

Содержание:

ВВЕДЕНИЕ

Информационные сервисы и компьютерные программы уже очень плотно вошли в жизнь человека. Сегодня, в разгар информационной эры развития человечества, невозможно представить себе существование человека без использования ИТ. Так как программное обеспечение стало неотъемлемой частью человеческого существования, аспекты разработки и внедрения ПО стали рассматривать на научном уровне, отдельные процессы и стадии разработки ПО породили новые направления в науке. Так например существуют отдельные направления в науке которые рассматривают: методы и подходы моделирования (ООП, UML, IDEF), подходы и методы разработки (структурное, процедурное, объектное, функциональное), тестирование и другие процессы.

Отдельным направлением является процесс разработки взаимодействия человека и машины – построение интерфейсов и диалогов общения компьютера и оператора (пользователя).

Инженерная наука Usability занимается вопросами создания и воплощения эффективных человеко-машинных интерфейсов (ЧМИ) или другими словами интерфейса пользователя (ИП, в литературе чаще встречается англ. аббревиатура UI - UserInterface). Ошибки и неэффективность ИП могут дорого обойтись как пользователю (от низкой производительности до катастроф), так и производителю (от потери рынков к судебным разбирательствам и финансовым претензиям).

Название происходит от английских слов USER (пользователь) и ABILITY (способность, умение, а еще: талант, одаренность, компетенция и правомочность).

Первыми на ИП обратили внимание, конечно, военные. Для них создание качественного интерфейса для летчика, ракетчика, и вообще любого оператора – зачастую вопрос жизни и смерти (прежде всего самого оператора).

Постепенно на Западе сложилось представление и наука о UI.

Проектирование взаимодействия помогает понять, чего именно хочет пользователь. Зная это, можно создавать качественные, понятны и более успешные цифровые продукты. Практически все виды проектирования, такие как

архитектура, графический дизайн и другие, влияющие на поведение пользователя. Также следует отметить, что проектирование имеет равные права с разработкой, маркетингом и управлением бизнесом. [16]

Актуальность темы. Проектирование взаимодействия - это относительно новая область знаний, начала стремительно развиваться в последние десятилетия с появлением цифровых интерактивных технологий. Более того, проектирование взаимодействия превратилось в довольно мощный инструмент управления разработкой программного продукта, а также на маркетинговый инструмент, предъявляет четкую информацию о том, кто будет использовать данный продукт и для чего.

Объектом исследования в работе является понятие пользовательского интерфейса, его места и значимости в жизненном цикле ПЗ.

Предметом исследования курсовой работы является типы меню как основного элемента пользовательского интерфейса.

Цель работы: раскрыть понятие интерфейса, проектирование взаимодействия, рассмотреть этапы проектирования и разработки пользовательского интерфейса при создании цифровых продуктов и программных приложений.

Задачи курсовой работы можно сформировать следующим образом:

- раскрыть основные понятия и виды пользовательского интерфейса;
- исследовать основы проектирования и способы создания качественного интерфейса пользователя;
- выделить типы меню, как элемента пользовательского интерфейса;
- рассмотреть различные способы построения меню;
- рассмотреть построение интерфейса на примере конкретного программного обеспечения.

1. МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ

Темой работы и объектом исследования является понятие меню. Это понятие неразрывно связано с понятиями «интерфейс» и «диалог». Хотя сегодня меню

можно рассматривать как отдельный элемент понятия интерфейс, но изначально (начальный период пользовательского ПО) весь интерфейс и состоял только лишь из меню и сообщений/вводов. Сегодня часто интерфейс программы называют – меню, несколько подменяя понятия. С другой стороны – невозможно рассматривать понятие «меню» без рассмотрения основных характеристик и общих требований к интерфейсу в целом.

1.1 Понятие и классификация интерфейсов

Пользовательский интерфейс - это набор программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером. В основе взаимодействий лежат диалоги. Под диалогом в данном случае следует понимать регламентированный обмен информацией между человеком и компьютером, который осуществляется в реальном масштабе времени и предназначен для эффективного совместного решения поставленной задачи. Диалоги состоят из отдельных процессов ввода/вывода, эти процессы реализуют обмен информацией между человеком и системой, обеспечивая информационные связи. Обмен информацией осуществляется передачей сообщения.

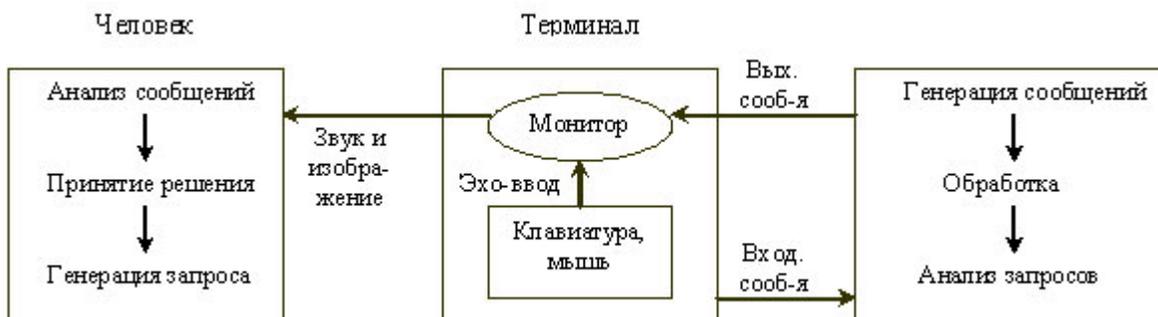


Рисунок 1.1 - Общая схема процесса диалога

В основном пользователь генерирует сообщения следующих типов:

- запрос информации
- запрос помощи
- запрос операции или функции
- ввод или изменение информации

В ответ пользователь получает подсказки или справки; информационные сообщения, требующие ответа; приказы, требующие действия; сообщения об ошибках и другую информацию.

1.1.1 Типы интерфейсов

Интерфейсы пользователя бывают двух типов:

1) процедурно-ориентированные:

-примитивные;

-меню;

-свободная навигация;

2) объектно-ориентированные:

-прямое манипулирование.

Процурно ориентированный интерфейс использует традиционную модель взаимодействия с пользователем, основанную на понятиях «процедура» и «операция». В рамках этой модели программное обеспечение предоставляет пользователю возможность выполнения некоторых действий, для которых пользователь определяет соответствие данных и следствием выполнения которых является получение желаемого результата.

Объектно-ориентированные интерфейсы используют модель взаимодействия с пользователем, ориентированную на манипулирование объектами предметной области. В рамках этой модели пользователю предоставляется возможность напрямую взаимодействовать с каждым объектом и инициировать выполнение операций, в процессе которых взаимодействуют несколько объектов. Задача пользователя формулируется как целенаправленное изменение некоторого объекта. Объект понимается в широком смысле слова - модель БД, системы и т.д.

Интерфейс Меню. В отличие от примитивного интерфейса, позволяет пользователю выбрать операцию из специального списка, выводимого ему программой. Эти интерфейсы предполагают реализацию множества сценариев работы, последовательность действий в которых определяется пользователями. Древовидная организация меню предполагает строго ограниченную реализацию. При этом возможны два варианта организации меню:

-каждое окно меню занимает весь экран

-на экране одновременно присутствуют несколько разноуровневых меню (Windows).

В условиях ограниченной навигации, независимо от варианта реализации, поиск пункта более чем двух уровневых меню оказывается довольно сложной задачей.

1.1.2 Типы диалогов с пользователем, сообщения системы

Классификация сообщений в соответствии с выполняемыми ими функциями:

Подсказка - это выходное сообщение системы, побуждающие пользователя, вводит данные.

Команда - входное управляющее сообщение, предназначенное для управления ходом диалога.

Входные данные - данные, необходимые для выполнения процесса.

Сообщение об ошибке - это сигнал диалогового процесса о том, что невозможно дальнейшее выполнение работы, потому что вызванный процесс выполнения задания не может обработать введенное сообщение.

Выходные данные - это данные, которые возвращает процесс по окончании обработки задания. Процесс выполнения задания передает выходные данные в стандартной форме на вход диалогового процесса, который преобразует их в подходящий для пользователя формат.

Сообщение о состоянии системы - это информация для пользователя о том, что произошло или происходит в системе.

Справочная информация - это информация, поясняющая, как должен поступить пользователь в той или иной ситуации.

Тип диалога определяет, кто из «собеседников» управляет процессом обмена информацией. Соответственно различают два типа диалога: управляемые программой и управляемые пользователем.

Диалог невозможен, если не существует языка, понятного «собеседникам». Описание языка, на котором ведется диалог, включает определение его **синтаксиса** - правил, определяющих допустимые конструкции (слова, предложения) языка или его форму, и **семантики** - правил, определяющих смысл синтаксически корректных конструкций языка или его содержание. В зависимости от вида используемых в конкретном случае синтаксиса и семантики различают три формы диалога:

- фразовую;
- директивную;
- табличную.

Следует иметь в виду, что типы и формы диалогов выбираются независимо один от другого: любая форма применима для обоих типов диалогов (рис. 1.2). Однако фразовая форма, которая используется в диалоге, управляемом пользователем, как правило, предполагает более сложные синтаксис и семантику языка диалога, так как программа должна «понимать» пользователя.



Рисунок 1.2 - Соответствие типов диалогов и его форм

Сложное программное обеспечение обычно взаимодействует с пользователем посредством диалогов различных типов и форм в зависимости от решаемых задач. Причем, помимо диалогов, происходящих в процессе нормальной работы программного обеспечения и называемых **синхронными**, предусматривают диалоги, возникающие по инициативе системы или пользователя при нарушении сценария нормального процесса. Такие диалоги называются **асинхронными**. Обычно их используют для выдачи экстренных сообщений от системы или пользователя.

При проектировании диалогов применяются различные методики, это могут быть сценарные формы, диаграммы интерфейсных состояний, может использоваться и такая форма, как граф диалога.



Рис. 8.13. Граф абстрактного диалога комбинированного типа

Рисунок 1.3 – пример графа диалога

1.2 Методы ведения диалога с ИС

Основное требование, предъявляемое к разработке интерфейса, заключается в том, что бы конечный пользователь, не имея специальных навыков и знаний в области программирования, мог удовлетворять свои информационные потребности и эффективно взаимодействовал с ЭВМ.

На первичном этапе развития систем и ПО применялся пакетный режим. При таком режиме пользователь не имеет прямого доступа к ЭВМ, его функции ограничиваются подготовкой исходных данных по комплексу (пакету) задач и передачей их в центр обработки, содержащий задание для ЭВМ на обработку,

программы и нормативно-справочные данные.

Диалоговый режим взаимодействия пользователя и ЭВМ обеспечивает возможность оперативного вмешательства человека в процесс обработки информации на ЭВМ. На практике часто можно наблюдать совместное использование режимов, помогающее за счет их частных преимуществ организовать более эффективную технологию решения задач на ЭВМ. Понимание при диалоге достигается наличием единой системы языковых знаков или кодов, из которых формируются сообщения. Общие знания и умения у партнеров диалога обеспечивают эффективное понимание друг друга.

Если роли четко определены и ограничены для каждого из участников диалога, то диалог называют «жестким», например, режим работы «вопрос - ответ» с указанием того, кому из партнеров принадлежит инициатива. Альтернативная жесткой структуре форма диалога задает множество предписанных вариантов диалога, представляемых пользователю в виде меню, как правило, иерархической структуры, из которого он выбирает направление решения задачи. Такой диалог называется гибким. Наконец, свободным называется диалог, позволяющий участникам общения обмениваться информацией произвольным образом. Такой тип диалога применяется крайне редко, так как требует формализации сообщений от пользователя, которые представлены в свободной форме, а эта задача сама по себе достаточно сложна. Хотя попытки построения таких систем периодически возникают, так в начале века очень популярной была идея реализации системы голосового управления компьютером и программами. Были реализованы некоторые аналоги, но в последствии технология не получила развития и распространения.

Меню как тип диалога очень удобен для конечного пользователя. Реализация диалога типа «меню» возможна через вывод на экран видеотерминала определенных функций системы.

Выбор конкретной функции пользователем может осуществляться:

набором на клавиатуре требуемой директивы или ее сокращенного обозначения;

набором на клавиатуре номера необходимой функции;

подведением курсора в строку экрана с нужной пользователю функцией;

нажатием функциональных клавиш, запрограммированных на реализацию данной функции.

