	Not null
Конец лечения	Lechenie_finish	datetime	Not null
ID лекарства	Drug_id	Int	Внешний ключ
ID лечения	Lechenie_name_id	Int	Внешний ключ

таблица 2.16 – Diagnoz\_istorii\_bolezni

Содержание поля	Имя поля	тип, длина	Примечание
ID лечения	Lechenie_id	int	Внешний ключ
ID истории	History_id	int	Внешний ключ

ID диагноз	Diagnoz_id	int (128)	Внешний ключ
------------	------------	-----------	--------------

Перед запуском БД необходимо выставить необходимые размеры первоначальной БД и ее роста. На этом этапе, необходимо знать какой объем памяти будет занимать создаваемая база данных. Объем внешней памяти, необходимый для функционирования системы, складывается из двух составляющих: память, занимаемая модулями СУБД (ядро, утилиты, вспомогательные программы), и память, отводимая под данные (М<sub>д</sub>). Наиболее существенным обычно является М<sub>д</sub>. Объем памяти, занимаемый программными модулями пользователя, обычно невелик по сравнению с объемом самих данных, поэтому может не учитываться. В проекте рассчитывается предполагаемый максимальный объем памяти, занимаемой БД. Расчет физической памяти приводится в таблицах 2.17 – 2.31.

таблица 2.17 – Расчет физической памяти для таблицы Otdelenie

Имя поля	тип	Длина
Otdelenie_id	int	4
Otdelenie_name	varchar (255)	255
Otdelenie_corpus	int	4
Otdelenie_procedur	varchar (255)	255
Hospital_id	int	4

Общая длина строки: 518 байт

Число строк: ~ 500

Общий объем требуемой памяти: ~ 485 500 байт

таблица 2.18 – Расчет физической памяти для таблицы Palata\_tip

Имя поля	тип	Длина
Tip_id	int	4
Tip_kolvo_mest	int	4

Общая длина строки: 8 байт

Число строк: ~ 15

Общий объем требуемой памяти: ~ 120 байт

таблица 2.19 – Расчет физической памяти для таблицы Drug

Имя поля	тип	Длина
Drug_id	int	4
Drug_name	varchar (255)	255
Drug_tip	varchar (255)	255
Drug_sutoch_doza	int	4
Drug_izgotovitel	varchar (255)	255
Drug_annotacia	varchar (255)	255

Общая длина строки: 1016 байт  
 Число строк: ~ 20  
 Общий объем требуемой памяти: ~ 20 320 байт

таблица 2.20 – Расчет физической памяти для таблицы Palata

Имя поля	тип	Длина
Palata_nomer	int	4
Palata_data_postup	datetime	3
Palata_data_vipiski	datetime	3
Tip_id	int	4
Otdelenie_id	int	4

Общая длина строки: 18 байт  
 Число строк: ~ 3 000  
 Общий объем требуемой памяти: ~ 54 000 байт

таблица 2.21 – Расчет физической памяти для таблицы Oborud\_v\_kabin

Имя поля	тип	Длина
Cabinet_nomer	int	4
Oborudovanie_id	int	4

Общая длина строки: 8 байт  
 Число строк: ~ 600  
 Общий объем требуемой памяти: ~ 4 800 байт

таблица 2.22 – Расчет физической памяти для таблицы History

Имя поля	тип	Примечание
History_id	int	4
History_ind_osob	varchar (255)	255
Data_postup	datetime	3
Data_vipiski	datetime	3
Personal_id	int	4
Pacient_id	int	4
Palata_nomer	int	4

Общая длина строки: 277 байт  
 Число строк: ~ 10 000  
 Общий объем требуемой памяти: ~ 2 770 000 байт

таблица 2.23 – Расчет физической памяти для таблицы Cabinet

Имя поля	тип, длина	Примечание
Cabinet_nomer	Int	4
Cabinet_specializ	varchar (255)	255

Otdelenie_id	Int	4
--------------	-----	---

Общая длина строки: 263 байт

Число строк: ~ 200

Общий объем требуемой памяти: ~ 52 260 байт

таблица 2.24 – Расчет физической памяти для таблицы Personal

Имя поля	тип, длина	Примечание
Personal_id	Int	4
Personal_name	varchar (255)	255
Personal_fam	varchar (255)	255
Personal_otch	varchar (255)	255
Personal_birth	Datetime	3
Personal_adress	varchar (255)	255
Personal_specializ	varchar (255)	255
Persona_stavka	Int	4
Otdelenie_id	int	4
Doljnost_id	int	4
Cabinet_nomer	int	4

Общая длина строки: 1029 байт

Число строк: ~ 12 000

Общий объем требуемой памяти: ~ 1 348 000 байт

таблица 2.25 – Расчет физической памяти для таблицы Hospital

Имя поля	тип, длина	Примечание
Hospital_id	int	4
Hospital_name	Varchar(200)	200
Hospital_adress	Varchar(200)	200
Hospital_specializ	Varchar(200)	200
Hospital_dir_name	Varchar(200)	200
Hospital_dir_fam	Varchar(200)	200
Hospital_dir_otch	Varchar(200)	200

Общая длина строки: 1 204 байт

Число строк: ~ 50 000

Общий объем требуемой памяти: ~ 60 200 000байт

таблица 2.26 – Расчет физической памяти для таблицы Doljnost

Имя поля	тип	Примечание
Doljnost_id	int	4
Doljnost_nazvnaie	Varchar(200)	200
Doljnost_oklad	int	4

Общая длина строки: 208 байт  
 Число строк: ~ 50  
 Общий объем требуемой памяти: ~ 10 040 байт

таблица 2.27 – Расчет физической памяти для таблицы Diagnoz\_ist\_bol

Имя поля	тип	Примечание
Lechenie_id	int	4
History_id	int	4
Diagnoz_id	int	4

Общая длина строки: 12 байт  
 Число строк: ~ 200  
 Общий объем требуемой памяти: ~ 2 400 байт

таблица 2.28 – Расчет физической памяти для таблицы Pacient

Имя поля	тип	Примечание
Pacient_id	Int	4
Pacient_name	Varchar(200)	200
Pacient_fam	Varchar(200)	200
Pacient_otch	Varchar(200)	200
Pacient_adress	Varchar(200)	200
Pacient_district	Int	4

Общая длина строки: 808 байт  
 Число строк: ~ 10  
 Общий объем требуемой памяти: ~ 8 080 байт

таблица 2.29 – Расчет физической памяти для таблицы Diagnoz

Имя поля	тип	Примечание
Diagnoz_id	int	4
Diagnoz_name	Varchar(200)	200

Общая длина строки: 204 байт  
 Число строк: ~ 40 800  
 Общий объем требуемой памяти: ~ 174 200 байт

таблица 2.30 – Расчет физической памяти для таблицы Lechenie\_name

Имя поля	тип, длина	Примечание
Lech_name_id	int	4
Lech_name_oblast	varchar(200)	200

Общая длина строки: 204 байт  
 Число строк: ~ 3 000



Общий объем требуемой памяти: ~ 612 000 байт

таблица 2.31 – Расчет физической памяти для таблицы Lechenie

Имя поля	тип, длина	Примечание
Lechenie_id	Int	4
Lechenie_start	datetime	3
Lechenie_finish	datetime	3
Drug_id	Int	4
Lechenie_name_id	Int	4

Общая длина строки: 18 байт

Число строк: ~ 50

Общий объем требуемой памяти: ~ 900 байт

таким образом, из полученных данным можно рассчитать приблизительный максимальный объем базы данных. Объем будет равен сумме объемов всех таблиц.

$M_d = 230000 + 304800 + 1536000 + 10160 + 454005 + 485500 + 102200 + 72348000 + 25950000 + 84400 + 10800 + 2120 + 285 + 174200 + 45000 + 25900 = 101763370$  байт = 97 (Мб).

## 2.5 Физическое проектирование

На рисунке 2.18 нужно указать почтовый ящик и пароль от My Oracle Support аккаунта для получения информации от Oracle.

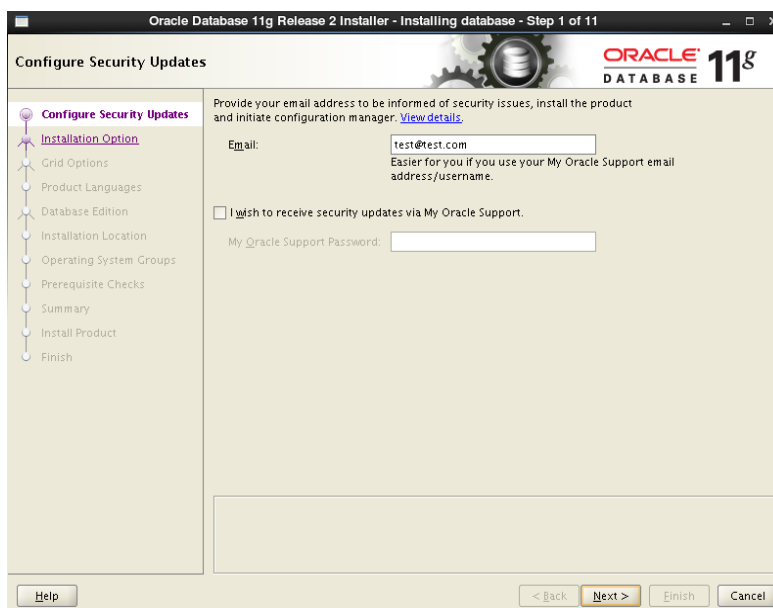


Рисунок 2.18 – Окно регистрации продукта

Затем выбираем пункт Install database software only как указано на рисунке 2.19.

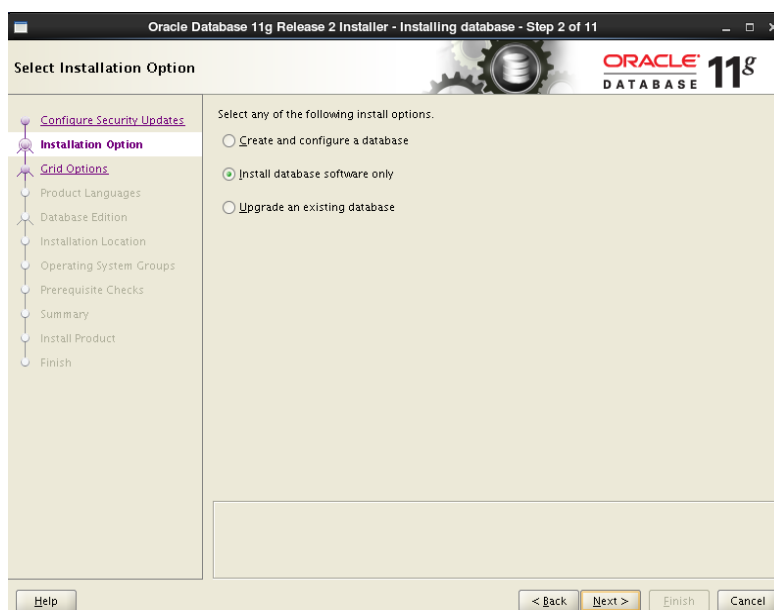


Рисунок 2.19 – Окно выбора инициализации продукта

После выбираем в какой кластер устанавливать данный продукт RAC или single database как показано на рисунке 2.20.

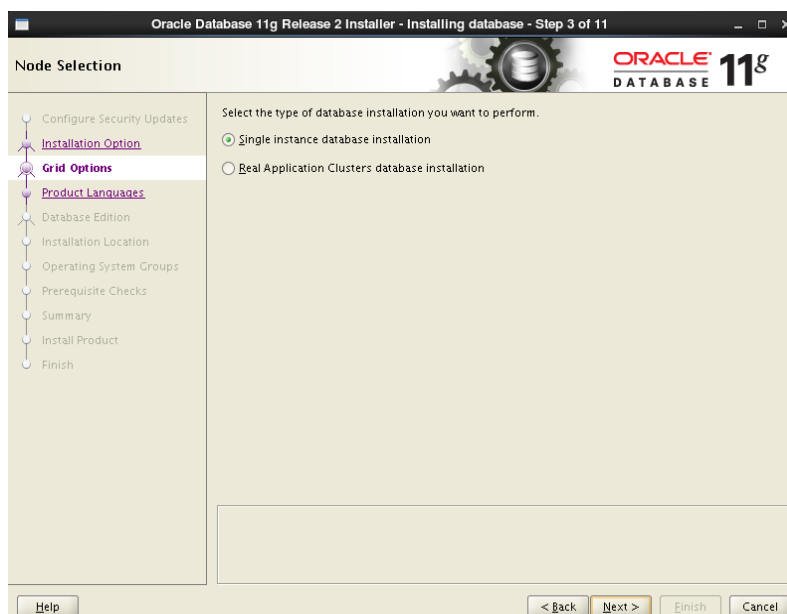


Рисунок 2.20 – Выбор типа базы данных

Следующее окно выбор языка как показано на рисунке 2.21.

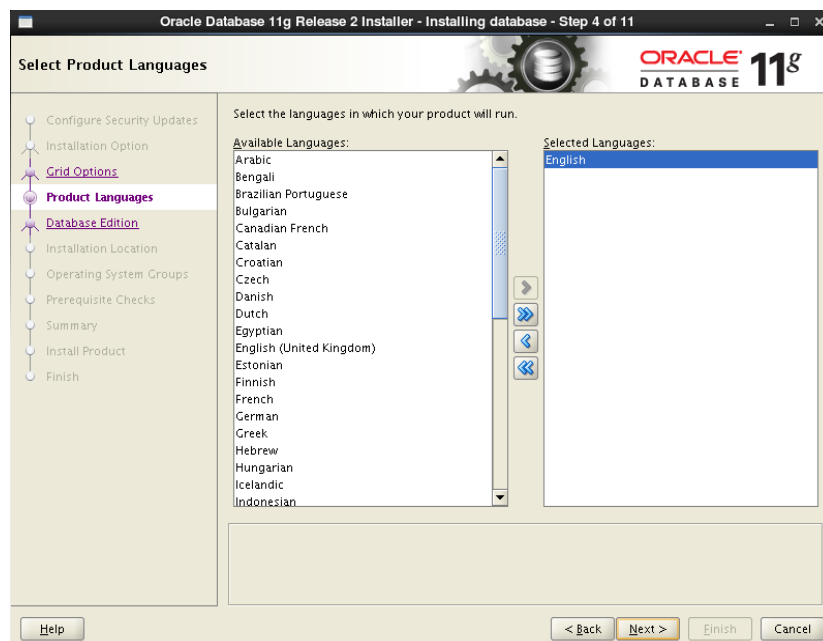


Рисунок 2.21 – Окно выбора языка

На следующем рисунке 2.22 выбираем какую именно редакцию БД хотим установить.

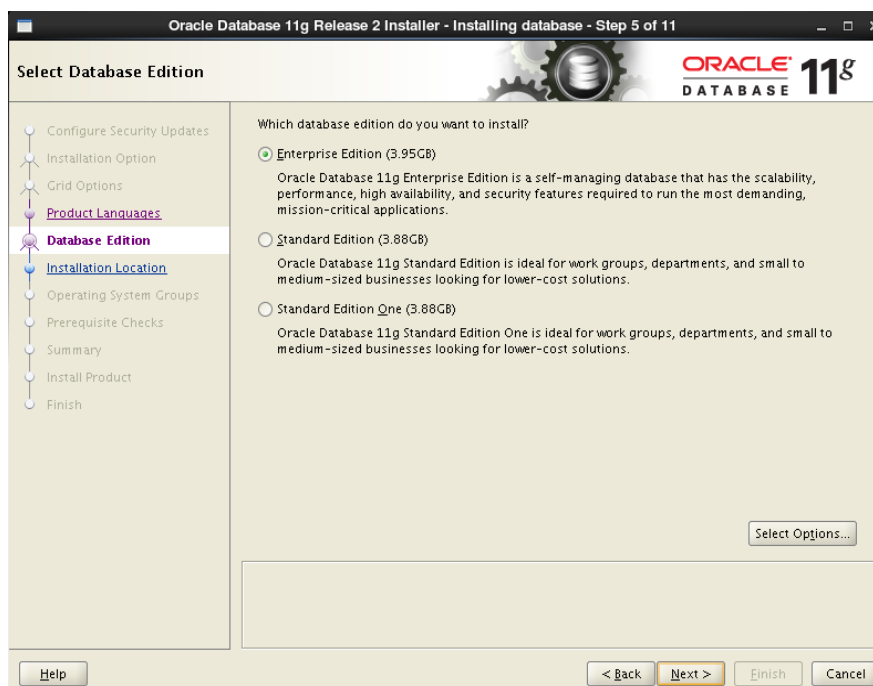


Рисунок 2.22 – Выбор редакции БД

Как показано на рисунке 2.23 указываем файлы БД.

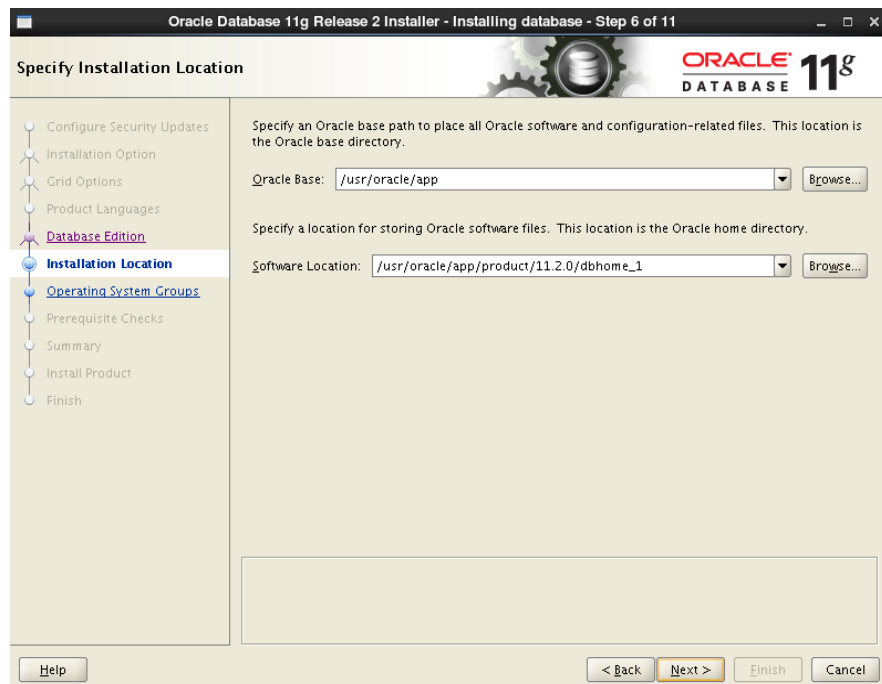


Рисунок 2.23 – Путь к файлам БД

Как показано на рисунке 2.24 указываем путь хранения файлов инвентаря БД.

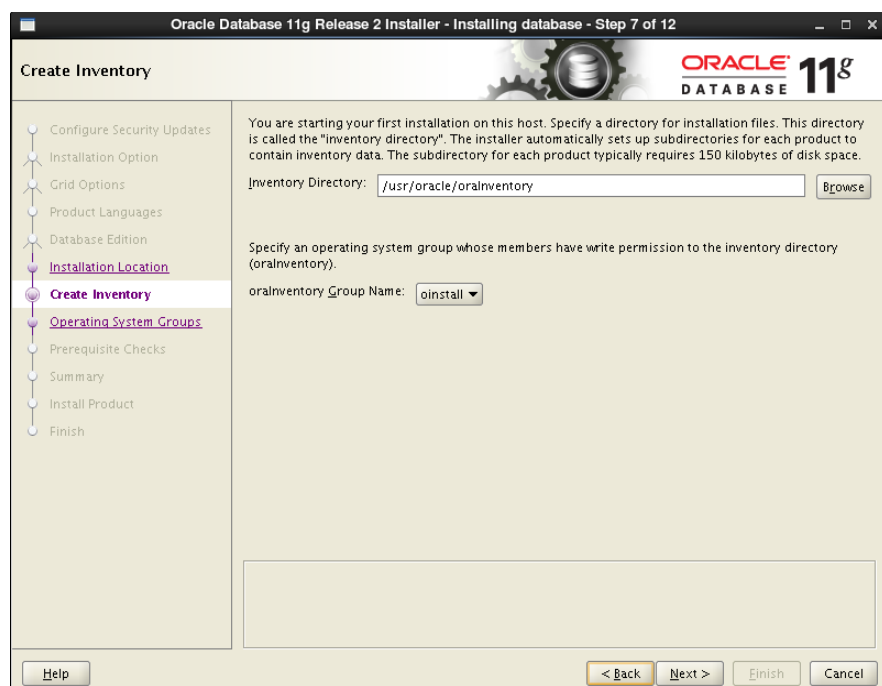


Рисунок 2.24 – Хранение файлов инвентаря БД

Указываем группу пользователей данной БД как показано рисунке 2.25.

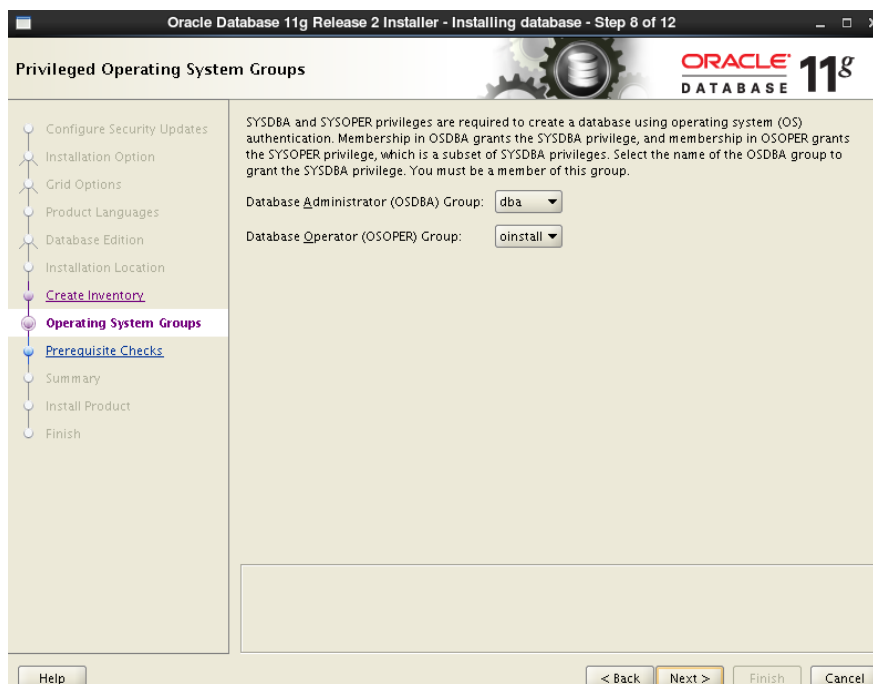


Рисунок 2.25 – Окно групп пользователей

Вовремя этого этапа происходит автоматическая проверка наличия всех необходимых пакетов и размеру памяти кластера. Версии пакетов могут отличаться вследствие этого проверку пройти не могут, для этого необходимо установить статус Ignored, как показано на рисунке 2.26.

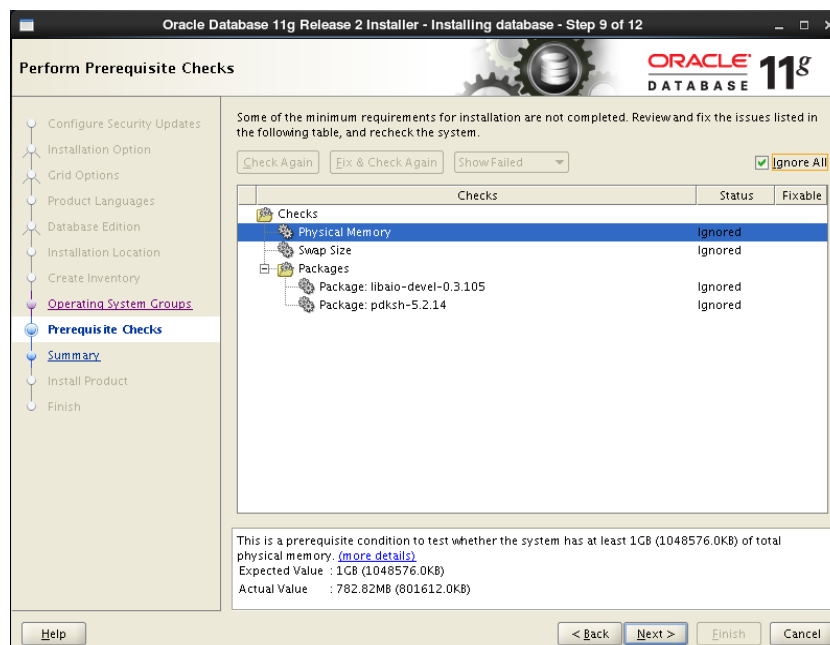


Рисунок 2.26 – Окно проверки пакетов

Перед началом установки на окне изображаются все параметры конфигурации, которые были выбраны как показано на рисунке 2.27.

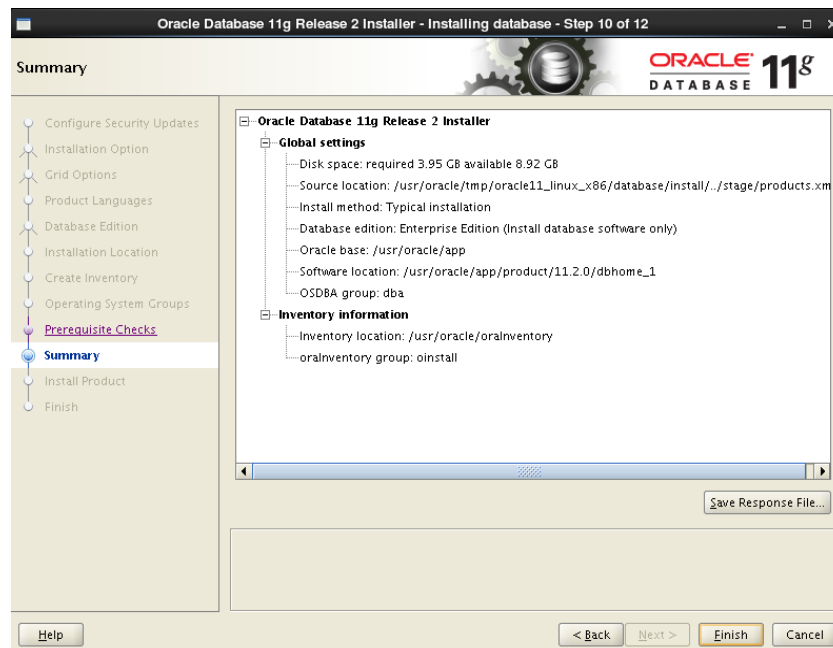


Рисунок 2.27 – Параметры конфигурации

На рисунке 2.28 изображен процесс установки.