При наличии различных вариантов ответов на ввод функций пользователем в последующих шагах производится детализация, или уточнение действий, например, какая информация должна вводиться, в каком виде или на какое устройство желательно осуществить вывод и т.д. Частным случаем диалога типа «меню» является режим ответа «ДА/НЕТ», т.е. пользователю предлагаются два альтернативных варианта ответа: ДА или НЕТ.

Для всех категорий пользователей программных средств, работающих в режиме диалога, обязательной является включаемая в них система помощи и средств обучения (HELP), ускоряющая как процесс освоения, так и процесс работы. Освоение основных функций любого пакета диалогового типа не должно требовать специальных знаний в области языков программирования, архитектуры ЭВМ и пр. Пользователь работает с различными диалоговыми программными системами, поэтому в них целесообразно закладывать некоторое единообразие. Например, использование функциональных клавиш F1 и F10 обеспечивает вызов помощи и выход из системы, применение управляющих клавиш или их комбинаций для управления состоянием процесса вычислений и т.д.

1.3 Виды меню

На сегодняшний день существует большое разнообразие типов меню и видов интерфейсов. Объем работы не позволяет детально рассмотреть и исследовать все существующие на сегодня типы. По этому - остановимся только на некоторых типах меню.

Исторически первыми применяемыми типами меню были меню в виде списка пунктов с предоставлением выбора пункта пользователю. Этот тип меню широко применялся для ОС без графического интерфейса или в консольных приложениях. Рассмотрим пример такого меню.

1.3.1 Консольное текстовое меню

Рассмотрим пример кода реализации подобного меню. Обычно такое меню предполагает наличие двух блоков в программе: визуализация – вывод пунктов меню и структура ветвления – вызов функций в зависимости от выбора пользователя.

Пример листинга вывода пунктов меню

```
static int menu()
{
int i=0;

Console.WriteLine("===== М Е Н Ю
=====");

Console.WriteLine("(1)-Создать массив случайных эл.");
Console.WriteLine("(2)-Создать массив и ввести эл.");
Console.WriteLine("(3)-Создать массив и заполнить из файла");
Console.WriteLine("(4)-Выделить фрагмент массива");
Console.WriteLine("(5)-Сумма всех эл. фрагмента");
Console.WriteLine("(6)-Сумма эл. фрагмента по строкам");
Console.WriteLine("(7)-Сумма эл. фрагмента по столбцам");
Console.WriteLine("(8)-Индексы максимальных эл. фрагмента по строкам");
Console.WriteLine("(9)-Индексы максимальных эл. фрагмента по столбцам");
Console.WriteLine("(0)-Выход");

Console.WriteLine("=====");

Console.WriteLine("Ваш выбор:");

return i = Convert.ToInt32(Console.ReadLine());
}
```

Как видно из примера, функция включает только операторы вывода и оператор считывания значения с клавиатуры. Функция возвращает значения целого типа – пункт выбранный пользователем. При вызове данной функции в основной программе этот параметр выступает ключом в структуре ветвления, по которому происходит вызов нужной функции в зависимости от выбора пользователя.

Пример структуры ветвления на основе конструкции switch

```
int punct=menu();  
while (punct!=0)  
{  
switch (punct)  
{  
case 1:  
{  
my_ar.setvalue(1);  
Console.WriteLine("Исходный массив:");  
my_ar.ShowAr();  
break;  
}  
case 2:  
{  
Console.WriteLine("Введите эл. массива:");  
my_ar.setvalue(2);  
Console.WriteLine("Исходный массив:");  
my_ar.ShowAr();  
break;  
}  
}
```

Вид текстового меню в процессе выполнения программы представлен на следующем рисунке

```

=====  М Е Н Ю  =====
(1)-Создать массив случайных эл.
(2)-Создать массив и ввести эл.
(3)-Создать массив и заполнить из файла
(4)-Выделить фрагмент массива
(5)-Сумма всех эл. фрагмента
(6)-Сумма эл. фрагмента по строкам
(7)-Сумма эл. фрагмента по столбцам
(8)-Индексы максимальных эл. фрагмента по строкам
(9)-Индексы максимальных эл. фрагмента по столбцам
(0)-Выход
=====
Ваш выбор:

```

Рисунок 1.4 - Текстовое меню

1.3.2 Псевдографические меню, меню с использованием DOS-графики

Разработчики всегда пытались привлечь внимание к своему продукту используя более яркий и привлекательный интерфейс. Естественно ограничиваясь только текстовыми выводами особого разнообразия добиться сложно. Появились интерфейсы в так называемом псевдографическом формате. При построении интерфейса используются псевдографические символы для формирования рамок, разделение зон и так далее. Ярким и широко известным представителем является файловый менеджер VolcovCommander.



Рисунок 1.5 - Псевдографический интерфейс

Другим подходом, который начал применяться во второй половине прошлого века, стал графический интерфейс на основе DOS-графики (16 или 256 цветов). Параллельно с использованием графических элементов (в большей степени прямоугольные объекты) начали использовать и манипулятор мышь.



Рисунок 1.6 – Графический DOS интерфейс

Интерфейс строился на основе графических примитивов графической библиотеки выбранного языка разработки, например рассмотрим фрагмент построения меню представленного на рисунке 1.6 средствами языка С

Фрагмент кода:

```
int gdriver=DETECT,gmode,errorcode;

initgraph(&gdriver,&gmode,"");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Error: %s\n", grapherrormsg(errorcode));

getch();

exit(1);

}
```

```
//*****МЕНЮ*****
```

```
m1:  
setcolor(15);  
setfillstyle(1,1);  
bar (1,1,640,480);  
line(75,1,75,18);  
outtextxy(10,10,"<<Back");  
outtextxy(80,10,"Start game");  
setcolor(15);  
outtextxy(220,10,"-= GUESS A MELODY =-");  
setcolor(0);  
bar (250,462,400,480);  
outtextxy(260,465,"About the program");
```

1.3.3 Оконные интерфейсы

После появления ОС Windows 98 и распространения графического интерфейса на основе окон, среды разработки программного обеспечения стали включать специальные редакторы – проектирование форм и интерфейсов. В разных средах разработки применяются разные подходы, но базовый набор компонентов достаточно похож. На рис. 1.7 представлено приложение в среде IDE Delphi 7 на этапе проектирования форм.

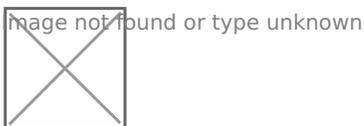


Рисунок 1.7 – проектирование интерфейса в среде Delphi

Пояснения к интерфейсным элементам:

1 – Категории продукта/сырья – ListBox (список пунктов)

2 – Список продуктов в категории – StringGrid (таблица для отображения и вывода)

3 – Реализация меню на основе Label

3.1-3.4 – Область отображения движения по складу –DBStringGrid (таблица для вывода данных из БД)

4 – меню выбора склад сырья/ склад продукции –ComboBox (выпадающий список)

На рисунке 1.8 приведены примеры использования контекстного меню и формы ввода данных.

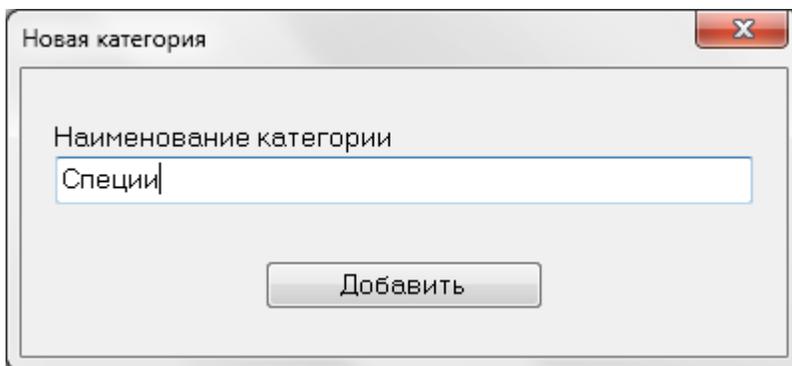
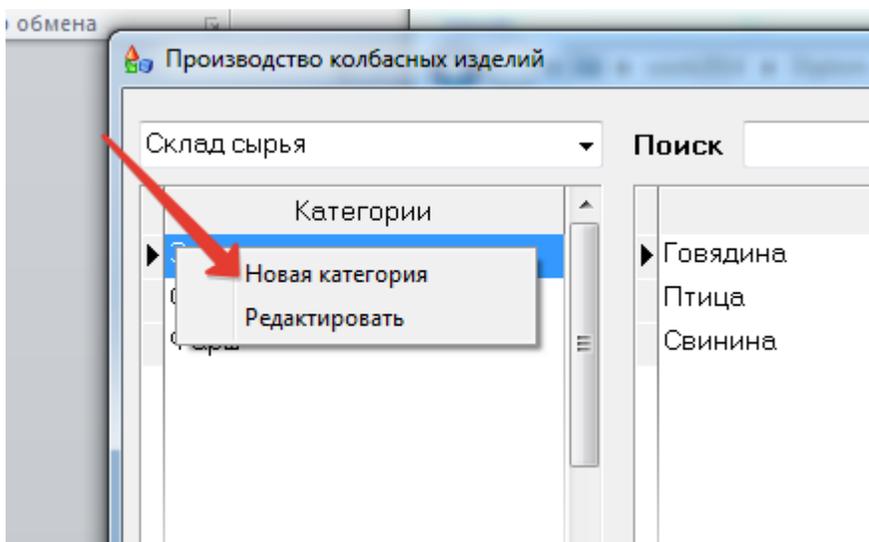


Рисунок 1.8 – контекстное меню и форма ввода

1.3.4 Рисованное меню

Рисованные меню применяются при разработке мультимедийных программ, справочников и энциклопедий, игровых программ. В зависимости от задумки автора и необходимой сложности конечной реализации могут применяться как стандартные языковые средства, так и специальные средства для работы с графикой. Примером наиболее простого способа реализации является перезагрузка ранее подготовленных изображений с реакцией на события движения, позиционирования курсора мыши и клика в определенных активных областях. Рассмотрим пример реализации такого меню для игровой обучающей программы по экономике.

Внешний вид главного окна (основного меню) представлен на рисунке 1.9.

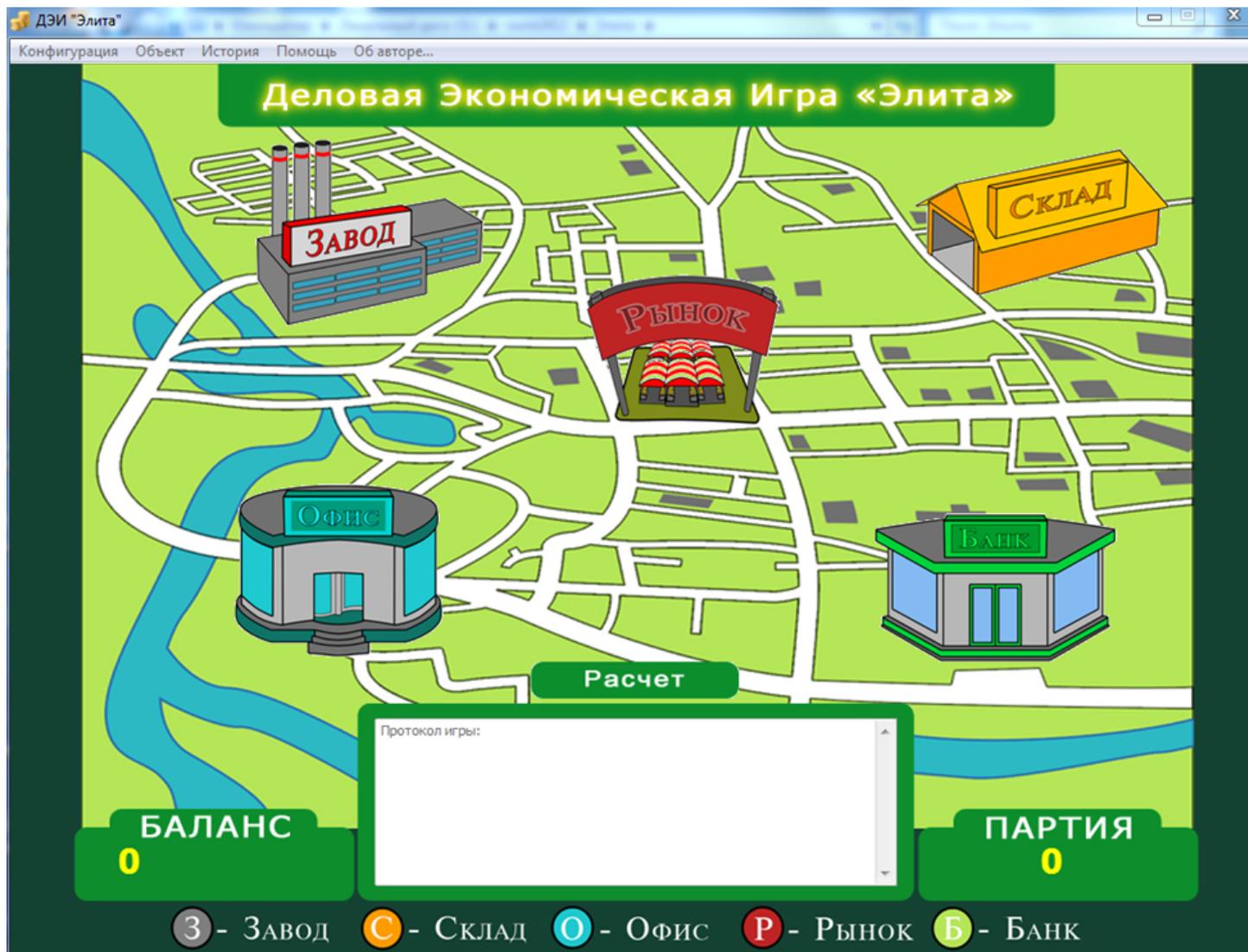


Рисунок 1.9 – главное меню игры «Элита»

Меню использует заранее заготовленные графические элементы



Не выделенный пункт меню «Банк»



Выделенный пункт меню «Банк»

Расчет

Не выделенный пункт меню «Расчет»

Расчет

Выделенный пункт меню «Расчет»

Интерактивность и изменение внешнего вида в зависимости от реакции и действий пользователя реализуется следующим кодом.

Данная часть программы является главным меню программы. Реализована эта часть на основе компонента Image. На этом компоненте, по событию onClick, можно перейти на другую часть программы.

- Сделать невидимой главную форму;
- Вызвать форму в зависимости от действий и выбора пользователя.

Реализуется следующим кодом:

```
procedure TForm1.Image2Click(Sender: TObject);
```

```
begin
```

```
Form1.AlphaBlend := true;
```

```
Form1.AlphaBlendValue := 0;
```

```
Form2.ShowModal;
```

```
end;
```

Подсветка пунктов меню реализована следующим кодом

```
procedure TForm1.Image2MouseMove(Sender: TObject; Shift: TShiftState; X,  
Y: Integer);
```

```
begin
```

```
Form1.Image1.Picture.LoadFromFile(getcurrentdir+'\Image\Fon\Банк_Сел.png');
```

```
end;
```

```
procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,  
Y: Integer);
```

```
begin
```

```
Form1.Image7.Picture.LoadFromFile(getcurrentdir+'\Image\Fon\Банк.png');
```

```
Form1.Image6.Picture.LoadFromFile(getcurrentdir+'\Image\Fon\Рынок.png');
```

```
Form1.Image2.Picture.LoadFromFile(getcurrentdir+'\Image\Fon\Производство.png');
```

```
Form1.Image3.Picture.LoadFromFile(getcurrentdir+'\Image\Fon\Выход.png');
```

```
Form1.Image5.Picture.LoadFromFile(getcurrentdir+'\Image\Fon\Статистика.png');
```

```
end;
```

```
procedure TForm1.Image2Click(Sender: TObject);
```

```
begin
```

```
Form1.AlphaBlend := true;
```

```
Form1.AlphaBlendValue := 0;
```

```
Form2.ShowModal;
```

end;

Примеры других рисованных меню представлены в приложении 1.

1.4 Требования, учитываемые при проектировании

Разработки интерфейсов всегда должны учитывать физические и умственные способности людей, которые будут работать с программным обеспечением. Люди на короткое время могут запомнить весьма ограниченный объем информации и совершают ошибки, если приходится вводить вручную большие объемы данных или работать в напряженных условиях. Физические возможности людей могут существенно различаться, поэтому при проектировании интерфейсов необходимо постоянно помнить об этом.

Основой принципов проектирования интерфейсов человеческие возможности.

В табл. 1.1 представлены основные принципы, применимые при проектировании любых интерфейсов.

Таблица 1.1 - Принципы проектирования интерфейсов

Принцип	Описание
Учет знаний пользователя	В интерфейсе необходимо использовать термины и понятия, взятые из опыта будущих пользователей системы.
Согласованность	Интерфейс должен быть согласованным в том смысле, что однотипные (но разные) операции должны выполняться одним и тем же способом.
Минимум неожиданностей	Поведение системы должна быть прогнозируемой.

Способность к восстановлению	Интерфейс должен иметь средства, позволяющие пользователям восстановить данные после ошибочных действий.
Руководство пользователя	Интерфейс должен предоставлять необходимую информацию в случае ошибок пользователя и поддерживать средства контекстно-зависимой справки.
Учет разнородности пользователей	В интерфейсе должны быть средства для удобного взаимодействия с пользователями, имеющими разный уровень и возможности.

Принцип учета знаний пользователя предполагает следующее: интерфейс должен быть настолько удобен при реализации, чтобы пользователям не понадобилось особых усилий, чтобы привыкнуть к нему. В интерфейсе должны использоваться термины, понятные пользователю, а объекты, управляемые системой, должны быть непосредственно связаны с рабочей средой пользователя. Например, если разрабатывается система, предназначенная для авиадиспетчеров, то управляемыми объектами в ней должны быть самолеты, траектории полетов, сигнальные знаки и тому подобное. Основную реализацию интерфейса в терминах файловых структур и структур данных необходимо скрыть от конечного пользователя. Принцип согласованности интерфейса предполагает, что команды и меню системы должны быть одного формата, параметры должны передаваться во все команды одинаково и пунктуация команд должна быть похожей. Такие интерфейсы сокращают время на обучение пользователей. Знания, полученные при изучении какой-либо команды или части приложения, можно затем применить при работе с другими частями системы.

Всегда нужно соблюдать следующий принцип: количество неожиданностей должна быть минимальной, поскольку пользователей раздражает, когда система вдруг начинает вести себя непредсказуемо. При работе с системой у пользователей формируется определенная модель ее функционирования. Если его действие в одной ситуации вызывает определенную реакцию системы, естественно ожидать, что такое же действие в другой ситуации приведет к аналогичной реакции. Если же происходит совсем не то, что ожидалось, пользователь или удивляется, или не знает, что делать. Поэтому разработчики интерфейсов должны

гарантировать, что подобные действия произведут похожее впечатление.

Очень важный принцип восстанавливаемости системы, поскольку пользователи всегда допускают ошибки. Правильно спроектированный интерфейс может уменьшить количество ошибок пользователя (например, использование меню позволяет избежать ошибок, возникающих при вводе команд с клавиатуры), однако все ошибки устранить невозможно. В интерфейсах должны быть средства, по возможности предотвращают ошибки пользователя, а также позволяют корректно восстановить информацию после ошибок. Эти средства бывают двух видов.

1. Подтверждение деструктивных действий - Если пользователь выбрал потенциально деструктивную операцию, то он должен еще раз подтвердить свое намерение.
2. Возможность отмены действий - Отмена действия возвращает систему в то состояние, в котором она находилась до их выполнения. Не лишней будет поддержка многоуровневой отмены действий, поскольку пользователи не всегда сразу понимают, что совершили ошибку.

Следующий принцип - поддержка пользователя. Средства поддержки должны быть встроены в интерфейс и систему и обеспечивать различные уровни помощи и справочной информации. Должно быть несколько уровней справочной информации - от основ для начинающих до полного описания возможностей системы. Справочная система должна быть структурированной и не перегружать пользователя лишней информацией при простых запросах к ней.

Принцип учета разнородности пользователей предполагает, что с системой могут работать разные их типы. Часть пользователей работает с системой нерегулярно, время от времени. Но существует и другой тип "опытные пользователи", которые работают с приложением ежедневно по несколько часов. Случайные пользователи нуждаются в таком интерфейсе, который "руководил" бы их работой с системой, тогда как опытным пользователям нужен интерфейс, который позволил бы им максимально быстро взаимодействовать с системой.

1.5 Последовательность проектирования

Во многих проектах разработки ИП и программирования продукта осуществляются параллельно, особенно на ранних стадиях. На более поздних этапах учитываются

требования ИП и обратной связи, которые выявляются в результате тестирования ПО на удобство применение.

Процесс проектирования ИП состоит из 3 обобщенных этапов:

- Сбор и анализ информации от пользователей;
- Разработка и реализация ИП;
- Подтверждение качества ИП.

Данный подход может использоваться как при разработке ООИП, так и при проектировании традиционных проблемно-ориентированных интерфейсов или ИП. Этот процесс не зависит от материальной и программной платформ, ОС, применяемого инструментария.

Первый этап - сбор и анализ информации от пользователей. Прежде чем приступить к разработке ПО, нужно определить, какие проблемы пользователи хотят решить, и как они привыкли работать. Следует обратить внимание на ограничения, которые накладываются компьютерными системами на техническое и программное обеспечения. Предложенное разработчиком решение о ИП должно отвечать не только нынешним, но и будущим потребностям пользователей.

Существует ряд ключевых вопросов, которые следует задать на этапе анализа информации пользователей.

Первый этап может быть разделен на 5, перечисленных в таблице, подэтапов.

Таблица 1.2 - Шаги этапа сбора и анализа информации пользователей

Шаг	Описание шага
Определение профиля пользователя	Профиль позволяет составить представление о возрасте, образовании, предпочтения пользователя и др.
Анализ задач, которые стоят перед пользователем	Определение, чего хотят пользователи и каким образом они собираются решать свои задачи

Анализ рабочей среды пользователей

Отвечает на вопрос "В какой рабочей среде пользователи решают свои задачи?", то есть определяет характеристики среды, которые могут влиять на выполнение пользователями своей работы

Соответствие требований пользователей задачам, которые стоят перед ними

Проверка требований на реалистичность. Если требования пользователя не соответствуют выполняемым задачам, то нужно предложить более оптимальный вариант; следует проверить, не превышают возможности продукта потребности пользователя.

Итерационный подход предусматривает возвращение к этапу анализа требований пользователя, чтобы проверить, не изменились в процессе проектирования и разработки профиль пользователей, задачи, характеристики среды или же требования. Для построения качественного продукта нужно периодически возвращаться к первому этапу, чтобы обновлять сведения о пользователях.

После проведенного анализа можно перейти непосредственно к проектированию и реализации интерфейса.

Последовательность проектирование меню в частности и интерфейса в целом включает в себя такие фазы как:

1. Разработка сценария – определения последовательности и вида взаимодействия ПО и пользователя в зависимости от функциональных требований к ПО и вариативности возникших ситуаций.
2. Выбор типов диалогов и системы сообщений.
3. Выбор используемых компонентов для построения меню (в случае графического интерфейса).
4. Проектирование и прототипизирование интерфейса. Разработка эскизов, определение взаимосвязей, размещения, выбор внешних параметров элементов интерфейса: размер, цвет и другие.

5. Анализ возможности верификации и проверки корректности ввода данных пользователем – внесение корректив в выбор элементов с целью уменьшения ошибок и влияния человеческого фактора. Например – выбор из предложенных вариантов вместо ввода, использование компонентов типа Data/Time Picker вместо ввода даты в текстовое поле, повторные запросы при выборе критических параметров (например - удаление информации), ограничение допустимых диапазонов данных.

6. Реализация интерфейса с помощью выбранных средств.