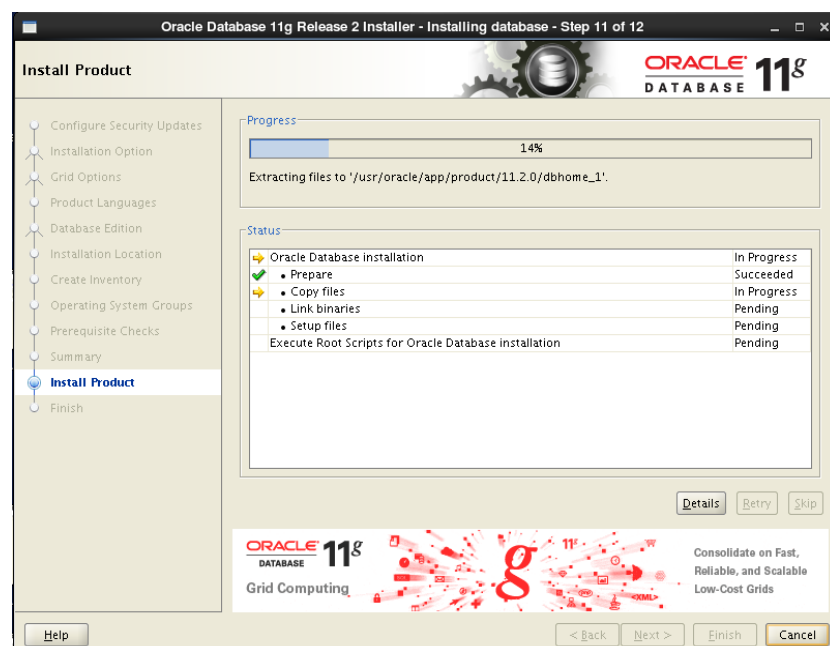


Рисунок 2.28 – Процесс установки

По окончании установки необходимо набрать и запустить скрипты, указанные на рисунке 2.29 находясь под правами root.

```
#/usr/oracle/oraInventory/orainstRoot.sh
#/usr/oracle/app/product/11.2.0/dbhome_1/root.sh
```

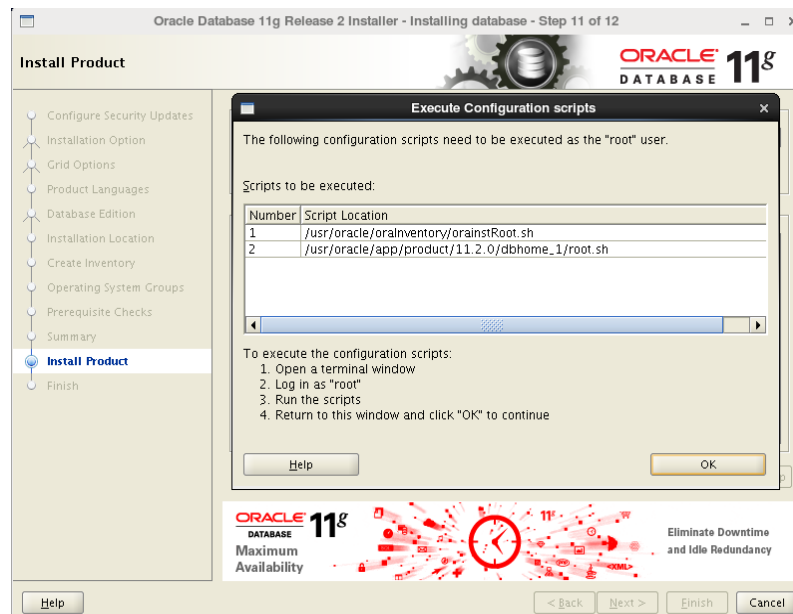


Рисунок 2.29 – Скрипты пользователя root

также необходимо добавить в файл профиля пользователя oracle

```
# vi ~/.bash_profile
```

```
Export ORACLE_HOME=$ORACLE_BASE/product/11.2.0/dbhome_1
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID=ORCL
```

### *Изменение файла tnsnames.ora вручную*

Необходимо внести изменения в файл tnsname.ora для того чтобы сконфигурировать метод локального именования. Необходимо перейти в каталог хранения файла tnsnames.ora, `$ORACLE_HOME/network/admin`, и отредактировать этот файл. Для добавления новых баз данных в систему необходимо также физически добавить отображения имен служб новых баз данных в файл tnsnames.ora для всех пользователей.

```
# TNSNAMES.ORA Network Configuration File:
/u01/app/oracle/product/10.1.0/db_1/network/admin/tnsnames.ora
# generated by Oracle configuration tools.
orcl =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = localhost) (PORT =
1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
    )
  )
salesprod =
  (DESCRIPTION =
    (ADDRESS_LIST =
```

```

        (ADDRESS = (PROTOCOL = TCP) (HOST =
172.11.150.1) (PORT = 1521))
    )
    (CONNECT_DATA =
        (SERVICE_NAME = salesprod.world)
    )
)

```

## 2.6 Логическое проектирование

Цель логического проектирования заключается в создании реляционных табличных структур на языке DDL.

Создание Базы данных:

```

create tablespace Hospital datafile
'/u01/app/oracle/oradata/Hospital.dat' size 100M
reuse autoextend on next 5M maxsize 200M
/
Commit

```

Ниже представлен код создания нескольких таблиц:

```

create table hospital
(
hospital_id int primary key NOT NULL,
hospital_name varchar(50) NOT NULL,
hospital_adress varchar(30) NOT NULL,
hospital_specializ varchar(30),
hospital_director_name varchar(30) NOT NULL,
hosptial_director_fam varchar(30) NOT NULL,
hospital_director_otch varchar (30) NOT NULL
);

create table doljnost
(
doljnost_id int primary key not null,
doljnost_nazvanie varchar(20),
doljnost_oklad int
);

create table oborudovanie
(
oborudovanie_id int primary key not null,
oborudovanie_primenenie varchar(20)
);

```



## 2.6.1 Обоснование выбора СУБД

Для разработки базы данных для бэкэнда приложения была выбрана СУБД Oracle Database Express Edition 11g Release 2. Она помогает существенно снизить затраты на поддержку и повысить качество работы с данными за счет мощной интерпретации языка SQL, названного TRANSACT SQL, а такое приложение как Oracle database, позволяет облегчить настройку и поддержку БД системы. Система является, безопасной, надежной, а также простой в управлении. Идеально соответствует выполнению нужных нам задач, связанных с разработкой и поддержкой базы данных, в том числе для web-приложений, а также других систем с использованием крупных объемов данных.

В результате выполнения кода, получим настоящую базу данных. Диаграмма физической модели данной базы данных представлена на рисунке 2.30.

Версия Oracle database, должна была заменить которая Oracle database, кодовое имя Katmai получила. В период активной разработки Microsoft неохотно делилась информацией крайне о новой версии. На Oracle database презентации Пол Флесснер (на тот момент занимавший пост вице-президента подразделения Microsoft, занимавшегося разработкой Oracle database), уверенно заявил, что релиз новейшей версии состоится не позднее, чем через два года после выхода Oracle database. Однако в апреле 2007 года ещё не было никакой информации о скором выходе продукта, или хотя бы о начале его бета-тестирования. тем не менее, в австрийском блоге на TechNet была опубликована информация о программе Katmai Technology Adoption Program (сокр. TAP), начало которой было якобы запланировано на июнь 2007 года. также были упомянуты слухи о том, что новая версия выйдет в 2008 году, но Microsoft на тот момент ни подтверждала, ни отрицала эту информацию. Некоторые источники привязывали выход Katmai к выходу Longhorn Server and Visual Studio Orcas, из-за чего согласно этой информации новая версия должна была выйти в первой половине 2008 года. Oracle также отказывалась комментировать эту информацию.

## 2.6.2 Меры по обеспечению безопасности

Одной из наиболее важных частей в БД является разработка прав доступа к ней, т.к. нужна защита от несанкционированного доступа и защита от доступа. Для защиты от сбоев разрабатывается стратегия резервного копирования. В моем случае, из-за специфики web-приложений ограничения доступа к информации осуществляется через само приложение.

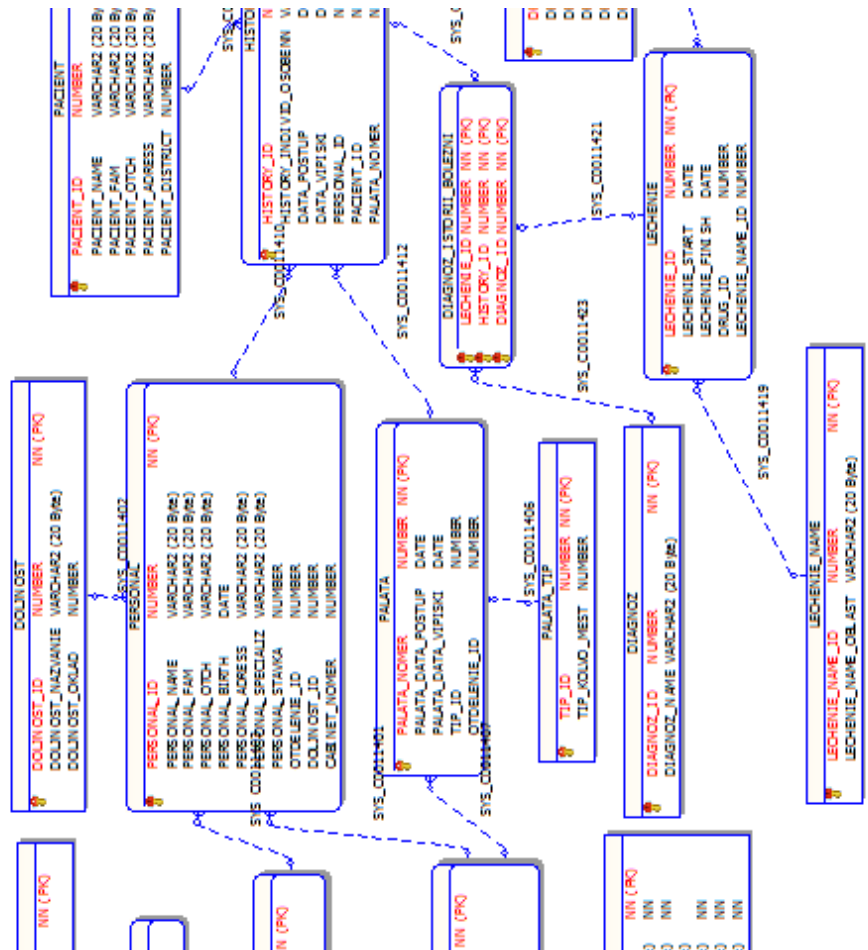


Рисунок 2.30 – Диаграмма физической модели данной базы данных  
**2.6.3 Создание пользователей, логинов и задание им паролей**

Создание пользователя admin:

```
create user Maksat identified by 1
default tablespace Hospital quota 40M on Hospital
temporary tablespace temp
Commit
```

Создание остальных пользователей

```
grant create user,alter user to Maksat with admin option
grant create role to Maksat with admin option
grant create session,alter session to Maksat with admin
option
grant create table to Maksat
grant create procedure to Maksat
grant create trigger to Maksat
grant create sequence to Maksat
grant create view to Maksat
commit
```

## 2.7 Установка и настройка Microsoft visual studio 2010

### 2.7.1 Установка Microsoft visual studio 2010

C# (произносится си шарп) – это объектно–ориентированный язык программирования, разработанный в 1998—2001 гг. группой инженеров. Руководитель Андерса Хейлсберга в компании Microsoft как язык для разработки приложений на платформе Microsoft.NET Framework и вскоре был стандартизирован как ECMA–334 и ISO/IEC 23270.

C# относится к группе языков с C синтаксисом, у C# синтаксис наиболее схож с языком C++ и Java. Он поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), имеет статическую типизацию, делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников – языков C++, Pascal, Модула, Smalltalk и в особенности Java – C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов).

Дистрибутив Microsoft visual studio 2010 можно скачать бесплатно для некоммерческого использования. Далее описывается установка и настройка MS Visual studio 2010. На рисунке 2.31 показано первое диалоговое окно установки приложения.

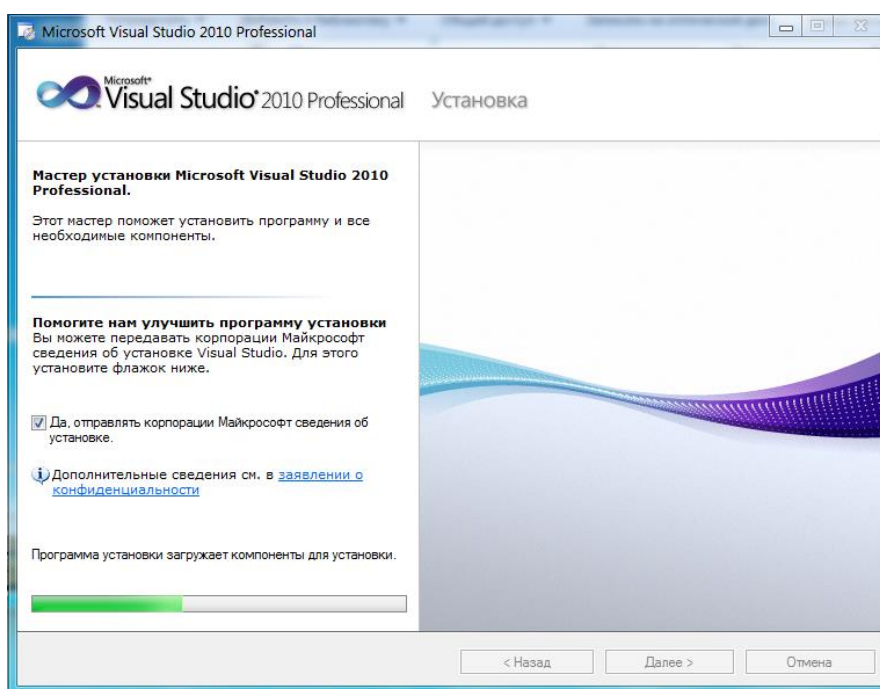


Рисунок 2.31 – Окно установки приложения

На рисунке 2.32 показано лицензионное соглашение для установки.