2. Проектирование системы меню по задаче «аис «учет заказов по обслуживанию офисной техники»»»

2.1. Экономическая сущность задачи

В современных реалиях, офисное оборудование, в том или ином виде находится практически в каждом доме, если брать обычную пользовательскую технику, и, конечно же, на каждом предприятии, только в этом случае масштабы использования и сложность оборудования намного выше. При таком распространения возникает проблема в надлежащем обслуживании техники. Небольшие организации уже не могут справиться со все большим числом клиентов и сложностью оборудования, которые могут предоставить на обслуживание различные предприятия. В таком случае предприятия, обеспечивающие ремонтно-эксплуатационные работы с оргтехникой, развиваются дальше, становятся больше, или же возникают новые, ориентированы на возрастающий спрос по обслуживанию оргтехники.

«РН-Информ» является одним из тех предприятий, которые направлены на очень широкий круг клиентов, начиная от рядовых пользователей заканчивая обслуживанием крупных предприятий с профессиональным офисным оборудованием. Основываясь на мировом опыте, предприятием была избрана модель поведения на рынке, в основе которой: высокое качество продукции, индивидуальный подход к каждому заказчику, гибкие условия сотрудничества.

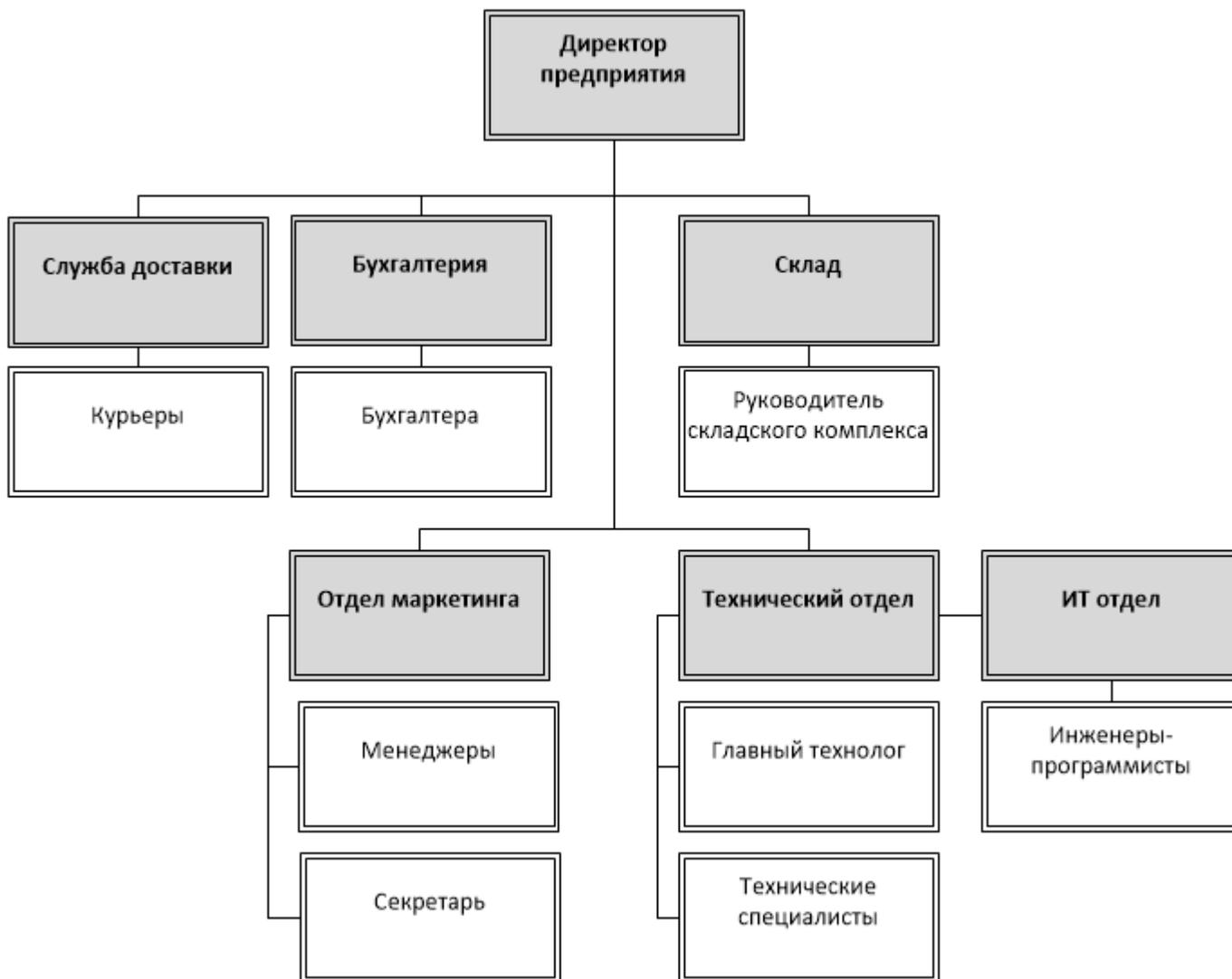


Рисунок 2.1 – Структура предприятия «PH-Информ»

Сервисный центр – одно из приоритетных направлений деятельности «PH-Информ». Таким образом, предприятие представляет слаженно работающий коллектив, состоящий из 30 человек, готовый к решению сложных задач на благо заказчиков, в следующих направлениях:

- печатающая техника
- компьютерная техника
- аудио-видео системы
- телекоммуникации
- сервис

Услуги предоставляемые «PH-Информ»:

- Ремонт печатающей техники: МФУ, копиры, принтеры, сканеры, плоттеры

- Ремонт компьютерной техники, включая: ПК, ноутбуки, планшеты
- Ремонт электроники: мониторы, ИБП, и т.д.
- Комплексное обслуживание предприятий по ИТ-направлению
- Установка и настройка мини-АТС, IP-телефония
- Заправка лазерных картриджей

В оказании услуг принимают участие «Сотрудники» предприятия. Чтобы предоставить нужные услуги заказчику и получить прибыль.

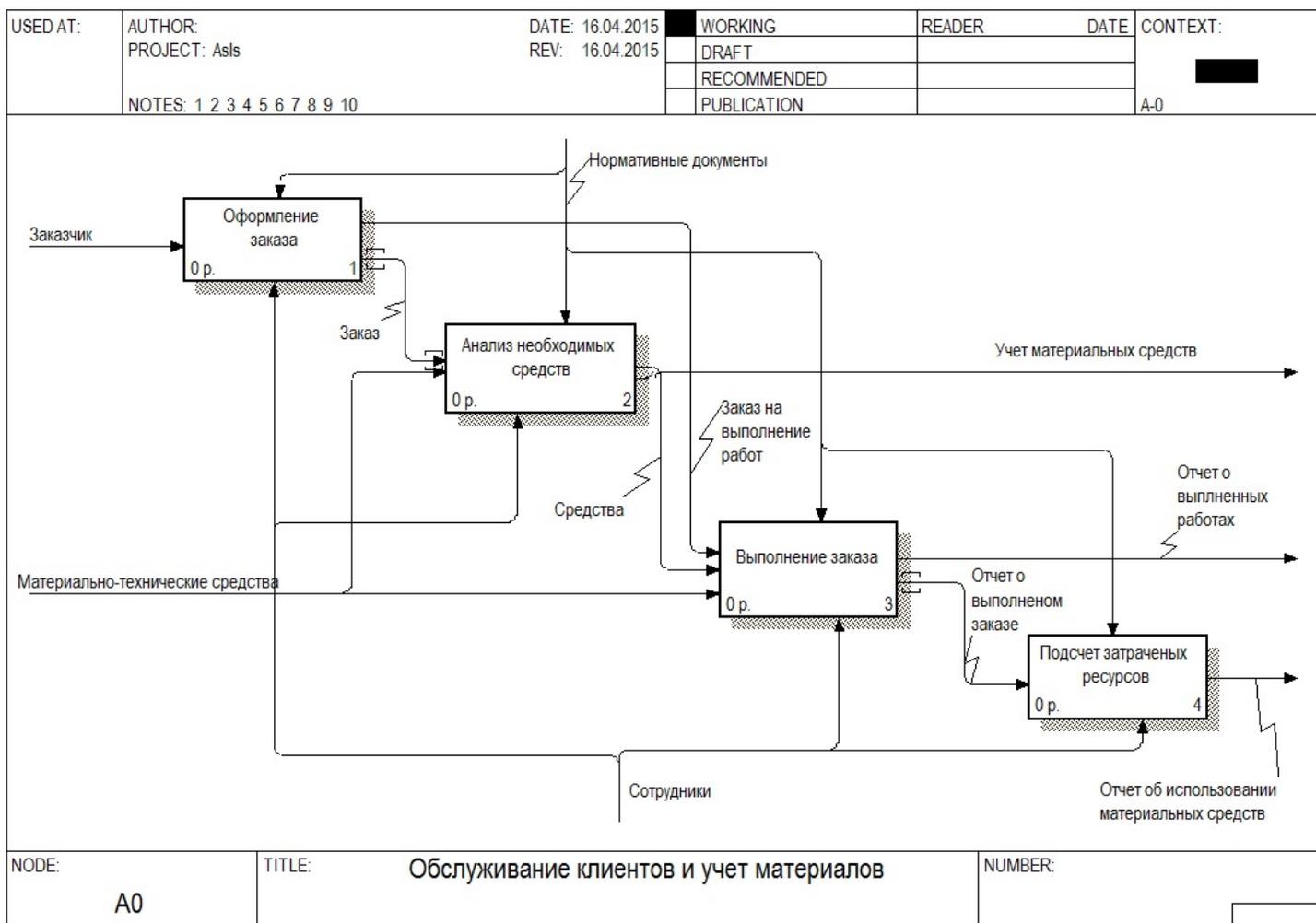


Рисунок 2.2 – Функциональная модель предприятия. Декомпозиция

Весь процесс разбивается на четыре группы:

- 1. «Оформление заказа» представляет собой процесс регистрации заказчика и оформление документов на выполнение работ;
- 2. «Анализ необходимых средств» иллюстрирует деятельность сотрудников по анализу проблемы описанной заказе, и обоснованию (выявлению) использования средств для осуществление ремонтно-профилактических

- работ;
3. «Выполнение заказа» – непосредственно процесс осуществления ремонтно-профилактических работ.
 4. «Подсчет затраченных ресурсов» – процесс оставления бланков и актов о выполненных работах. Осуществляется инвентаризация материалов, потраченных на ремонт и подсчет затрат.

Общие связи, перешедшие с диаграммы верхнего уровня, опишем с помощью отчета:

Таблица 2.1 – Описание связей модели (рис 2.2)

Название	Определение
Заказчик	Лицо решившее отдать технику на ремонтно-профилактические работы
Заказ	Сформированный пакет документов относительно потребностей заказчика
Нормативные документы	Документы, содержащие определенные требования к предприятию, его работникам и к процессу выполнения работ
Средства	Товарно-материальные средства необходимые для выполнения заказа
Материально-технические средства	Оборудование, необходимое для осуществления ремонтно-профилактических работ
Сотрудник	Сотрудники предприятия
Учет материальных средств	Средств на складском помещении

Отчет о выполненных работах	Акт и квитанция, предоставляемые заказчику, описывающие выполненную работу
Отчет об использовании материальных средств	Отчет, содержащий инвентаризационную информацию по использованным товарно-материальным средствам

Если кратко обобщить суть разрабатываемой системы, то кратко постановку задачи можно сформулировать следующим образом.

Фирма занимается ремонтом, профилактическими работами и обслуживанием офисной и копировальной техники (принтеры, копиры, сканеры, ксероксы). Необходимо разработать систему учета заказов по обслуживанию офисной техники. Система должна вести учет выполненных заказов и, кроме этого, в системе предполагается учет и ведение склада расходных материалов, затраченных при выполнении работ. Например, при выполнении заказа «профилактики и заправка принтера» в качестве расходных материалов могут быть использованы: тонер (определенной марки), барабан (при его замене), чистящие ножи и другие расходные материалы.

2.2. Проектирование системы меню

Пользователь является ключевым действующим объектом системы. Он оперирует всеми ее возможностями путем управления ее функциональными подсистемами, которые в свою очередь создают внутренние исполняющие потоки. На рис. 2.3 показаны общие варианты использования разрабатываемой системы.



Рисунок 2.3 – Варианты использования системы

2.3. Описание меню пользователя

Сценарий диалога в проектируемой системе разработан по принципам и стандартам визуальных интерфейсов построенных по технологии WindowsForms для операционных систем семейства Windows.

Данная технология позволяет создавать многофункциональные приложения с богатым графическим интерфейсом. При этом сценарий диалога можно представить в качестве древовидной структуры, для упрощения навигации пользователей по программному обеспечению.

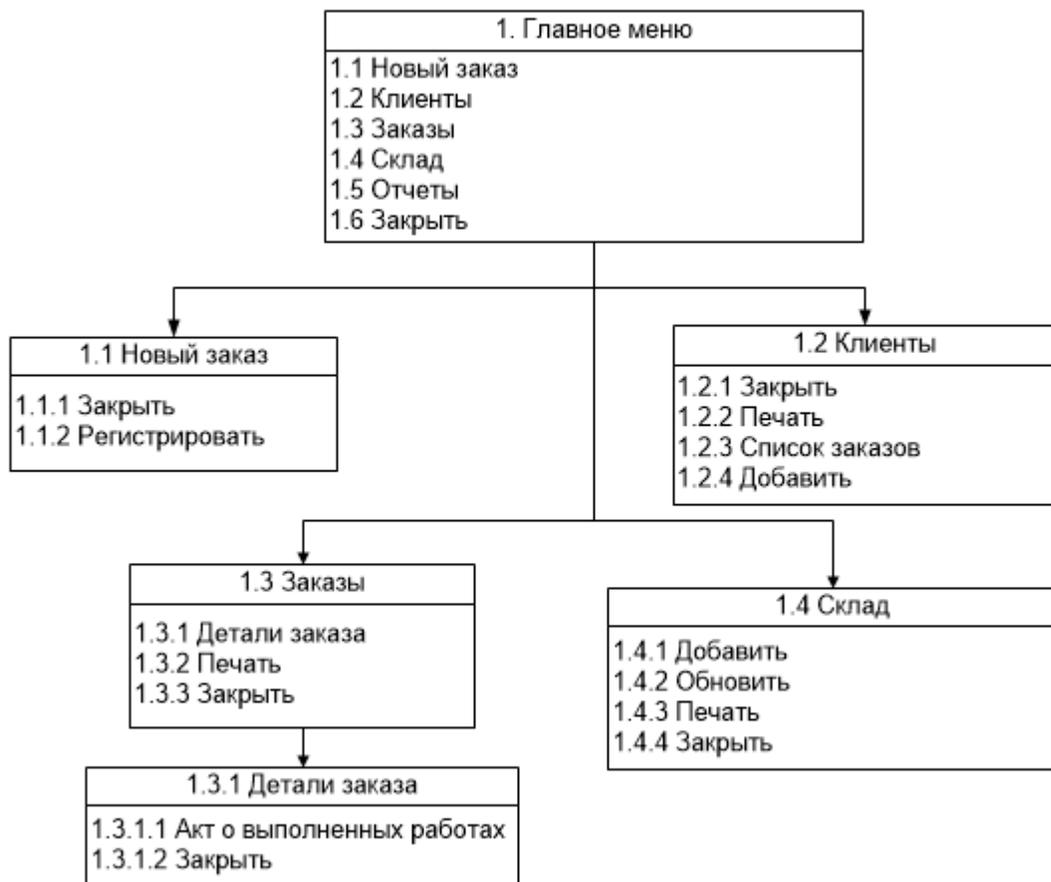


Рисунок 2.4 – Сценарий диалога программы менеджера (администратора)

На рис. 2.4 изображено дерево основного сценария диалога в системе. Но кроме него, так же присутствует еще одно, вспомогательное меню. Покажем его на рис 2.5.

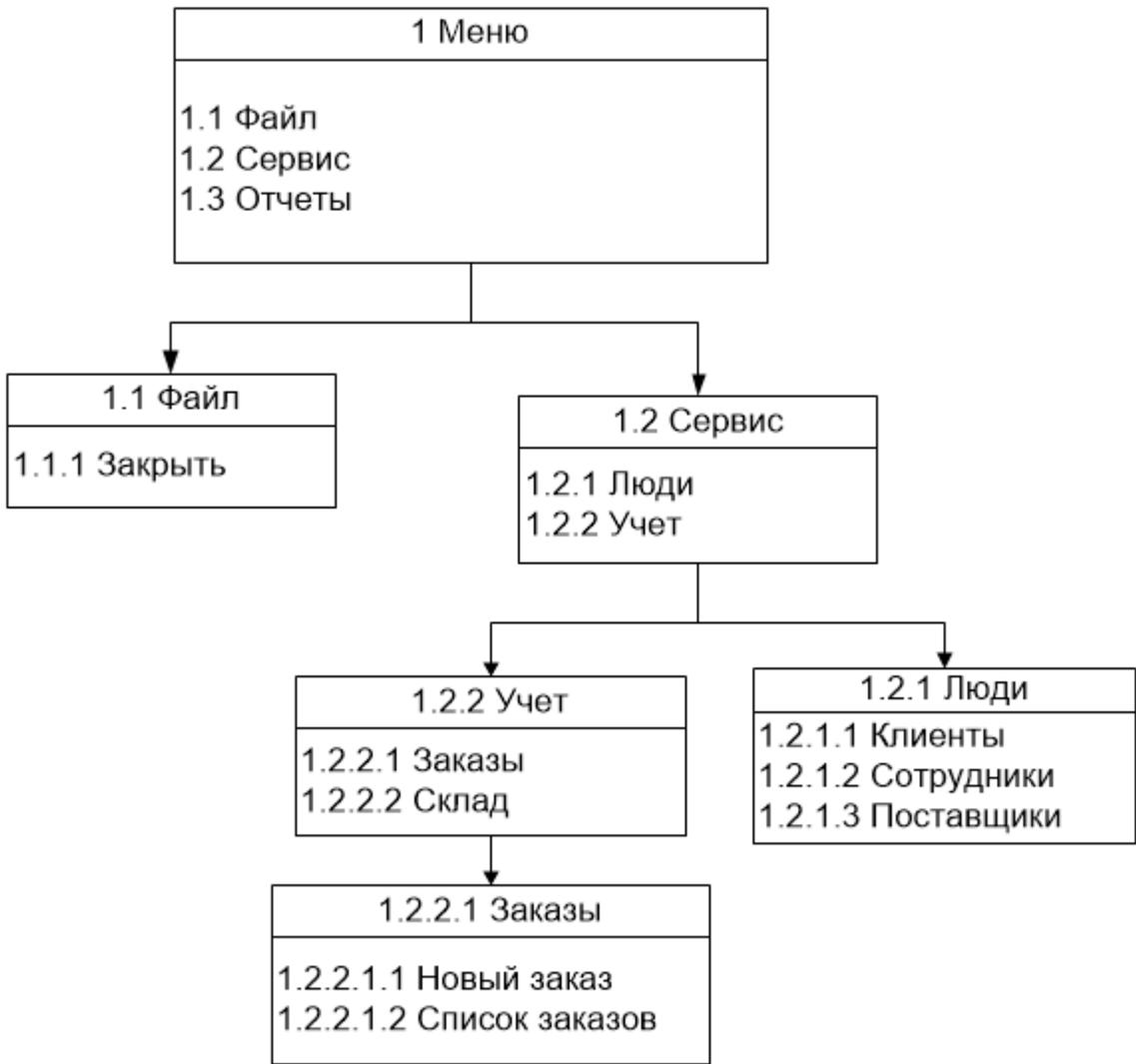


Рисунок 2.5 – Системное меню программы менеджера (администратора)

Дерево функций (Рис. 2.6) состоит из трех типов: основные, внутренние, служебные.

Основные функции – это функции, которыми может управлять пользователь при основной работе с программой.

Внутренние функции – функции, происходящие без непосредственного взаимодействия с ними пользователя.

Служебные функции – функции связанные с доступом к программе.



Рисунок 2.6 – Дерево функций

2.4. Описание меню и других элементов интерфейса

Интерфейс программного обеспечения построен на базе технологии WindowsForms. Форма Win Forms является видимой поверхностью, на которой отображается

информация для пользователя. Приложение Win Forms строится путем помещения элементов управления на форму и написанием кода для реагирования действий пользователя, такие как щелчки мыши или нажатия клавиш. Элемент управления – это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.[20]

При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления, создается событие. Приложение реагирует на эти события с помощью кода и обрабатывает события при их возникновении.

Опишем интерфейс разработанного ПО, и рассмотрим все основные моменты в ходе работы с программой.

Программа менеджера (серверная часть)

Для работы с приложением, пользователю необходимо пройти процесс авторизации, форма которой выводится сразу после загрузки (Рис. 2.6), после чего будет доступно главное окно (форма) приложения (Рис. 2.7) с расположенными на ней основными элементами управления.

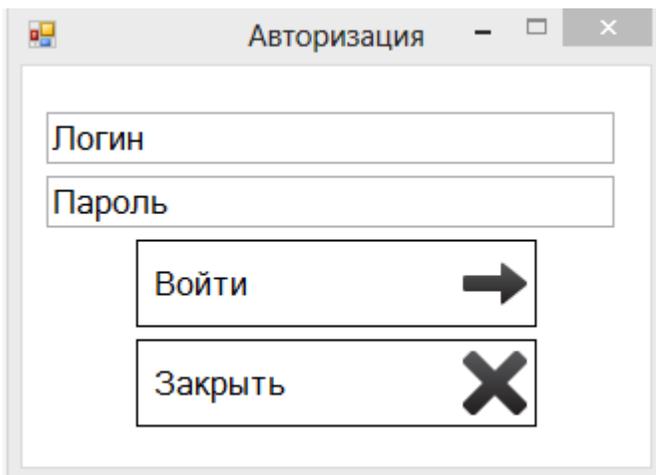


Рисунок 2.6 – Форма авторизации

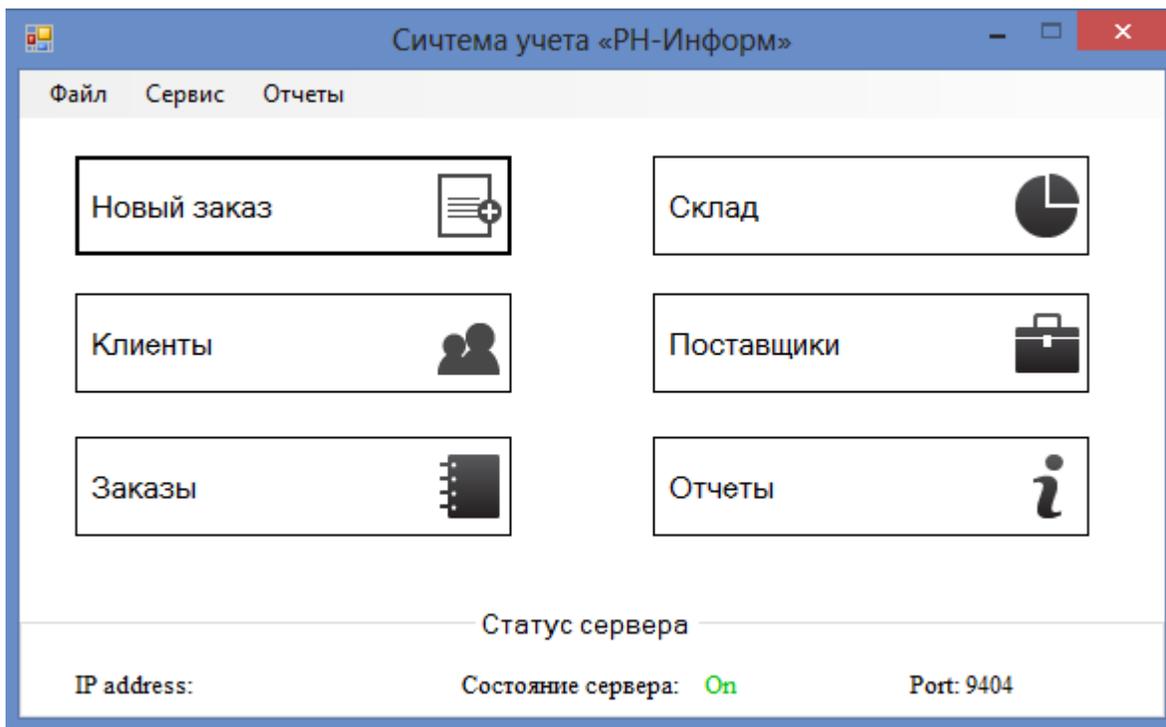


Рисунок 2.7 – Основная форма программы

Как видно на рис. 2.7, основная форма имеет шесть «горячих» объектов управления: «Новый заказ», «Клиенты», «Заказы», «Склад», «Поставщики», «Отчеты». Они так же дублируются в верхнем сервисном меню. Рассмотрим каждую вкладку, в которую позволяют перейти эти элементы в отдельности.