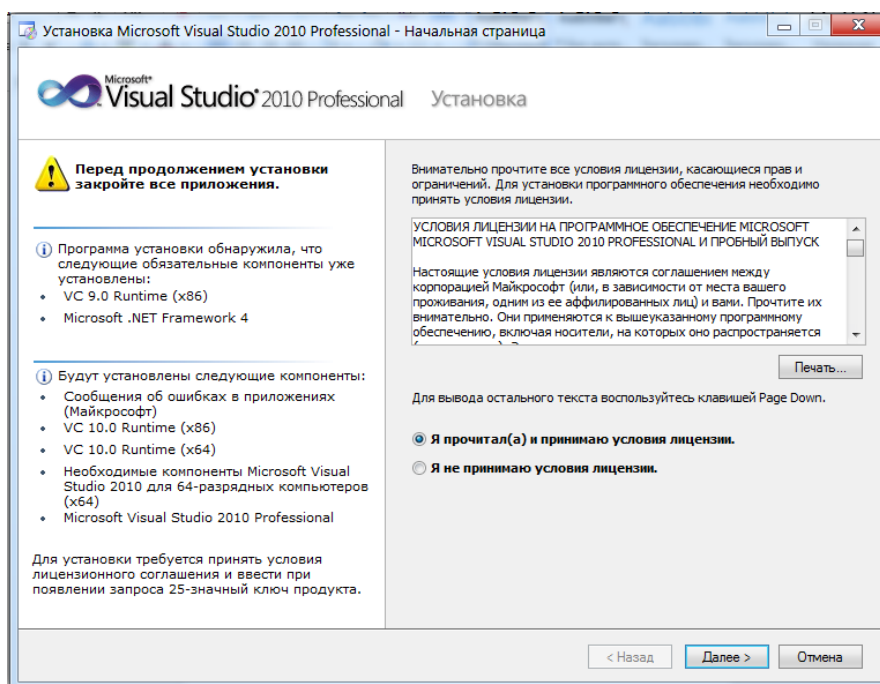


Рисунок 2.32 – Окно лицензионного соглашения

На рисунке 2.33 показан путь установки и тип устанавливаемого программного обеспечения.

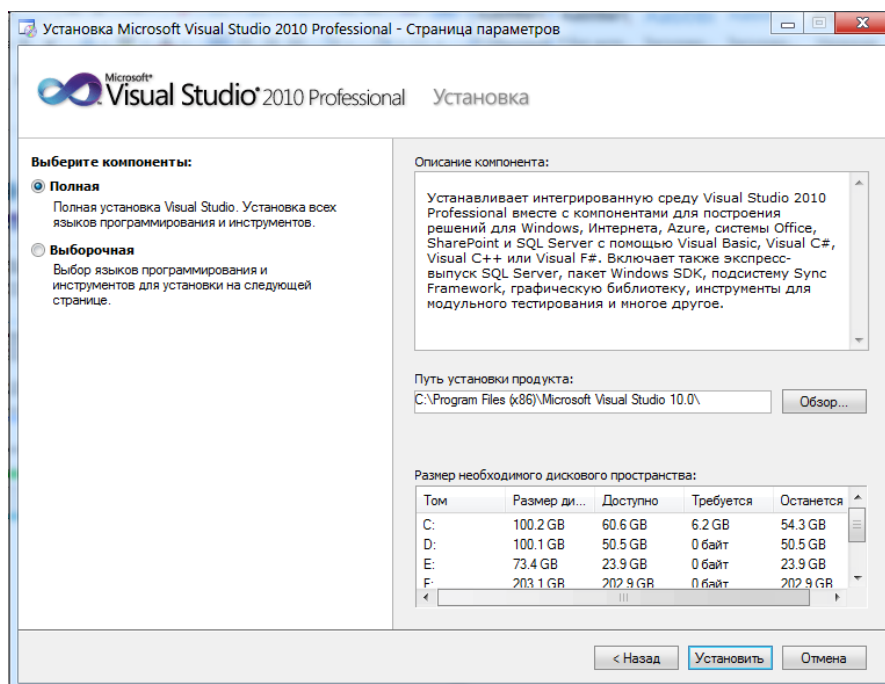


Рисунок 2.33 – Путь установки ПО

Процесс установки MS Visual Studio 2010 показан на рисунке 2.34.

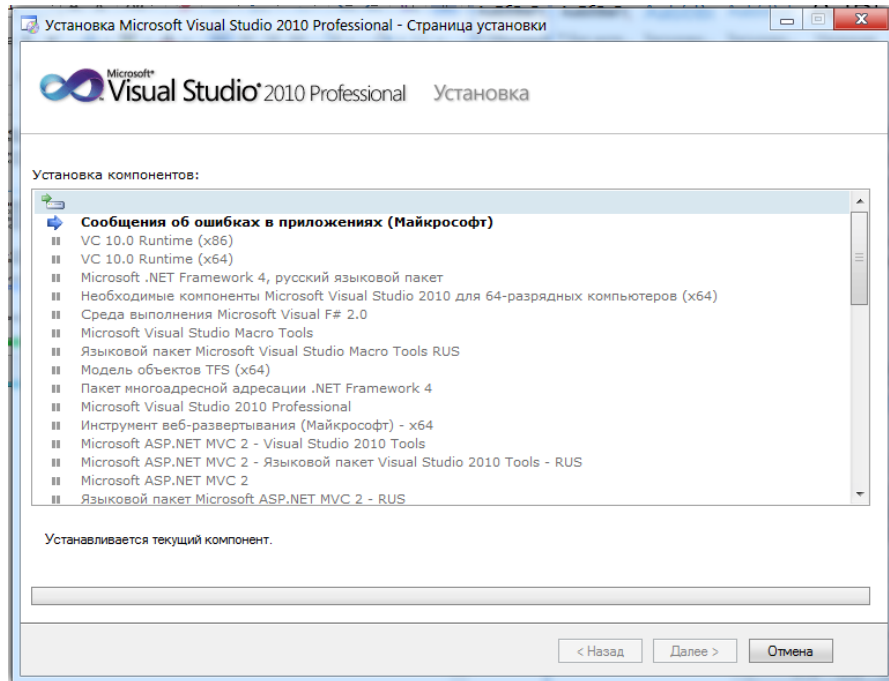


Рисунок 2.34 – Процесс установки ПО

## 2.7.2 Настройка MS visual studio 2010

Как показано на рисунке 2.35 выбираем пункт «Приложения Windows From Applications» для создания форм приложения.

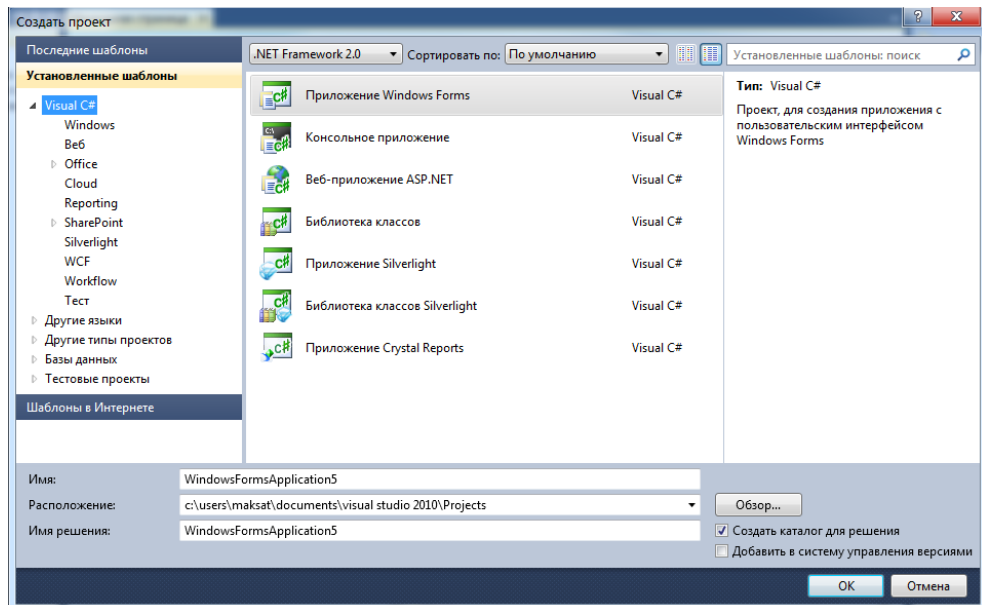


Рисунок 2.35 – Окно выбора типа приложения

Далее, как показано на рисунке 2.36 мы устанавливаем и загружаем дополнительные ссылки для полной работоспособности базы.

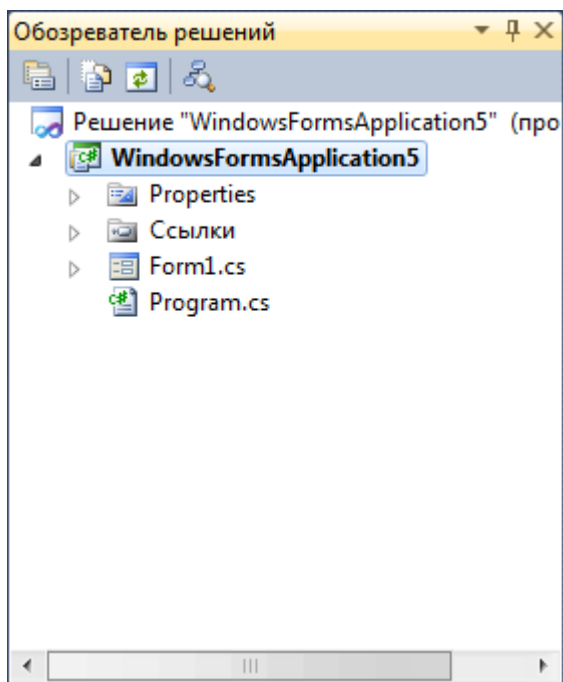


Рисунок 2.36 – Окно загрузки ссылок для Oracle DB

На рисунке 2.37 показано окно выбора подключения базы данных. В моем случае orcl.Maksat.

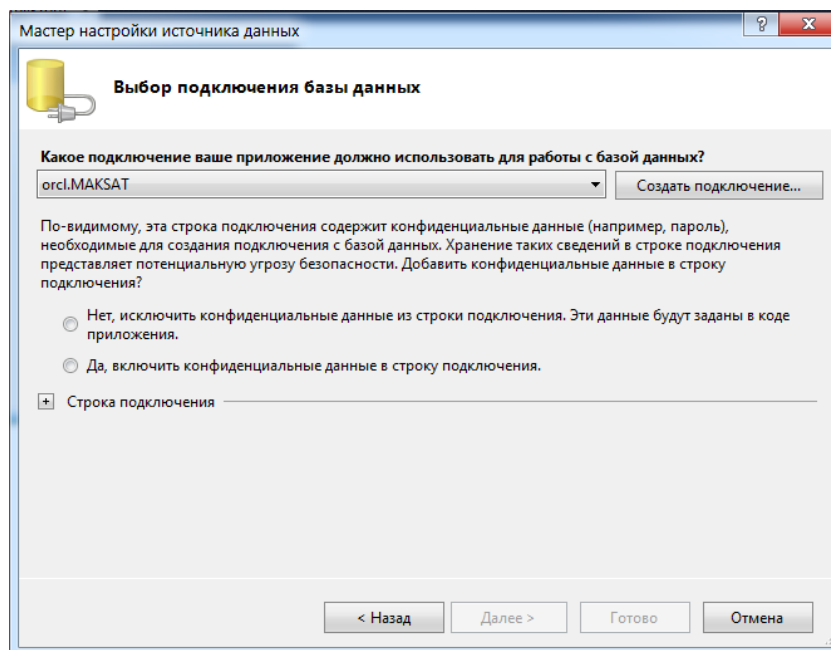


Рисунок 2.37 – Окно выбора подключения базы данных

Далее идет окно выбора объектов базы данных. В моей работе подключаются таблицы, представления, хранимые функции и процедуры, пакеты, триггеры как показано на рисунке 2.38.

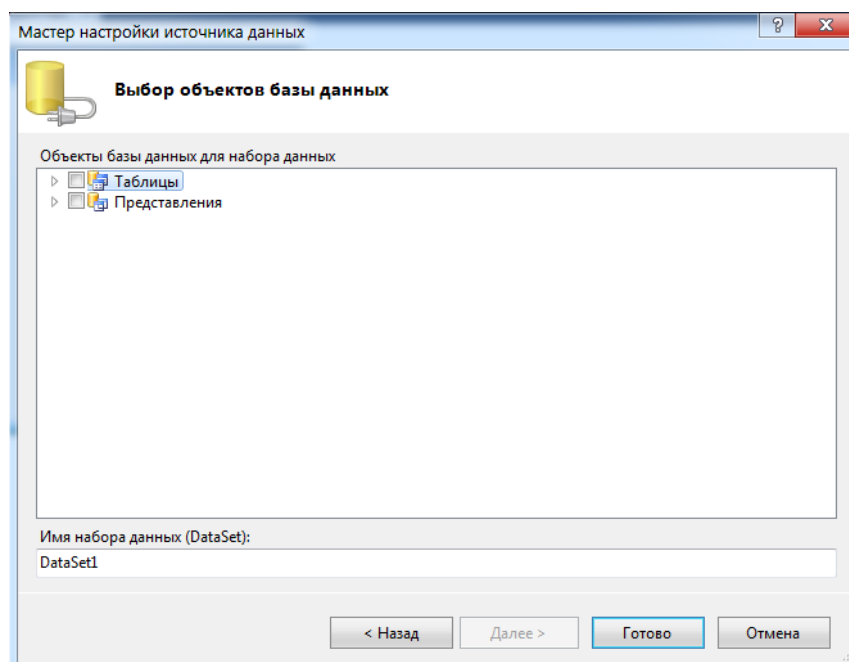


Рисунок 2.38 – Окно выбора объектов базы данных

На рисунке 2.39 показано окно входа в систему. Здесь, пользователь вводит свое имя пользователя и пароль. Система сама определяет его статус.

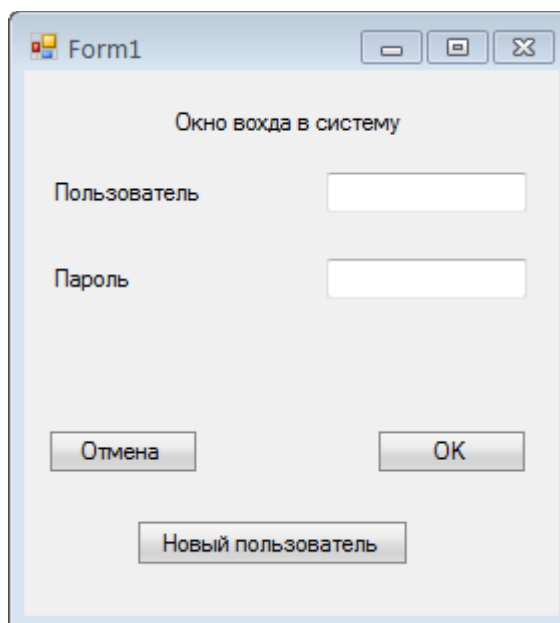
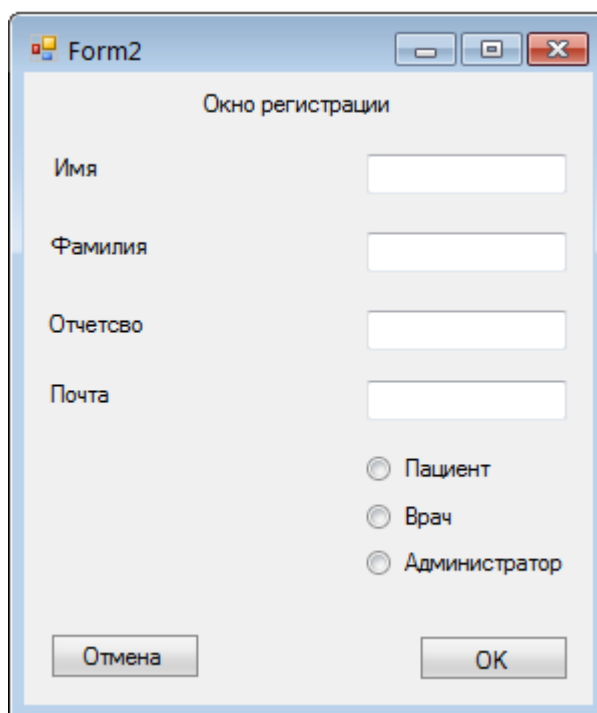


Рисунок 2.39 – Окно входа в систему

Если пользователь не зарегистрирован, то он может зарегистрироваться в систему, выбрав свой статус пользователя: врач, пациент или администратор, как показано на рисунке 2.40.



The image shows a standard Windows-style dialog box titled "Form2" with the subtitle "Окно регистрации". It features four text input fields labeled "Имя", "Фамилия", "Отчетсво", and "Почта". Below these fields are three radio button options: "Пациент", "Врач", and "Администратор". At the bottom of the dialog are two buttons: "Отмена" on the left and "ОК" on the right. The window has standard minimize, maximize, and close controls in the top right corner.

Рисунок 2.40 – Окно регистрации



### 3 технико–экономическое обоснование проекта

#### 3.1 Описание работы и обоснование необходимости

тема дипломной работы: «Разработка автоматизированного рабочего места сотрудника больницы города Алматы».

Цель данной работы – разработать автоматизированное рабочее место сотрудника больницы города Алматы. Разработка включает в себя организацию структуры базы данных, автоматизацию работы и построение веб приложения.

В данной работе будут решены такие задачи как:

- 1) Будет рассчитана смета затрат и цен программного обеспечения.
- 2) Определены объем и трудоемкость разработки программного обеспечения.
- 3) Рассчитаны затраты на разработку информационной технологии.
- 4) Будет рассчитана цена программного продукта.
- 5) Будет произведен расчет эксплуатационных затрат при использовании информационных технологий.

#### 3.2 Расчет затрат на разработку информационных технологий

Расчет полных затрат на разработку проектного решения в виде информационных технологий ( $C_{\text{пi}}$ ) осуществляется по формуле

$$C_{\text{пi}} = Z_{\text{фот}} + Z_{\text{сзи}} + M_{\text{i}} + P_{\text{ci}} + P_{\text{mi}} + P_{\text{нки}} + П_{\text{зи}} + P_{\text{ни}} \quad (3.1)$$

где  $Z_{\text{фот}}$ – общий фонд оплаты труда разработчиков, тенге;

$Z_{\text{сзи}}$ – отчисления по социальному налогу, тенге;

$M_{\text{i}}$ – затраты на материалы, тенге;

$P_{\text{ci}}$ – затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;

$P_{\text{mi}}$ – затраты, связанные с эксплуатацией техники, тенге;

$P_{\text{нки}}$  – затраты на научные командировки, тенге;

$П_{\text{зи}}$ – прочие затраты, тенге;

$P_{\text{ни}}$ – накладные расходы, тенге.

Размер фонда оплаты труда разработчиков ( $Z_{\text{фот}}$ ) рассчитывается по формуле

$$Z_{\text{фот}} = Z_{\text{oi}} + Z_{\text{ди}} \quad (3.2)$$

где  $Z_{\text{oi}}$ – основная заработная плата, тенге;

$Z_{дi}$  – дополнительная заработная плата, тенге.

Определение объема и трудоемкости разработки программного обеспечения

Общий объем ( $V_0$ ) программного продукта определяется исходя из количества и объема функции, реализуемых программой

$$V_0 = \sum_{j=1}^n V_i \quad (3.3)$$

где  $V_i$  – объем отдельной функции ПО;

$n$  – общее число функций.

$$V_0 \approx 27\ 000$$

На основании информации о функциях разрабатываемого ПО по каталогу функций определяется объем функций и общий объем ПО, который уточняется (корректируется) с учетом условий разработки ПО в организации. Уточненный объем ПО ( $V_y$ ) рассчитывается по формуле

$$V_y = \sum_{i=1}^n V_{yi} \quad (3.4)$$

Где  $V_{yi}$  – уточненный объем отдельной функции ПО (ЛОС).

$$V_y = 27\ 610$$

Общая трудоемкость небольших проектов рассчитывается по формуле

$$T_0 = T_n * K_c * K_m * K_n \quad (3.5)$$

где  $K_c$  – коэффициент, учитывающий сложность ПО;

$K_r$  – поправочный коэффициент, учитывающий степень использования при разработке стандартных модулей;

$K_n$  – коэффициент, учитывающий степень новизны ПО.

Нормативная трудоемкость ПО ( $T_n$ ) определяется на основе принятого в расчет  $V_y$  и категории сложности, которая уточняется с учетом сложности и новизны проекта и степени использования стандартных модулей при разработке.

Нормативная трудоёмкость ( $T_n$ ) служит основой для определения общей трудоемкости ( $T_0$ ), расчет которой осуществляется различными способами в зависимости от размера проекта.

Коэффициент сложности рассчитывается по формуле

$$K_c = 1 + \sum_{i=1}^n K_i \quad (3.6)$$

Где  $K_i$  – коэффициент, соответствующий степени повышения сложности ПО за счет конкретной характеристики

$n$  – количество учитываемых характеристик

$K_c=0.12$  так как у меня две характеристики

Поправочный коэффициент, учитывающий степень использования при разработке ПО стандартных модулей ( $K_m$ ), определяется удельным весом этих модулей в общем объеме проектируемого продукта.

В данной работе составляет от 40% до 60, следовательно, значение

$K_m= 0.6$

Поправочный коэффициент рассчитывающий новизну разрабатываемого ПО( $K_n$ ) определяется на основе данных представленных в таблице 3.1.

таблица 3.1 – поправочный коэффициент рассчитывающий новизну разрабатываемого ПО

Категория новизны	Степень новизны	Использование		Значение $K_n$
		На основе нового типа ПК	В среде новой ОС	
А	Принципиально новые ПО, не имеющие доступных аналогов	–	–	1,0

В моем случае для 2–ой категории сложности при объеме ПО равном 27 000 категория будет,  $t_n=566$ . Следовательно,  $t_o$  будет равно

$t_o= 566*0.12*0.6*1.0 = 40.752$  чел/дней

$$Ч = t_o / (T_p \times \Phi_{эф}) \quad (3.7)$$

где  $\Phi_{эф}$ – эффективный фонд времени работы одного работника в течение года (дн.);

$T_o$ – общая трудоемкость разработки проекта (чел./дн.);

$T_p$ – срок разработки проекта (лет).

Срок разработки проекта ( $T_p$ ) определяется по формуле

$$T_p = t_o / (Ч \times \Phi_{эф}) \quad (3.8)$$

Эффективный фонд времени работы одного работника ( $\Phi_{эф}$ ) рассчитывается по формуле

$$\Phi_{\text{эф}} = D_{\text{г}} - D_{\text{п}} - D_{\text{в}} - D_{\text{о}} \quad (3.9)$$

где  $D_{\text{г}}$  – количество дней в году;  
 $D_{\text{п}}$  – количество праздничных дней в году;  
 $D_{\text{в}}$  – количество выходных дней в году;  
 $D_{\text{о}}$  – количество дней отпуска.

так как  $D_{\text{г}} = 365$ ;  $D_{\text{п}} = 14$ ;  $D_{\text{в}} = 104$ ;  $D_{\text{о}} = 24$ , то по формуле (3.9) получим

$$\Phi_{\text{эф}} = 365 - 14 - 104 - 24 = 223 \text{ дней}$$

Плановое число разработчиков  $Ч_{\text{р}} = 1$ , следовательно, по формуле (3.8)

$$T_{\text{р}} = 40.752 / (1 * 223) = 0.18 \text{ лет}$$

$$0.18 * 365 * 24 = 1576.8 \text{ часов}$$

так как нам известны все данные то по формуле (3.7) получаем

$$Ч = 40.752 / (0.18 * 223) = 1 \text{ чел.}$$

Основная заработная плата исполнителей на конкретное ПО рассчитывается по формуле

$$Z_{oi} = \sum_{i=1}^n T_{\text{чи}} \times T_{\text{ч}} \times \Phi_{\text{п}} \times K \quad (3.10)$$

где  $n$  – количество исполнителей, занятых разработкой конкретного ПО;  
 $T_{\text{чи}}$  – часовая тарифная ставка  $i$ -го исполнителя (тыс. тенге);  
 $\Phi_{\text{п}}$  – плановый фонд рабочего времени  $i$ -го исполнителя (дней) = 22 раб. дней в месяце;  
 $T_{\text{ч}}$  – количество часов работы в день (час) = 8 часов;  
 $K$  – коэффициент премирования = 1.25.

По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов–исполнителей, участвующих в разработке ПО, с определением образования, специальности, квалификации и должности (таблица 3.2).

таблица 3.2 – Сведения по работникам, задействованным в проекте

Специалист – Исполнитель	Количество, человек	Заработная плата в месяц, тенге
Программист	1	900 000

Итого	1	900 000
-------	---	---------

Размер фонда оплаты труда разработчиков ( $Z_{\text{Фот}}$ ) рассчитывается по формуле

$$Z_{\text{Фот}} = Z_{oi} + Z_{di} \quad (3.11)$$

где  $Z_{oi}$  – основная заработная плата, тенге;

$Z_{di}$  – дополнительная заработная плата, тенге.

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки на установленную при 40-часовой недельной норме рабочего времени расчетную среднемесячную норму рабочего времени в часах ( $\Phi_p$ ).

$$T_{\text{ч}} = \frac{T_{\text{м}}}{\Phi_p} \quad (3.12)$$

где  $T_{\text{ч}}$  – часовая тарифная ставка (тыс.тенге);

$T_{\text{м}}$  – месячная тарифная ставка (тыс.тенге).

По формуле 3.12 можно определить часовую тарифную ставку исполнителя

$$T_{\text{ч}} = \frac{900\,000}{176} = 5113.63 \text{ тенге в час}$$

По формуле (3.10) можно рассчитать основную заработную плату исполнителей:

$$Z_{oi} = 5113.63 * 1576.8 * 1.25 = 10078964.73 \text{ тенге}$$

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

$$Z_{di} = Z_{oi} * N_d / 100 \quad (3.13)$$

где  $N_d$  – коэффициент дополнительной заработной платы разработчиков и равна 18%.

$$Z_{di} = 10078964.73 * 0,18 = 1814213.65 \text{ тенге}$$

$$Z_{\text{фот}} = 10078964.73 + 1814213.65 = 11893178.38 \text{ тенге}$$

Социальный налог составляет 11% (ст. 358 п. 1 НК РК) от дохода работника, и рассчитывается по формуле

$$Z_{\text{сзи}} = (\text{Фот} - \text{ПО}) * 11\% \quad (3.14)$$

где ПО – пенсионные отчисления, которые составляют 10% от Фот и социальным налогом не облагаются

$$ПО = \Phi O_T \times 10\% \quad (3.15)$$

По формулам (21) и (20) получается

$$ПО = 10078964.73 \times 0,1 = 1007896,47 \text{ тенге}$$

$$З_{сзi} = (10078964.73 - 1007896,47) \times 0,11 = 997817.5 \text{ тенге}$$

Величина затрат на материалы на основании исходных данных определяется по формуле

$$M_i = (З_{осн.} \times H_{мз}) / 100\% \quad (3.16)$$

где  $H_{мз}$  – норма расхода материалов от основной заработной платы (3–5%).

$$M_i = 10078964.73 \times 0.04 = 403158.59 \text{ тенге}$$

Первоначальная стоимость оборудования одного ПК с периферией указана в таблице 3.3.

таблица 3.3 – Первоначальная стоимость оборудования одного ПК с периферией

Наименование	Цена, тенге
Ноутбук	115 000
ID Net	9 000
Итого:	124 000

$$P_{ci} = 124\ 000 \text{ тенге}$$

Расходы по статье «Машинное время» ( $P_{mi}$ ) включают оплату машинного времени, необходимого для разработки и отладки ПО, которое определяется по нормативам (в машино-часах) на 100 строк исходного кода ( $H_{мв}$ ) машинного времени в зависимости от характера решаемых задач и типа

$$P_{mi} = Ц_{mi} \times (V_{oi}/100) \times H_{мв} \quad (3.17)$$

где  $Ц_{mi}$  – цена одного машино-часа (тыс. тенге);

$V_{oi}$  – общий объем ПО (строк исходного кода);

$H_{MB}$  – норматив расхода машинного времени на отладку 100 строк исходного кода (машино-часов) = 12.