При переходе по пункту меню «Новый заказ», откроется форма-бланк добавления нового заказа в систему (Рис. 2.8), в котором будут доступны такие поля ввода: Заказчик, Тип обслуживания, Вид проблемы, Тип техники, Наименование техники, Серийный номер и комментарий от заказчика.

Система учета - Новый заказ

Заказчик:

Тип обслуживания:

Вид проблемы:

Тип техники:

Наименование техники:

Сарийный номер техники:

Комментарий:

Закрыть  Регистрировать 

Рисунок 2.8 – Форма добавления нового заказа

Пункт меню главной формы «Клиенты» позволяет перейти на форму, отображающую полный список клиентов обратившихся на предприятие и зарегистрированных в системе. В данном меню можно просмотреть список заказов по каждому клиенту или добавить нового клиента в систему (Рис. 2.10). Так же есть возможность поиска по фамилии клиента.

Сичтема учета - Список клиентов

1 для 5 | Поиск по фамилии: | Обновить таблицу

Имя	Фамилия	Отчество	Дата рождения	Адрес	Номер телефона
Дмитрий	Иванович	Петров	21 июня 1990 г.	ул. Первомайск...	(072) 456-5335
Александр	Потапенко	Пушкарев	19 апреля 2015 г.	-	(078) 225-8275
Филип	Филипов	Филипенко	20 июля 1997 г.	-	(073) 744-2332
Антонина	Брукская	Олеговна	14 июня 1944 г.	пр. Правды 15/2	(078) 424-3457
Анжелла	Пирожкова	Дмитриевна	22 апреля 2015 г.	пр. Волги 2а	(078) 624-2344

Закреть | Печать | Список заказов | Добавить

Рисунок 2.9 – Форма списком клиентов

Сичтема учета - добавление

Имя *

Фамилия *

Отчество *

Дата рождения 22 апреля 2015 г.

Телефон * () - -

Адрес

Компания

Отмена ↓ | Ок ✓

Рисунок 2.10 – Форма-бланк добавления нового клиента

В форме добавления клиентов звездочкой отображены обязательные параметры для заполнения, без которых клиент не будет добавлен в систему и базу данных.

Компания указывается в том случае, если клиент является лишь ее представителем и осуществляет заказ от нее.

Форма заказов, на которую можно перейти с помощью пункта меню «Заказы», позволяет пользователю просмотреть полный список зарегистрированных заказов в системе, а так же позволяет осуществлять фильтрацию данных по различным основным категориям, таким как: тип обслуживания, тип техники, дата регистрации, статус выполнения.

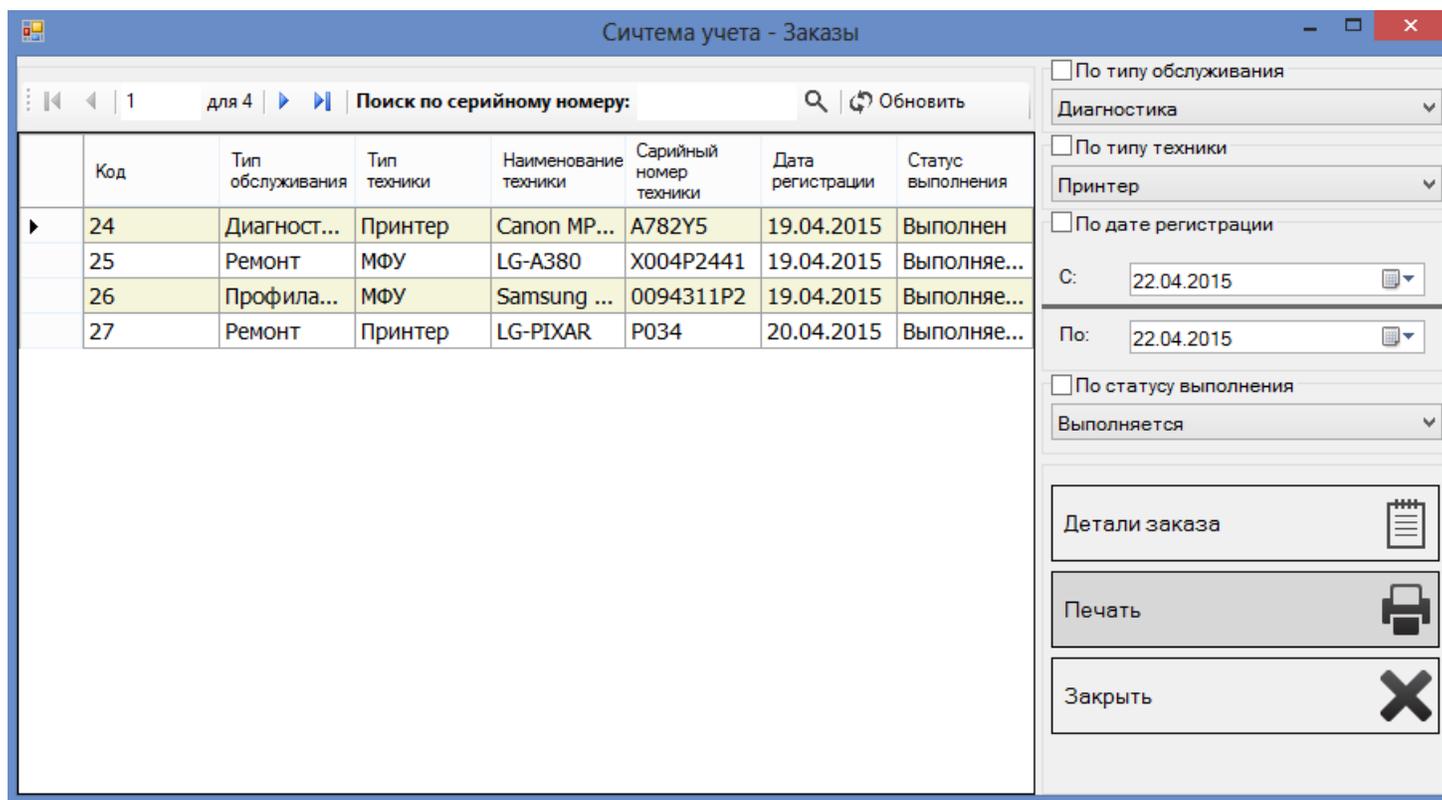


Рисунок 2.11 – Форма списка заказов

Если перейти по пункту «Детали заказа» данного меню (перед этим выбрав сам заказ со списка), можно попасть на форму, в котором описывается вся информация по выбранному заказу (Рис 2.12).

Сичтема учета - Заказ

Код	24
Заказчик	Иванович Дмитрий
<hr/>	
Тип обслуживания	Диагностика
Вид проблемы	-
<hr/>	
Тип техники	Принтер
Наименование техники	Canon MP-540
Сарийный номер техники	A782Y5
<hr/>	
Дата регистрации	19.04.2015
Заказ принял	Санторанский Картошка Импера
Статус выполнения	Выполнен

Комментарий
Полная диагностика

Закреть 

Акт о выполненных работах 

Рисунок 2.12 – Полная информация о заказе

Пункт меню «Склад» открывает форму с информацией о товаре в складском помещении предприятия. Так же предоставляет возможность по добавлению или обновлению информации.

Система учета - Склад

1 для 7 | Поиск по наименованию | По комплектующим: Картриджи для мѣ

Тип комплектующих	Наименование	Количество	Номенал (шт/мл)	Цена за единицу	Поставщик
▶ Картриджи для матр...	Ribbon cartridge LQ-50	10	шт.	769	ООО "Магнус"
Картриджи для матр...	Картридж EPSON A3 ...	15	шт.	996	ООО "Карт-Идж"
Чернила и чернильни...	ColorPix PP-202	17	шт.	305	ООО "Карт-Идж"
Чернила и чернильни...	СНПЧ Canon MP-500	22	шт.	415	ООО "Магнус"
Чернила и чернильни...	Чернила AA-202 "Че...	1000	мл.	798	ООО "Магнус"
Чернила и чернильни...	Чернила AA-202 "Же...	850	мл.	640	ООО "Карт-Идж"
Чернила и чернильни...	Чернила AA-202 "Же...	500	мл.	453	ООО "Магнус"

Тип компл.: Картриджи для матричных | Поставщик: ООО "Магнус"

Наименование: Ribbon cartridge LQ-50 | Количество: | шт.

Производитель: EPSON | Цена (ед): | руб.

Добавить

Обновить

Печать

Закрыть

Рисунок 2.13 - Меню склада

Меню «Поставщики» отображает информацию, связанную с зарегистрированными в фирме (и системе) поставщиками комплектующих на предприятие.

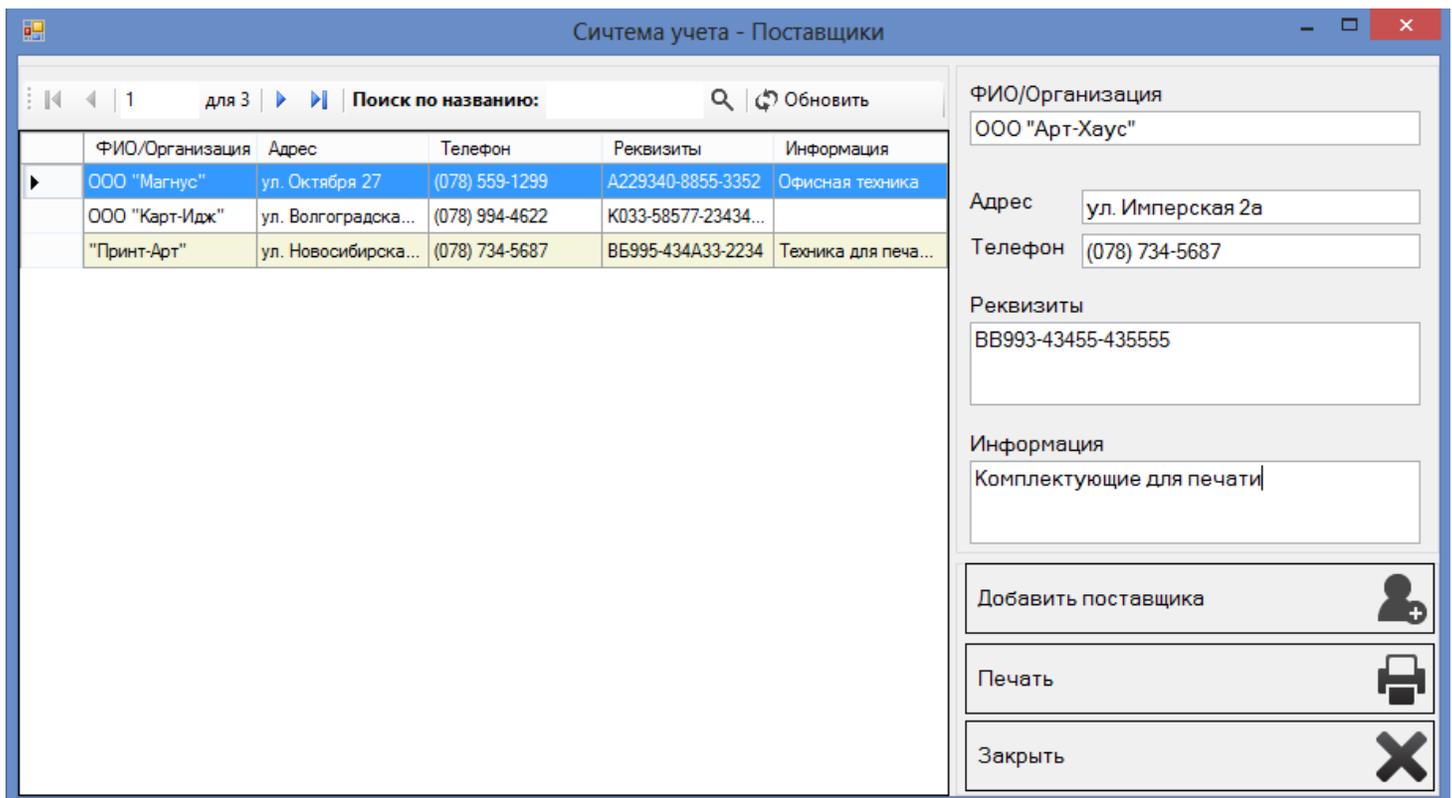


Рисунок 2.14 – Форма «Поставщики»

Программа сотрудника технического отдела (клиентская часть)

При открытии клиентской части программы, как и в случае с серверной, происходит процесс авторизации пользователя, после чего будет доступна форма со список имеющиеся заказов на предприятии и заказов, которые ведет лично сотрудник.

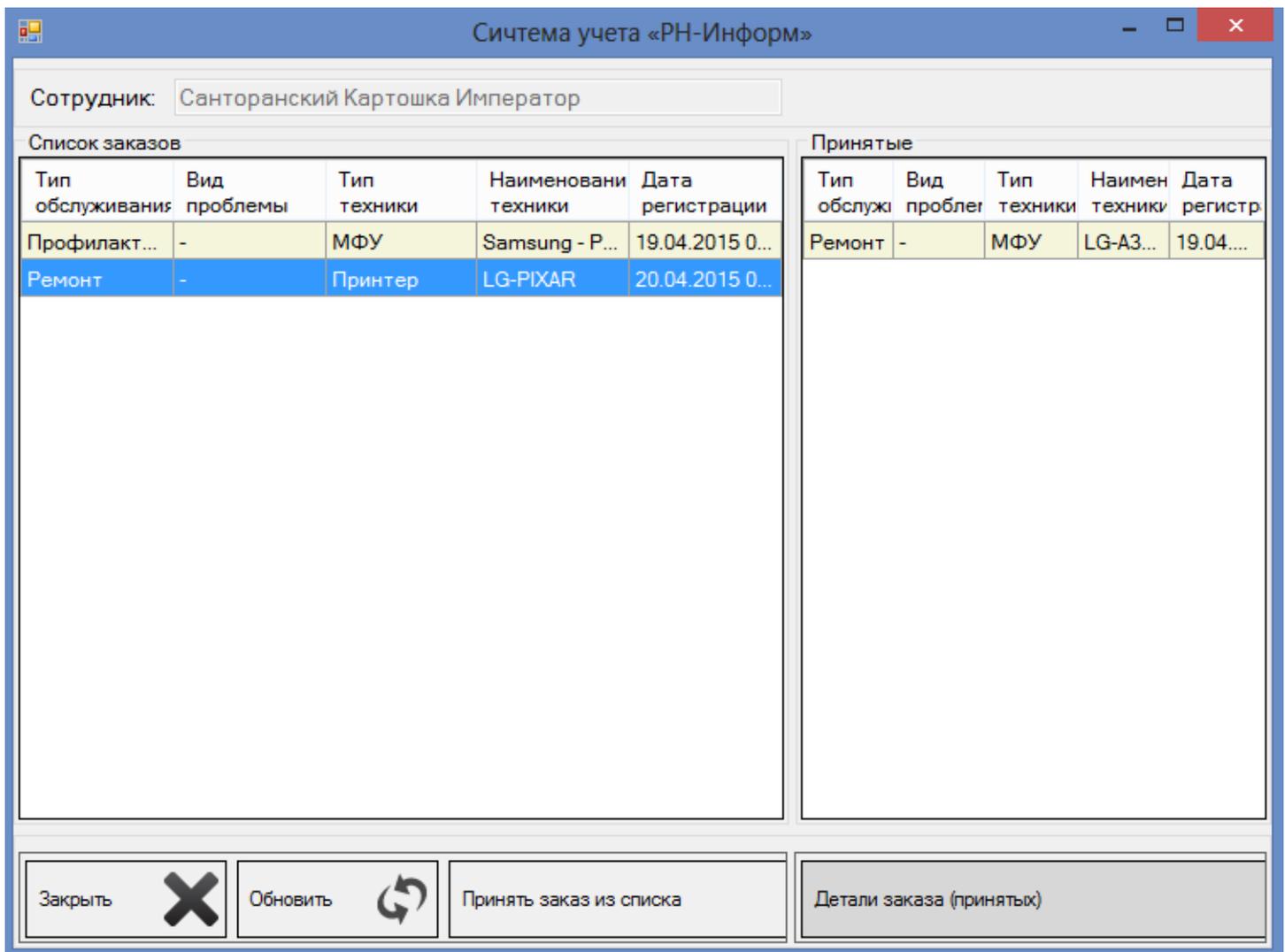


Рисунок 2.15 – Главная форма клиентской части программы

По выполнению ремонтно-профилактических работ, сотрудник должен сформировать акт о выполненных работах:

Система учета - Акт о работах

Код	Тип комплектующих	Наименование	Количество	Номенал (шт/мл)	Цена за единицу	Поставщик
1	Картриджи для мат...	Ribbon cartridge LQ-50	10	шт.	769	ООО "Магнус"
2	Картриджи для мат...	Картридж EPSON A3 LQ-...	15	шт.	996	ООО "Карт-Идж"
5	Чернила и черниль...	ColorPix PP-202	17	шт.	305	ООО "Карт-Идж"
6	Чернила и черниль...	СНПЧ Canon MP-500	21	шт.	415	ООО "Магнус"
7	Чернила и черниль...	Чернила AA-202 "Черный"	940	мл.	1	ООО "Магнус"
8	Чернила и черниль...	Чернила AA-202 "Желтый"	240	мл.	12	ООО "Карт-Идж"

№	Наименование работы (услуги)	Количество	Ед. изм.	Цена	Сумма
1	Ремонт	-	-	420	
2	Чернила AA-202 "Ж...	60	мл.	78	
3	Чернила AA-202 "Ч...	60	мл.	60	
4	СНПЧ Canon MP-500	1	шт.	415	973

Тип компл.:

Наименование:

Количество: /

Заказ №:

Заказчик:

Общая сумма:

Рисунок 2.16 – Форма-бланк акта о выполненных работах

2.7. Схема данных

Для разрабатываемой системы сущностями физической модели данных являются: «Аккаунт», «Акт», «Вид проблемы», «Заказ», «Заказчик», «Комплектующие», «Поставщик», «Производитель», «Склад», «Сотрудник», «Тип комплектующих», «Тип обслуживания», «Тип техники».

Рассмотрим детально только некоторые таблицы

Таблица 2.2 – Сущность «Аккаунт»

Имя поля Тип данных

Код Счетчик

логин Текстовый

Пароль Текстовый

Сотрудник Числовой

Статус логический

В сущности «Авторизация» хранятся пара логин/пароль пользователя, а так же установленные права доступа. При попытке авторизации программа будет сравнивать поступившие логин/пароль с уже существующими данными в этой сущности.

Таблица 2.3 - Сущность «Акт»

Имя поля	Тип данных
Код	Счетчик
№	Текстовый
Наименование работы (услуги)	Текстовый
Количество	Текстовый
Ед. изм.	Текстовый
Цена	Денежный
Сумма	Денежный

Заказ	Числовой
Заказчик	Текстовый
Дата	Дата/время

Сущность «Акт» содержит информацию об акте выполненных работ по конкретному заказу.

Таблица 2.4 - Сущность «Заказ»

Имя поля	Тип данных
Код	Счетчик
Заказчик	Числовой
Обслуживания	Числовой
Вид проблемы	Числовой
Тип техники	Числовой
Наименование техники	Текстовый
Серийный номер техники	Текстовый
Комментарий	Поле МЕМО
Дата регистрации	Дата/время

Статус выполнения	Текстовый
-------------------	-----------

Заказ принял	Числовой
--------------	----------

Сущность «Заказы» содержит информацию о заказе, которую вносит менеджер в программу. Каждый заказ привязывается к клиенту (заказчику). Эта таблица является связующей во всей БД системы. Через нее осуществляется основной функционал ПО. Общая схема данных представлена на рисунке 2.17

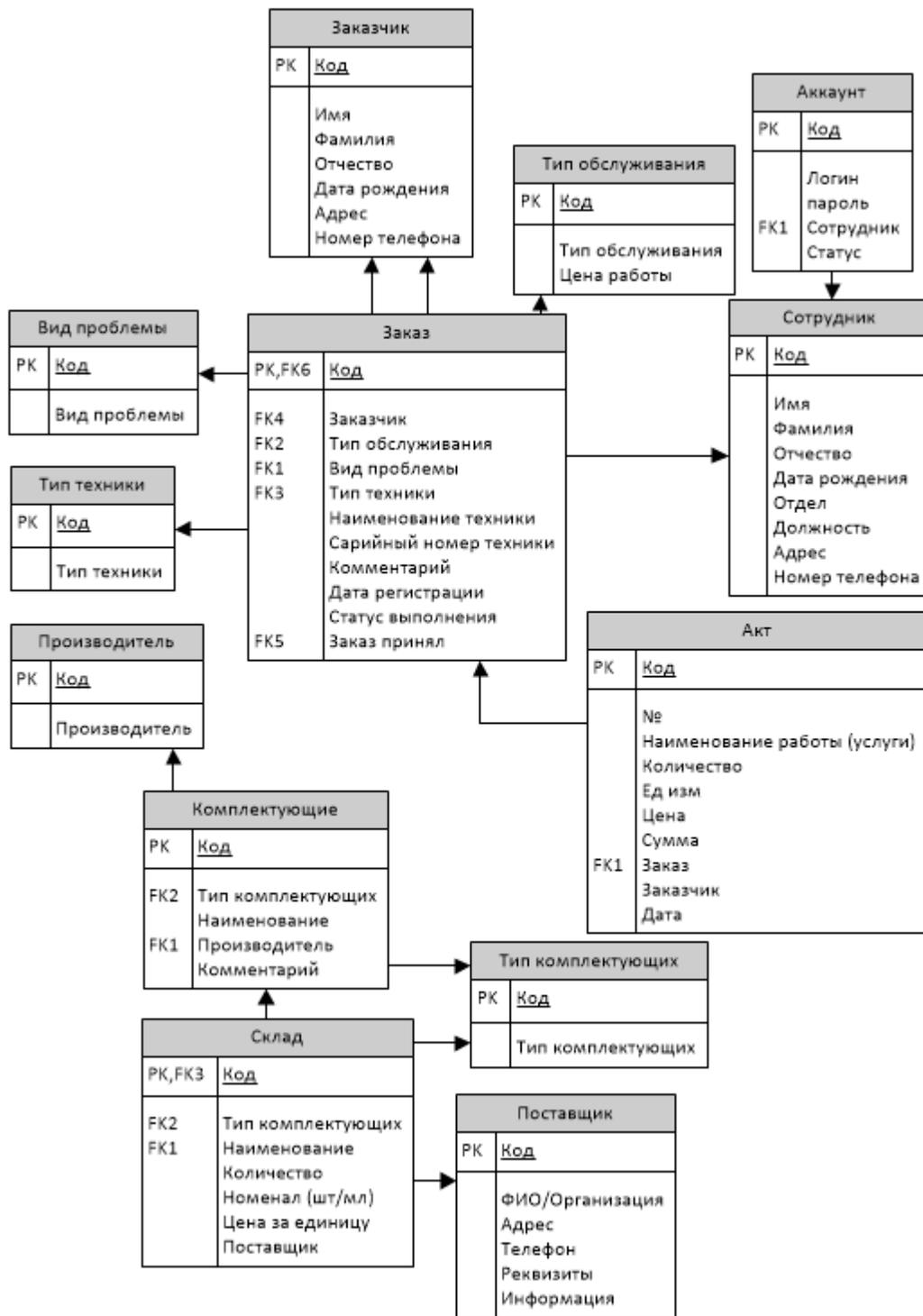


Рисунок 2.17 – Физическая схема базы данных

ЗАКЛЮЧЕНИЕ

Интерфейс пользователя - совокупность средств для обработки и отображения информации, максимально приспособленных для удобства пользователя; в графических системах интерфейс реализуется многооконный режим, изменениями цвета, размера, видимости (прозрачность, полупрозрачность, невидимость) окон, их расположением, сортировкой элементов окон, удобными настройками, как самих окон, так и отдельных их элементов (файлы, папки, ярлыки, шрифты и т.п.), доступностью многопользовательских настроек.