Учитывая формулу (3.17) получим

$$P_{Mi} = 5113,63 * (27\ 000 / 100) * 12 = 16\ 568\ 161,2 \text{ тенге}$$

Расходы по статье «Научные командировки» ( $P_{HKi}$ ) на конкретное ПО определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате

$$P_{HKi} = Z_{oi} * H_{PHK} / 100 \quad (3.18)$$

Где  $H_{PHK}$  норматив расходов на командировки в целом по организации в (%), в дипломной работе нужно брать 30%.

Рассчитываем значения по формуле (4.18) получаем

$$P_{HKi} = 10\ 078\ 964,73 * 0,3 = 3\ 023\ 689,42 \text{ тенге}$$

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

$$P_{zi} = Z_{oi} * H_{Pz} / 100 \quad (3.19)$$

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

Рассчитываем значения по формуле (3.19) получаем

$$P_{zi} = 10\ 078\ 964,73 * 0,2 = 2\ 015\ 792,94 \text{ тенге}$$

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

$$P_{ni} = Z_{oi} * H_{PH} / 100\% \quad (3.20)$$

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

$H_{PH}$  – норматив накладных расходов в целом по организации в (%), в дипломной работе нужно брать 70%.

Рассчитываем значения по формуле (3.20) получаем

$$P_{ni} = 10078964,73 * 0,7 = 7055275,3 \text{ тенге}$$

Рассчитываем полные затраты на разработку программного обеспечения по формуле (3.1).

$$C_{pi} = 11\,893\,178,38 + 997\,817,5 + 403\,158,59 + 124\,000 + 16\,568\,161,2 + 3\,023\,689,42 + 7\,055\,275,3 + 2\,015\,792,94 = 41\,781\,073,34 \text{ тенге}$$

Результаты выполненных расчетов представлены в таблице 3.4.

таблица 3.4 – Затраты на разработку программного обеспечения

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{фот}$	11 893 178,38	28,5
Социальный налог	$Z_{сzi}$	997 817,5	2,4
Материалы	$M_i$	403 158,59	1,0
Спецоборудование	$P_{ci}$	124 000	0,3
Машинное время	$P_{mi}$	16 568 161,2	39,7
Научные командировки	$P_{nki}$	3 023 689,42	7,2
Прочие затраты	$P_{zi}$	2 015 792,94	4,8
Накладные расходы	$P_{ni}$	7 055 275,3	16,9
Итого:		41 781 073,34	100

Результаты выполненных расчетов представлены на структурной диаграмме.

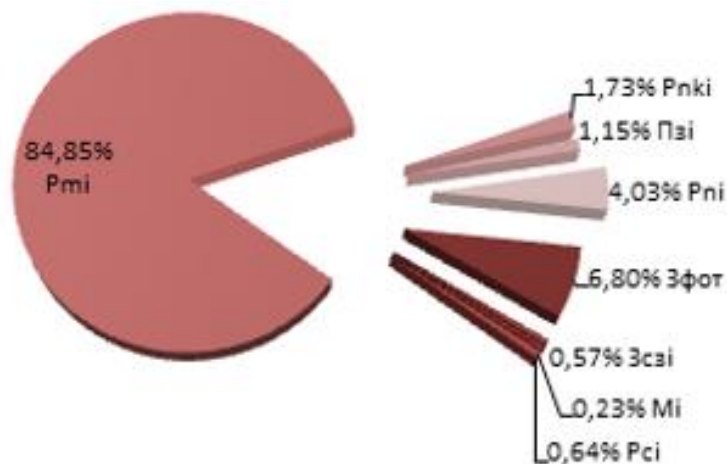


Рисунок 3.2 – Структура себестоимости программного продукта



### 3.3 Расчет цены программного продукта

Расчет цены ПП в организационно–экономической части дипломной работы предлагается производить следующим образом:

– если ПП разработан одной организацией по заказу другой и не предназначен для тиражирования, то затраты на разработку ПП считаются его себестоимостью, и при формировании цены применяется затратный метод;

– если ПП предназначен для тиражирования, то конечная цена определяется путем экспертных оценок на основании ценностного подхода с учетом текущих цен конкурентов (если существуют аналогичные ПП).

Расчет цены ПП, который разработан одной организацией по заказу другой и не предназначен для тиражирования, осуществляется по формуле

$$C_{ПП} = Z_{РПР} + \Pi_n + НДС \quad (3.21)$$

где  $C_{ПП}$  – цена программного продукта, тенге;

$Z_{РПР}$  – затраты на разработку проектного решения, в данном случае программного продукта, тенге;

$\Pi_n$  – планируемая прибыль, тенге;

$НДС$  – налог на добавленную стоимость, тенге.

Планируемая прибыль рассчитывается по формуле

$$\Pi_n = Z_{РПР} + R_{НПП} \quad (3.22)$$

где  $R_{НПП}$  – нормативная рентабельность ПП, определяемая организацией.

$НДС$ , начисленный на ПП, определяется следующим образом

$$НДС = (Z_{РПР} + \Pi_n) * k_{НДС} \quad (3.23)$$

где  $k_{НДС}$  – ставка налога на добавленную стоимость.

По формуле (3.22) учитывая, что  $Z_{РПР} = C_{pi}$ ,  $R_{НПП}$  – это процент рентабельности по отношению к себестоимости составляет 20.3%

$$R_{НПП} = C_{pi} * 0,203 = 19\,863\,536,2 * 0,203 = 4\,032\,297,85 \text{ тенге}$$

$$\Pi_n = 19\,863\,536,2 + 4\,032\,297,85 = 20\,266\,765,98 \text{ тенге}$$

По формуле (3.23) получаем

$$НДС = (19\,863\,536,2 + 20\,266\,765,98) * 0,12 = 7142307,75 * 0,12 = 4815636 \text{ тенге}$$

По формуле (3.21) получаем

$$C_{\text{пп}} = 19863\ 536.2 + 20\ 266\ 765.98 + 4\ 815\ 636 = 44\ 945\ 938 \text{ тенге}$$

### Вывод

Стоимость разработки составила 44 945 938 тенге.

Наибольшую долю затрат составляет машинное время 16 568 161,2 тенге, его доля составила 83,41%.

Данная работа не преследует прибыли, главная цель автоматизация работы в больнице города Алматы. Отчеты по данной работе представлены в таблице 3.5, в которой представлены все затраты на разработку ПО, выразив их значения в числовой форме, а также разделил на доли в проекте.

т а б л и ц а 3.5 – Затраты на разработку программного обеспечения

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{\text{ФОТ}}$	11 893 178,38	28,5
Социальный налог	$Z_{\text{сзи}}$	997 817,5	2,4
Материалы	$M_i$	403 158,59	1,0
Спецоборудование	$P_{\text{си}}$	124 000	0,3
Машинное время	$P_{\text{ми}}$	16 568 161,2	39,7
Научные командировки	$P_{\text{нки}}$	3 023 689,42	7,2
Прочие затраты	$P_{\text{зи}}$	2 015 792,94	4,8
Накладные расходы	$P_{\text{ни}}$	7 055 275,3	16,9
Итого:		41 781 073,34	100

## **4 Безопасность жизнедеятельности**

### **4.1 Анализ потенциально опасных и вредных факторов, воздействующих на обслуживающий персонал при эксплуатации технического оборудования**

Главной целью данного проекта является разработка программы автоматизированного рабочего места сотрудника каждой больницы города Алматы, с целью усовершенствования рабочей среды и ускорения работы с клиентами. В данной работе используются следующие технологии: Oracle Database Express Edition 11g Release 2, приложения написанное на JAVA для установки связи сотрудника одного из больниц г. Алматы и пациента, желающего записаться на прием, PLSQLDeveloper.

В настоящее время все предприятия, учреждения или организации не могут функционировать достаточно эффективно без использования компьютерной техники. Постоянное развитие любого предприятия, учреждения или организации, а как следствие объёмов и сложности информации требует расширения компьютерных сетей и автоматизированных информационных систем. Но кроме очевидных выгод компьютерная техника несет в себе опасность здоровью и поэтому актуальной становится проблема охраны труда человека в процессе работы, сохранение его здоровья и работоспособности.

Существует несколько вредных факторов, воздействующих на работников, занятых на работе с видеодисплейными терминалами (ВДТ) и персональными компьютерами (ПК):

- 1 воздействие электромагнитных полей (радиочастот), статического электричества;
- 2 неудовлетворительный микроклимат помещений;
- 3 недостаточная освещенность;
- 4 психоэмоциональное напряжение.

Без строгого учёта правил техники безопасности и производственной санитарии, неточного выполнения требований техники безопасности может привести к аварии, либо к профессиональным заболеваниям и производственному травматизму. Охрана труда обеспечивается системой законодательных актов, социально-экономических, организационных, технических, гигиенических и лечебно-профилактических мероприятий и средств, направленных на создание таких условий труда, при которых исключено воздействие на работающих опасных и вредных производственных факторов. Создание наиболее благоприятных, комфортных условий труда, улучшение охраны труда и техники безопасности, без сомнения, ведет к более высокой производительности труда, социальному развитию и повышению благосостояния.

Согласно ГОСТ 12.1.005–88 ССБт «Оптимальные и допустимые нормы микроклимата, в зависимости от категории работ», работа людей в помещении

относится к работе лёгкой тяжести, так как управление оборудованием осуществляется дистанционно с помощью компьютеров.

С целью создания нормальных условий для работников предприятий связи установлены нормы производственного микроклимата. В помещениях при работе с ЭВМ должны соблюдаться следующие климатические условия:

1) Холодный период года:

– оптимальная температура 22–24 С°, допустимая температура 18–26 С°;

– относительная влажность 40–60 %, допустимая влажность 75%;

– скорость движения воздуха относительная и допустимая 0,1 м/с.

2) тёплый период года:

– оптимальная температура 23–25 С°, допустимая температура 20–30 С°;

– относительная влажность 40–60 %, допустимая влажность 55%;

– скорость движение воздуха относительная 0,1 м/с и допустимая 0,1–0,2 м/с.

## 4.2 Планировка рабочего места

Эргономика – прикладная наука целью, которой является приспособление труда к физиологическим и психическим возможностям человека для обеспечения наиболее эффективной работы, которая не создаёт угрозы здоровью человека.

Практика показывает, что планировка рабочего места должна удовлетворять требованиям удобства выполняемых работ и экономии энергии, и времени оператора, рационального использования производственных площадей и удобства обслуживания устройств ПК.

При планировке рабочего места необходимо учитывать удобство расположения дисплеев, принтеров, пульта ПК, а также зоны досягаемости рук оператора. Эти зоны, установленные на основании антропометрических данных тела человека, дают возможность рационально разместить компьютер, его клавиатуру и дисплей.

Высота рабочей поверхности стола должна регулироваться в пределах 680–800 мм; при отсутствии такой возможности должна составлять 725 мм.

Дисплей должен удовлетворять следующим требованиям:

1 важнейшие элементы конструкции должны быть расположены в центре поля зрения (клавиатура);

2 элементы должны быть сгруппированы по функциональному признаку;

3 рабочие поверхности должны быть расположены наклонно, по возможности перпендикулярно взгляду оператора;

4 экран видеомонитора должен находиться от глаз пользователя на оптимальном расстоянии 600–700 мм, но не ближе 500 мм с учётом размеров знаков и символов.

Важнейшими характеристиками зрительного восприятия оператора являются: яркость, контрастность между объектами и фоном, и острота зрения. Контрастность по отношению к фону влияет на восприятие цветов. так, например, лучше воспринимаются комбинации цветов: черный на желтом, черный на белом, зеленый на черном, белый на черном. Отсюда следует оптимальность выбора цветов:

- для экрана: белый на черном;
- для клавиатуры: черный на белом.

Наиболее удобно сиденье, имеющее выемку, соответствующую форме бедер и наклон назад. Спинка стула должна быть изогнутой формы, обнимающей поясницу. Рабочий стул (кресло) должен быть снабжен подъёмно-поворотным механизмом, обеспечивающим регулицию высоты сидения и спинки. Рабочее кресло должно иметь подлокотники. Регулировка каждого параметра должна легко осуществляться, быть независимой и иметь надёжную фиксацию. На рабочем месте необходимо предусматривать подставку для ног.

Клавиатура должна располагаться на поверхности стола таким образом, чтобы соответствовать локтю сидящего оператора. Его рука должна быть согнута на 90 градусов в локтевом суставе, а предплечье – лежать горизонтально.

Правильное расположение персонального компьютера и пользователя изображен на рисунке 4.1.

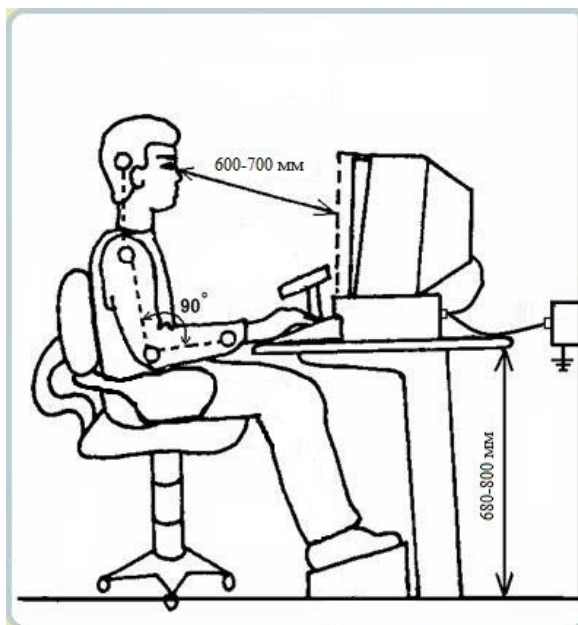


Рисунок 4.1 – План рабочего места на ПК

Отдел внедрения данной системы состоит из 1 человека.

Помещение имеет размеры: длина (L) = 6.5 метров, ширина (B) = 4.5 метра, высота (H) = 4 метра. Помещение находится в здании на 1-м этаже.