Особое место и внимание в интерфейсе пользователя традиционно уделяется его эффективности и удобства пользования (usability).

Понятный, удобный, дружелюбный - его основные характеристики.

Проектирование взаимодействия возникло на стыке многих дисциплин, но, несмотря на это, имеет свой уникальный набор задач и методов. В нем присутствуют элементы графического дизайна, информационного дизайна, понятия HCI, что является основой для проектирования взаимодействия с компьютеризированными системами, однако, проектирование взаимодействия невозможно свести ни к графическому дизайну, ни к архитектурному проектированию. Это самостоятельная задача, имеющая свои подходы и методологию решения.

Общая структура пользовательского интерфейса определяет структуру взаимодействия с пользователем в целом - от расположения функциональных элементов на экране до интерактивного взаимодействия, включая визуальный язык, который используется для представления данных, концепций, функциональности. По Алану Куперу, форму и поведение необходимо проектировать как тандем:

- инфраструктура интерфейса включает в себя инфраструктуру взаимодействия,
- визуальную структуру, а иногда и физическую, то есть инфраструктуру взаимодействия с физическим устройством.

Факторы, которые могут влиять на эффективность использования, разделены на три основные группы: состояние и предыдущий опыт пользователя, свойства системы и контекст использования (то есть ситуация).

Внешнее представление и поведение цифрового продукта должны отвечать способу его использования, а не личным взглядам проектировщиков. Качественные

исследования помогают изучать взгляды, потребности, поведение потенциальных пользователей, предметную область, способы использования и прочее.

Визуальный дизайн фокусируется на эстетике программы и связанных материалов: на стратегической реализации изображений, цветов, шрифтов и других элементов.

Интерфейс и основные требования к нему менялись в соответствии с изменениями и развитием ИТ в целом. Интерфейс прошел длинный путь - от примитивного текстового списочного варианта до сложных адаптивных интерфейсов с элементами ИИ и экспертных подходов.

В работе рассмотрены основные понятия, связанные с разработкой интерфейса и меню как одного из основных интерфейсных элементов.

Рассмотрены фазы развития и изменения типов меню на протяжении последних десятилетий.

На конкретном практическом примере проиллюстрированы фазы проектирования и реализации меню.

Работа полностью соответствует тематике, цели и задания, поставленные в работе, выполнены в полном объеме.

ЛИТЕРАТУРА

1. 10 user interface design fundamentals [Электронный ресурс]. – Режим доступа: <http://blog.teamtreehouse.com/10-user-interfacedesign-fundamentals>
2. Wikipedia, the free encyclopedia. Human-computer interaction [Электронный ресурс]. – Режим доступа: http://en.wikipedia.org/wiki/Human-computer_interaction
3. Алан Купер об интерфейсе. Основы проектирования взаимодействия. – Пер.с англ. – СПб.: Символ'Плюс, 2009. – 688 с.
4. Основы дизайна интерфейса [Электронный ресурс]. – Режим доступа: http://froland.ru/samodel/vbguide/ch6_8_1.html
5. Проектирование и дизайн интерфейсов [Электронный ресурс]. – Режим доступа: <http://www.uimodeling.ru/process/user-interfacedesign.html>
6. 10 Interface Design Fundamentals [Электронный ресурс]. – Режим доступа: <http://blog.teamtreehouse.com/10-user-interface-designfundamentals>

7. Wikipedia, the free encyclopedia. Usability [Электронный ресурс]. – Режим доступа: <http://en.wikipedia.org/wiki/Usability>
8. Проектування користувачького інтерфейсу [Электронный ресурс]. – Режим доступа: <http://www.simulation.kiev.ua/dbis/lection17.html>
9. Wikipedia, the free encyclopedia. Graphic Design [Электронный ресурс]. – Режим доступа: http://en.wikipedia.org/wiki/Graphic_design
10. Разработка интерактивных сайтов [Электронный ресурс] – Режим доступа – <http://webstudio2u.net/ru/site-develop/444-interactivesite.html>
11. Designing Interactive Systems: People, Activities, Contexts, Technologies. [Электронный ресурс] / David Benyon, Dr. Phil Turner, 46 Dr. Susan Turner. – Режим доступа – http://books.google.com.ua/books?id=iWe7VkfW0zMC&pg=PA2&hl=ru&source=gbs_toc_r&cad=3#v=onepage&q&f=false
12. Digital Media: Technological and Social Challenges of the Interactive World [Электронный ресурс] / Edited by Megan A. Winget and William Aspray – Режим доступа – http://books.google.com.ua/books?id=boS5Vfh80osC&printsec=frontcover&dq=Digital+interactive+media&hl=ru&sa=X&ei=5O2PUKbnIcjGtAaqjYCYAw&redir_esc=y
13. Wikipedia, the free encyclopedia. WordPress [Электронный ресурс]. – Режим доступа: <http://en.wikipedia.org/wiki/WordPress>
14. WordPress Lessons [Электронный ресурс]. – Режим доступа: [http://codex.wordpress.org/WordPress_Lessons#Designing_Your_WordP ress_Site](http://codex.wordpress.org/WordPress_Lessons#Designing_Your_WordPress_Site)
15. Интерфейс и создание первых записей WordPress [Электронный ресурс]. – Режим доступа: <http://wp-admin.com.ua/interfeys-isozdanie-pervyih-zapisey-wordpress/>
16. Основы проектирования взаимодействия в электронных системах / Войтенко О.А. [Электронный ресурс]. – Режим доступа: <http://conference.ikto.net/>
17. Гвоздева Т. В., Б. А. Баллод, Проектирование информационных систем, М, Издательство: Феникс, 2009 г., 512 стр.
18. Пирогов В. Ю., Информационные системы и базы данных. М, Организация и проектирование, Издательство: БХВ-Петербург, 2009 г.528 стр.
19. Чипига А. Ф., Информационная безопасность автоматизированных систем, М, Издательство: Гелиос АРВ, 2010 г., 336 стр.
20. Пирогов В. Ю., Информационные системы и базы данных. М, Организация и проектирование, Издательство: БХВ-Петербург, 2009 г.528 стр.

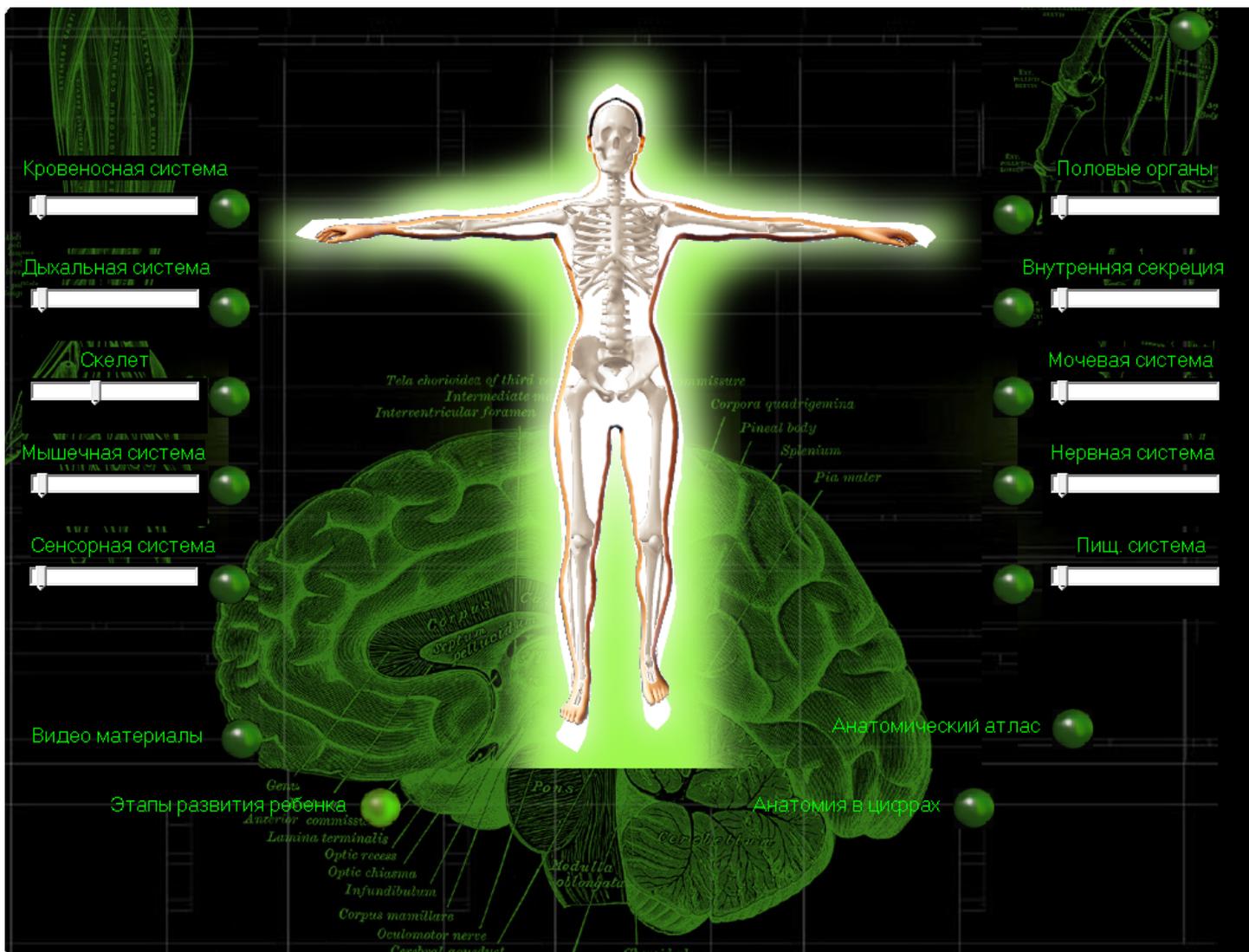
ПРИЛОЖЕНИЕ

О программе

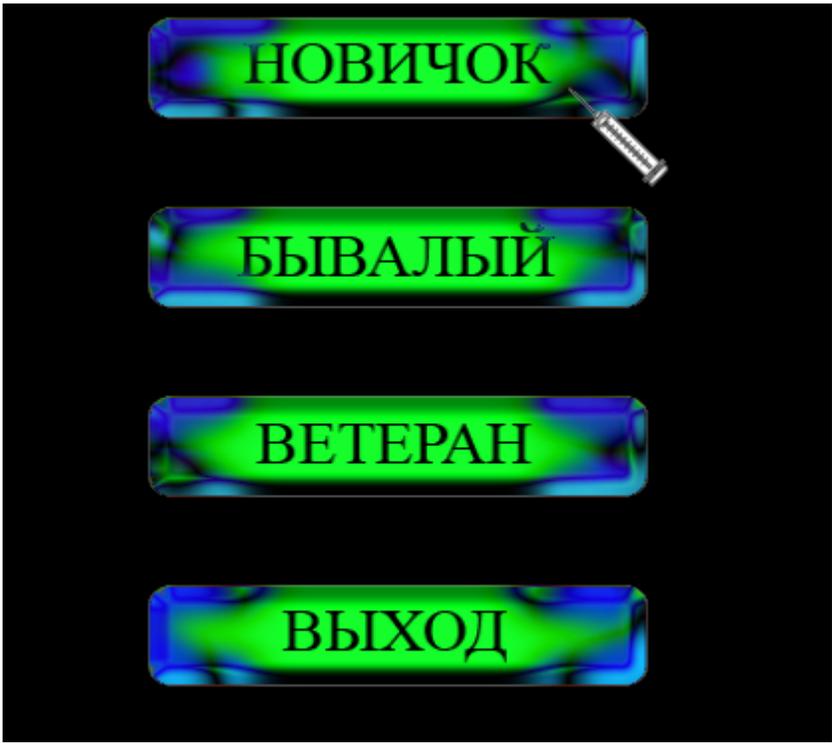
Электронная энциклопедия "Анатомия"



Далее



Игровая программа для обучения устному счету C++, OpenGL



Количество
20
Осталось
20
Ошибки
0
t : 36

9
1

0

ВЫХОД

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

+

-

Приложение 2 Диаграмма развертывания разработанного ПО