Согласно санитарным требованиям от 01.12.2011 года площадь на одно рабочее место пользователей ПК и Вт на базе электронно–лучевой трубки (далее – ЭлТ) составляет не менее 6 квадратных метров. Общая площадь рабочего помещения удовлетворяет санитарным требованиям 87,75 м<sup>2</sup>, что на 81,75 м<sup>2</sup> больше.

### 4.3 Расчет вентиляции помещения

В помещении, где находятся ЭВМ, системы отопления и системы кондиционирования следует устанавливать так, чтобы ни теплый, ни холодный воздух не направлялся на людей. На производстве рекомендуется создавать динамический климат с определенными перепадами показателей. температура воздуха у поверхности пола и на уровне головы не должна отличаться более, чем на 5 градусов. В производственных помещениях помимо естественной вентиляции предусматривают приточно–вытяжную вентиляцию. Основным параметром, определяющим характеристики вентиляционной системы, является кратность обмена, т.е. сколько раз в час сменится воздух в помещении.

$V_{\text{вент}}$  – объем воздуха, необходимый для обмена;

$V_{\text{пом}}$  – объем рабочего помещения.

Для расчета примем следующие размеры рабочего помещения (Рисунок 4.2).

Длина А = 6.5 м;

Ширина В = 4.5 м;

Высота Н = 3 м.

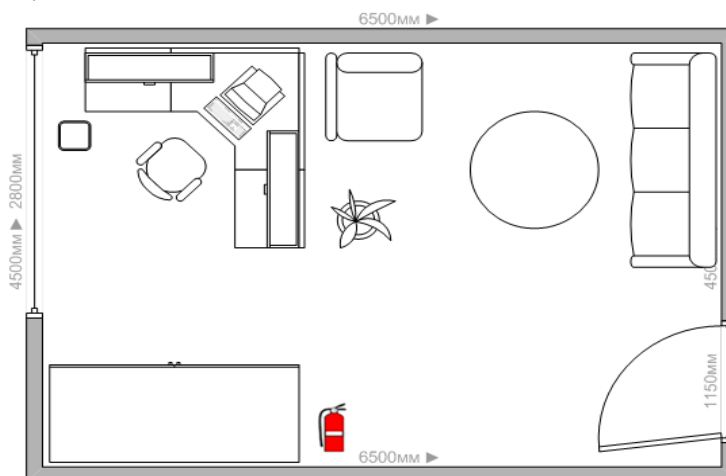


Рисунок 4.2 – План рабочего помещения

Соответственно объем помещения равен

$$V_{\text{пом}} = A * B * H = 87,75 \text{ м}^3$$

Необходимый для обмена объем воздуха  $V_{\text{вент}}$  определим исходя из уравнения теплового баланса

$$V_{\text{вент}} * C * (t_{\text{уход}} - t_{\text{приход}}) * Y = 3600 * Q_{\text{избыт}}$$

$Q_{\text{избыт}}$  – избыточная теплота (Вт);

$C = 1000$  – удельная теплопроводность воздуха (Дж/кгК);

$\gamma = 1.2$  – плотность воздуха (мг/см).

температура уходящего воздуха определяется по формуле

$$t_{\text{уход}} = t_{\text{р.м.}} + (H - 2) * t$$

где,  $t = 1-5$  градусов – превышение  $t$  на 1м высоты помещения;

$t_{\text{р.м.}} = 25$  градусов – температура на рабочем месте;

$H = 3$  м – высота помещения;

$t_{\text{приход}} = 18$  градусов.

$$t_{\text{уход}} = 25 + (3 - 2) * 2 = 27$$

$$Q_{\text{избыт}} = Q_{\text{изб.1}} + Q_{\text{изб.2}} + Q_{\text{изб.3}}$$

где  $Q_{\text{изб.}}$  – избыток тепла от электрооборудования и освещения.

$$Q_{\text{изб.1}} = E * P$$

где,  $E$  – коэффициент потерь электроэнергии на теплоотвод ( $E=0.55$  для освещения);

$P$  – мощность,  $P = 60 \text{ Вт} * 5 = 300 \text{ Вт}$ .

$$Q_{\text{изб.1}} = 0.55 * 300 = 165 \text{ Вт}$$

$Q_{\text{изб.2}}$  – теплопоступление от солнечной радиации,

$$Q_{\text{изб.2}} = m * S * k * Q_c$$

где  $m$  – число окон, примем  $m = 1$ ;

$S$  – площадь окна,  $S = 2.8 * 2 = 5,6 \text{ м}^2$ ;

$K$  – коэффициент, учитывающий остекление. Для двойного остекления

$k = 0.6$ ;

$Q_c = 127 \text{ Вт/м}$  – теплопоступление от окон.

$$Q_{\text{изб.2}} = 5,6 * 1 * 0,6 * 127 = 426,7 \text{ Вт}$$

$Q_{\text{изб.3}}$  – тепловыделения людей

$$Q_{\text{изб.3}} = n * q,$$

где,  $q = 80 \text{ Вт/чел.}$ ,

$n$  – число людей, например,  $n = 2$

$$Q_{\text{изб.3}} = 2 * 80 = 160 \text{ Вт}$$

$$Q_{\text{избыт}} = 165 + 426,7 + 160 = 751,7 \text{ Вт}$$

Из уравнения теплового баланса следует

$$V_{\text{вент}} = 3600 * 751,7 / (1000 * (27-18)) = 300,7 \text{ м}^3$$

Необходимо тщательно продумать месторасположение кондиционера в офисе. Можно установить канальный кондиционер за подвесным потолком и развести воздух в разные точки комнаты через воздуховоды. Это обеспечит равномерное распределение воздуха и температуры. Если высота подшивных потолков не позволяет установить канальный кондиционер (как в данном случае), можно предусмотреть два или даже три внутренних блока, расположенных в разных точках помещения. Такой вариант особенно оправдан в комнатах неправильной или вытянутой формы. Полупромышленные кондиционеры допускают подсоединять до трех внутренних блоков разного вида к одному наружному блоку. Это снизит стоимость всей системы и сохранит стену здания от множества блоков.

#### 4.4 Анализ пожарной безопасности

Согласно СНиП 2.04.09–84 здание по степени опасности развития пожара, от функционального назначения и пожарной нагрузки горючих материалов, относится к 1–ой группе категории D.

Причинами возникновения пожара могут быть:

- Возгорание элементов аппаратуры.
- Возгорание отделочных материалов от неисправных выключателей, розеток.
- Несоблюдение режимов эксплуатации оборудования, неправильное действие персонала.

При возникновении пожара может пострадать не только помещение, но и дорогостоящая аппаратура, привести к человеческим жертвам. Поэтому необходимо чтобы были приняты меры по раннему выявлению и ликвидации пожаров. Источниками зажигания могут оказаться электронные схемы ЭВМ, приборы, применяемые для технического обслуживания, устройства электропитания, кондиционеры воздуха, где в результате различных нарушений образуются перегретые элементы, и др.

В соответствии с требованиями правил пожарной безопасности помещение оборудованы углекислотными огнетушителями ОУ–5 с учетом – один огнетушитель на 100 м<sup>2</sup>. Общая площадь помещения управления составляет 24 м<sup>2</sup> таким образом устанавливаются 1 огнетушитель. В качестве огнетушащего вещества применяется комбинированный углекислотно–хладоновый состав. Расчетная масса комбинированного углекислотно–хладонового состава, кг, для объемного пожаротушения определяется по формуле

$$m_d = k \cdot g_n \cdot V$$



где  $k = 1,2$  – коэффициент компенсации не учитываемых потерь углекислотно–хладонового состава,  
 $gn = 0,04$  – нормативная массовая концентрация углекислотно–хладонового состава,  
 $V$  – объем помещения,

$$V = A \cdot B \cdot H$$

где,  $A = 6,5$  м – длина помещения,  
 $B = 4,5$  м – ширина помещения,  
 $H = 3$  м – высота помещения.

$$\text{тогда } V = 6 \cdot 4 \cdot 3 = 87,75 \text{ м}^3$$

$$m_d = 1,2 \cdot 0,04 \cdot 87,75 \approx 4,2 \text{ кг}$$

Расчетное число баллонов  $\xi$  определяется из расчета вместимости в 20–литровый баллон 4,2 кг углекислотно–хладонового состава.

Внутренний диаметр магистрального трубопровода  $d_i$ , мм, определяется по формуле

$$d_i = 12 \cdot \sqrt{2} = 17 \text{ мм}$$

Эквивалентная длина магистрального трубопровода  $l_2$ , м, определяется по формуле

$$l_2 = k_1 \cdot l$$

где  $k_1 = 1,2$  – коэффициент увеличения длины трубопровода для компенсации не учитываемых местных потерь,

$l = 3$  м – длина трубопровода по проекту тогда,

$$l_2 = 1,2 \cdot 3 = 3,6$$

Расход углекислотно–хладонового состава  $Q$ , кг/с, в зависимости от эквивалентной длины и диаметра трубопровода равна 1,4 кг/с

Расчетное время подачи углекислотно–хладонового состава  $t$ , мин, определяется по формуле

$$t = \frac{m_d}{60Q} = \frac{4,2}{60 \cdot 1,4} = 0,05 \text{ мин}$$

Масса основного запаса углекислотно–хладонового состава  $m$ , кг, определяется по формуле

$$m = 1.1 \cdot m_d \cdot \left( 1 + \frac{k_2}{k} \right)$$

где  $K_2=0.2$  – коэффициент учитывающий остаток углекислотно–хладонового состава в баллонах и трубопроводах

$$m = 1.1 \cdot 4.2 \cdot \left( 1 + \frac{0.2}{1.2} \right) = 5.39 \text{ кг}$$

таким образом из полученных результатов можно сделать вывод, что для обеспечения нормального функционирования системы автоматического пожаротушения потребуется 1 баллон углекислотно–хладонового состава вместимостью 20 литров, с массой смеси 3,5 кг. Автоматические установки газового пожаротушения имеют устройства для автоматического пуска в соответствии с ГОСТ 12,4.009–83.

#### **4.5 Вентиляция помещения**

Норма пространства помещения на одного человека составляет 6 м<sup>2</sup>. В офисе предполагается наличие воздуховодов, которые размещены за подвесными потолками. Воздух плавно подается и удаляется из офиса либо через вентиляционные решетки в стенах, либо через специальные розетки, установленные на потолке. также установлен кондиционер, который освежает воздух.

Можно утверждать, что система вентиляции соответствует нормам, следовательно мер по реконструкции вентиляции помещения применять не нужно.

#### **Вывод**

В данном разделе был произведён анализ условий труда в рабочем помещении. Уровень условий труда признан допустимым, и данные, полученные из расчетов полностью удовлетворяют требованиям стандартов безопасности жизнедеятельности.

также мы рассчитали все необходимые параметры для кондиционирования воздуха в помещении, т.е. автоматическое поддержание его состояния в помещении в соответствии с определенными требованиями независимо от изменения состояния наружного воздуха и условий в самом помещении.

Электротехническое оборудование в помещении является потенциальным источником возникновения пожара.



## **Заключение**

В данной дипломной работе мною было спроектировано и разработано автоматизированное рабочее место сотрудника больницы города Алматы, которое в значительной мере упростит посещаемость, внутреннюю структуру больницы, а также отношение пациента с врачом. Система создана гибкой и расширяемой, нет никаких сложностей, чтобы расширить ее или доработать. Возможность включения мультимедиа материалов и видеоконференции, пересылка сообщениями. Встроенные системы общения и взаимодействия снижают постороннюю нагрузку на пациентов–пользователей и врачей–пользователей.

В качестве технологий для хранения всей информации является: Oracle database 11g release 2. Приложение было разработано на Microsoft Visual Studio 2010 используя язык программирования С#.

Строение и дизайн выполнены в современном стиле, что будет легко в освоении новыми пользователями. Простота построения базы позволит администратору сопровождающему данную систему без лишних затрат вникнуть в саму систему.

В разработки использовались новейшие технологии. Система отлично работает как на компьютерах, так и на любых современных мобильных устройствах, дизайн автоматически адаптируется к разрешению экрана. Что позволит расширить круг пользователей не только с персональным компьютером, но и со смартфонами и планшетами.

## Список литературы

- 1 Сайт [http://scholar.urf.ac.ru/ped\\_journal/numero4/pedag/tsit3.html.ru](http://scholar.urf.ac.ru/ped_journal/numero4/pedag/tsit3.html.ru)
- 2 Адам Ф. jQuery для профессионалов = Pro jQuery. – 3-е изд. – СПб: Питер, 2012. – 689 с.
- 3 Сайт [http://dic.academic.ru/dic.nsf/fin\\_enc/19812](http://dic.academic.ru/dic.nsf/fin_enc/19812)
- 4 Ганник Дж. MSSQL 2008, Карманный справочник. – 2-е изд. – СПб: Питер, 2009. – 389 с.
- 5 Сайт <http://getbootstrap.com/css/>
- 6 Роб П., Корнел К. Системы баз данных: проектирование, реализация и управление. – 5-е изд. – СПб.: БХВ–Петербург, 2004. – 1040 с.
- 7 Котеров Д., Костарев А. PHP. В подлиннике. – СПб.: БХВ–Петербург, 2005. – 1120 с.
- 8 Зандстра М. PHP: объекты, шаблоны и методики программирования. – 3-е изд. – М.: Вильямс, 2010. – 560 с.
- 9 Сайт <http://php.ru/manual/>
- 10 Кузнецов М., Симдянов И. PHP . Практика создания Web-сайтов. – 2-е изд. – СПб.: БХВ–Петербург, 2008. – 1264 с.
- 11 Флэнаган Д. Чего не может JavaScript. Подробное руководство / Перевод Киселева А. – 5-е изд. – СПб.: Символ–Плюс, 2008. – 280 с.
- 12 Кебо Б. Библия jQuery. Руководство профессионала / Перевод тушко Г. – 2-е изд. – СПб.: Амфора, 2012. – 454 с.
- 13 Christian J. Test Driven JavaScript Development. – Addison: Wesley Professional, 2010. – 26 с.
- 14 Стешин А. Изучаем сервер под windows. – СПб: Питер, 2009. – 209 с.
- 15 Гукасян Г.М. Экономика от «А» до «Я»: тематический справочник. – М.: ИНФРА–М, 2009. – 480 с.
- 16 Иванов И.Н. Экономика промышленного предприятия: Учебник. – М.: ИНФРА–М, 2011. – 395 с.
- 17 Рофе А.И. Экономика труда: Учебник. – М.: КноРус, 2010. – 400 с.
- 18 Скляренко В.К., Прудников В.М. Экономика предприятия: Учебник. – М.: ИНФРА–М, 2006. – 528 с.
- 19 Белов С.В., Ильницкая А.В., Козьяков А.Ф. Безопасность жизнедеятельности. Учебник для вузов. – М.: Высшая школа, 2005. – 448 с.
- 20 СНиП РК 4.02–42–2006 Отопление, вентиляция и кондиционирование. – Астана: Издательство стандартов, 2007.
- 21 СНиП РК 2.02–15–2003 Пожарная автоматика зданий и сооружений. – Астана: Издательство стандартов, 2004.
- 22 Хакимжанов т.Е. Сборник задач по охране труда и безопасности жизнедеятельности: Учебное пособие для вузов. – Алматы: Эверо, 2007. – 274 с.

## Приложение А

### Листинг программы

```
create tablespace Hospital datafile
'/u01/app/oracle/oradata/Hospital.dat' size 100M
reuse autoextend on next 5M maxsize 200M
/
commit
/
create user Maksat identified by 1
default tablespace Hospital quota 40M on Hospital
temporary tablespace temp
/
commit
/
grant create user,alter user to Maksat with admin option
/
grant create role to Maksat with admin option
/
grant create session,alter session to Maksat with admin
option
/
grant create table to Maksat
/
grant create procedure to Maksat
/
grant create trigger to Maksat
/
grant create sequence to Maksat
/
grant create view to Maksat
/
commit
/
connect Maksat/1@orcl
/
create table hospital
(
hospital_id int primary key NOT NULL,
hospital_name varchar(50) NOT NULL,
hospital_adress varchar(30) NOT NULL,
hospital_specializ varchar(30),
hospital_director_name varchar(30) NOT NULL,
hosptial_director_fam varchar(30) NOT NULL,
hospital_director_otch varchar (30) NOT NULL
);
create table doljnost
(
doljnost_id int primary key not null,
doljnost_nazvanie varchar(20),
doljnost_oklad int
);
```

## *Продолжение приложения А*

```
create table oborudovanie
(
oborudovanie_id int primary key not null,
oborudovanie_primenenie varchar(20)
);

create table pacient
(
pacient_id int primary key not null,
pacient_name varchar(20),
pacient_fam varchar(20),
pacient_otch varchar(20),
pacient_adress varchar(20),
pacient_district int
);

create table palata_tip
(
tip_id int primary key not null,
tip_kolvo_mest int
);

create table diagnoz
(
diagnoz_id int primary key not null,
diagnoz_name varchar(20)
);

create table drug
(
drug_id int primary key not null,
drug_name varchar(20),
drug_tip varchar(20),
drug_sutoch_doza int,
drug_izgotovitel varchar(20),
drug_annotacia varchar (20)
);

create table lechenie_name
(
lechenie_name_id int primary key not null,
lechenie_name_oblast varchar(20)
);

create table otdelenie
(
otdelenie_id int primary key,
otdelenie_name varchar(50),
otdelenie_corpus int,
otdelenie_procedur varchar(20),
```

## *Продолжение приложения А*

```
hospital_id int references hospital(hospital_id)
);

create table cabinet
(
cabinet_nomer int primary key not null,
cabinet_specializ varchar(20),
otdelenie_id int references otdelenie(otdelenie_id)
);

create table personal
(
personal_id int primary key not null,
personal_name varchar(20),
personal_fam varchar(20),
personal_otch varchar(20),
personal_birth date,
personal_adress varchar(20),
personal_specializ varchar(20),
personal_stavka int,
otdelenie_id int references otdelenie(otdelenie_id),
doljnost_id int references doljnost(doljnost_id),
cabinet_nomer int references cabinet(cabinet_nomer)
);

create table palata
(
palata_nomer int primary key not null,
palata_data_postup date,
palata_data_vipiski date,
tip_id int references palata_tip(tip_id),
otdelenie_id int references otdelenie(otdelenie_id)
);

create table history
(
history_id int primary key not null,
history_individ_osobenn varchar(30),
data_postup date,
data_vipiski date,
personal_id int references personal(personal_id),
pacient_id int references pacient(pacient_id),
palata_nomer int references palata(palata_nomer)
);

create table oborud_v_kabinete
(
cabinet_nomer int references cabinet(cabinet_nomer),
oborudovanie_id int references
oborudovanie(oborudovanie_id),
```



## Продолжение приложения А

```
constraint PK_oborud_v_kab primary key  
(cabinet_nomer, oborudovanie_id)  
);
```

```
create table lechenie  
(  
lechenie_id int primary key not null,  
lechenie_start date,  
lechenie_finish date,  
drug_id int references drug(drug_id),  
lechenie_name_id int references  
lechenie_name(lechenie_name_id)  
);
```

```
create table diagnost_istorii_bolezni  
(  
lechenie_id int references lechenie(lechenie_id),  
history_id int references history(history_id),  
diagnost_id int references diagnost(diagnost_id),  
constraint PK_daagnost_istorii_bolezni primary key  
(lechenie_id, history_id, diagnost_id)  
);
```

## Приложение Б. Каталог функций программного обеспечения

№ п/п	Наименование (содержание) функций	Объем функций (строк исходного кода)		
		С использованием среды разработки приложений		
		Delphi (Borland)	C++ Builder (Borland)	Visual C++ (Microsoft)
<b>1. Ввод, анализ входной информации, генерация кодов и процессор входного языка</b>				
101	Организация ввода информации	100	110	150
102	Контроль, предварительная обработка и ввод информации	550		
103	Анализ входного языка (синтаксический и семантический)	630	660	980
104	Преобразование операторов входного языка и команды другого языка	1050	1050	980
105	Обработка входного заказа и формирование таблиц	750	900	1340
106	Преобразование входного языка в машинные команды (транслятор, препроцессор, макрогенератор)	1300	4300	5100
107	Синтаксический и семантический анализ входного языка и генерация кодов команд	8700	5400	5400
108	Процессор языка	3000	2300	2300
109	Организация ввода/вывода информации в интерактивном режиме	220	220	320
110	Организация ввода/вывода информации с сети терминалов	3680	3340	3200
110	Управление вводом/выводом	2700	2900	2400
<b>2. Формирование, введение и обслуживание баз данных</b>				
201	Генерация структуры базы данных	3450	3950	4300
202	Генерация структуры базы данных	1540	1610	2060
203	Формирование баз данных	1700	1750	2180
204	Обработка наборов и записей базы данных	2050	2350	2670
205	Обслуживание базы данных в пакетном режиме	1030	1100	1260
206	Обслуживание базы данных в интерактивном режиме	3800	4400	6950
207	Манипулирование данными	8400	8670	9550
208	Организация поиска и поиск в базе данных	5230	5460	5480
209	Реорганизация базы данных	130	190	220
210	Загрузки базы данных	3150	2950	2780
<b>3. Формирование и обработка файлов</b>				
301	Формирование последовательного файла	340	360	290
302	Сортировка файла	340	360	290
303	Автоматическая сортировка файлов	1040	1150	930

*Продолжение приложения Б*

304	Обслуживание файлов	520	540	420
305	Обработка файлов	750	800	720
306	Обработка файлов в диалоговом режиме	2400	2600	3050
307	Совместная обработка группы файлов	4900	5300	6180
308	Управление файлами	5130	5380	5750
309	Формирование файла	1100	1080	1020
4. Генерация программ и ПО, а также настройка ПО				
401	Генерация рабочих программ	3680	4120	3360
402	Генерация программ по описанию пользователей	10870	12330	9880
403	Формирование служебных таблиц	570	620	1070
404	Система генерации ПО	5120	5340	4980
405	Система генерации ПО	250	300	370
5. Управление ПО, компонентами ПО и внешними устройствами				
501	Монитор ПО (управление работой компонентов)	350	360	740
502	Монитор системы (управление работой комплекса ПО)	3750	3880	7740
503	Управление внешними устройствами и объектами	6850	7340	5900
504	Обработка прерываний	890	730	540
505	Управление внешней памятью	250	210	200
506	Обработка ошибочных и сбойных ситуаций	430	410	410
507	Обеспечение интерфейса между компонентами	730	750	970
6. Отладка прикладных программ, обмен информацией между магнитным диском и магнитной лентой, вспомогательные программные функции				
601	Отладка прикладных программ в интерактивном режиме	4500	4700	4300
602	Обмен информацией между магнитным диском и магнитной лентой	*	*	*
603	Копирование наборов данных на магнитной ленте и восстановление	*	*	*
604	Справка и обучение	680	680	720
605	Вспомогательные и сервисные программы	460	490	580
7. Расчетные задачи, формирование и вывод на внешние носители (АЦПУ) документов сложной формы и файлов				
701	Матстатистика и прогнозирование	8370	9570	9320
702	Расчетные задачи (расчет режимов обр-ки)	12600	15300	14800
703	Расчет показателей	410	500	460
704	Процессор отчетов	1070	1230	3200
705	Формирование и вывод на внеш.носители	2650	2850	3500
706	Предварит.обработка и печать файлов	540	560	470
707	Графический вывод результатов	300		480
708	Интерактивный редактор текста	3800	3910	4540

**Приложение В.**  
**Укрупненные нормы времени на разработку ПО в зависимости от**  
**уточненного объема ПО и группы сложности ПО**

Объем ПО(строки исходного кода, LOC)	Категории сложности ПО			Категории сложности ПО
	1-я	2-я	3-я	
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
200	—	—	21	1
300	—	—	23	2
400	—	—	25	3
500	—	—	27	4
600	—	33	28	5
700	—	36	30	6
800	—	38	32	7
900	—	40	34	8
1000	51	43	36	9
1200	54	45	38	10
1400	57	48	40	11
1600	60	50	42	12
1800	64	54	45	13
2000	68	57	48	14
2200	73	61	51	15
2400	76	64	54	16
2600	81	68	57	17
2800	86	72	60	18
3000	91	76	64	19
3200	97	81	68	20
3400	103	86	72	21
3600	110	92	77	22
3800	117	98	82	23
4000	124	104	87	24
4200	133	111	93	25
4400	141	118	99	26
4600	151	126	105	27
4800	160	134	112	28
5000	170	142	119	29
5500	182	152	127	30
6000	194	162	135	31
6500	206	172	144	32
7000	220	184	154	33
7500	235	196	164	34
8000	252	210	175	35
8500	268	224	187	36

9000	288	240	200	37
------	-----	-----	-----	----

*Продолжение приложения В*

9500	307	256	214	38
10000	327	273	228	39
11000	349	291	243	40
12000	374	312	260	41
13000	399	333	278	42
14000	427	356	297	43
15000	456	380	317	44
16000	487	406	339	45
18000	520	434	362	46
20000	556	464	387	47
22000	595	496	414	48

24000	636	530	442	49
26000	679	566	472	50
28000	727	606	505	51
30000	775	646	540	52
32000	830	692	577	53
34000	888	740	617	54
36000	950	792	660	55
38000	1016	847	706	56
40000	1087	906	755	57

**Приложение Г.  
Дополнительные коэффициенты сложности ПО**

Характеристика ПО	Значения $K_c$
1. Функционирование ПО в расширенной операционной среде (связь с другими ПО)	0,08
2. Интерактивный доступ	0,06
3. Обеспечение хранения, ведения и поиска данных в сложных структурах	0,07
4. Наличие у ПО одновременно нескольких характеристик по табл.Г4.1, приложение Г	
4.1 2 характеристики	0,12
4.2 3 характеристики	0,18
4.3 Свыше 3-х характеристик	0,26

**Приложение Д.  
Значения поправочного коэффициента, учитывающего  
использование стандартных модулей типовых программ и ПО ( $K_T$ )**

Степень охвата реализуемых функций разрабатываемого ПО стандартными модулями, типовыми программами и ПО	Значения $K_T$
1. От 60 % и выше	0,6
2. От 40 % до 60	0,7
3. От 20 % до 40 %	0,8
4. До 20 %	0,9
5. ПО не используемые для реализации функций	1,0

**Приложение Е.  
Поправочные коэффициенты, учитывающие новизну ПО ( $K_H$ )**

Категория новизны	Степень новизны	Использование		Значение $K_H$
		На основе нового типа ПК	В среде новой ОС	
А	Принципиально новые ПО, не имеющие доступных аналогов	+	+	1,75
		-	+	1,6
		+	-	1,2
		-	-	1,0
Б	ПО, являющиеся развитием определенного параметрического ряда ПО	+	+	1,0
		-	-	0,9
		+	-	0,8

В	ПО, являющиеся развитием определенного параметрического ряда ПО, разработанных для ранее освоенных типов конфигурации ПК и ОС	–	–	0,7
---	---	---	---	-----

**Приложение Ж.**  
**Оценка значений среднего машинного времени на отладку 100 строк  
исходного кода без применения ПО**

Наименование подсистемы АС и СОД	Средний расход машинного времени, ч/100 строк кода
1. Общесистемные задачи: ведение линейных файлов, информационно–поисковые системы, расширение возможностей средств обработки данных, организация вычислительного процесса	12
2. Задачи расчетного характера	15
3. Оперативное управление производством, расчеты по ценообразованию	7
4. техническая подготовка производства, транспортное, энергетическое и инструментальное обслуживание производства	8
5. Бухгалтерский учет, финансовые расчеты, учет пенсий и пособий, учет страховых операций, качество продукции	13
6. Управление кадрами	13
7. технико–экономическое планирование	13
8. Материально–техническое снабжение, реализация и сбыт готовой продукции	13