

## Содержание:

## Введение

В нынешнее время сложилась ситуация, что во многих отраслях производства, экономики, услуг проводится автоматизация их работы, документооборота, вводятся новые принципы обмена информацией.

Если в системе управления технологическими процессами объектом для управления являются отдельные комплексы технологического оборудования, то в различных организационно-экономических системах это, как правило, коллективы людей, которые взаимодействуют с технологическим оборудованием.

Большое распространение получает автоматизация различных процессов, происходящих в организации: автоматизация документооборота бизнес-процессов отдела продаж, обмен корреспонденцией и прочее. При этом, несмотря на повышение компьютеризации в обществе, в некоторых сферах до сих пор не используются средства, которые позволяют хорошо автоматизировать процессы сбора и обобщения информации.

**Целью данной работы** является разработка интерфейса для автоматизированной системы учета технического обслуживания компьютерной техники.

Для достижения поставленной цели нужно выполнить ряд задач:

- ○ рассмотреть теоретические основы построения интерфейса: понятие, виды, структуру и т.д.;
- постановка задачи;
- выбор варианта и средств построения интерфейса;
- проектирование интерфейса системы.

В первом разделе дана рассмотрены теоретические основы разработки интерфейса. Дано понятие интерфейса, рассмотрено его назначение, виды интерфейса.

Во второй главе проведен анализ предметной области, дана постановка задачи, обоснована необходимость проектирования программного обеспечения и его интерфейса. Выбран вариант построения интерфейса, средства его разработки.

В третьей части работы описаны этапы построения интерфейса. Приведены результаты проектирования.

1.

## **Интерфейс программ: понятие, виды, структура**

1.

### **Понятие программного интерфейса и его назначение**

Взаимодействие оператора с вычислительной машиной является важным звеном вычислительного процесса при решении различных прикладных задач как научного, так и производственного плана. Создание программ в области организации рыночных отношений при создании информационных сайтов различных организаций и предприятий, при создании программ управления производственными процессами, учета выпускаемой продукции и ее реализации, управления качеством и даже при такой задаче, как сортировка электронной почты секретарем, требуется разработка удобного для пользователя взаимодействия с ЭВМ [6].

Проектирование – итерационный процесс, при помощи которого требования к ПС транслируются в инженерные представления ПС. Обычно в проектировании выделяют две ступени: предварительное проектирование и детальное проектирование. Предварительное проектирование формирует абстракции архитектурного уровня, детальное проектирование уточняет эти абстракции. Кроме того, во многих случаях выделяют интерфейсное проектирование, цель которого – сформировать графический интерфейс пользователя (GUI). Схема информационных связей процесса проектирования приведена на рисунке 1.

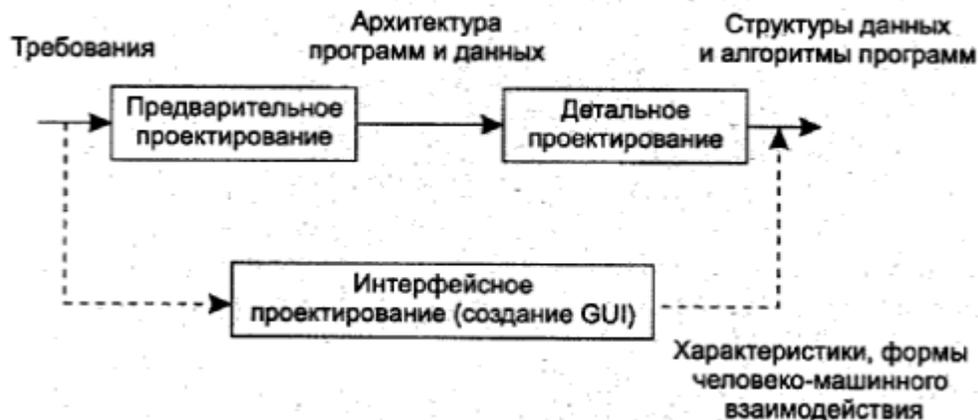


Рис.1.1. Схема информационных связей процесса проектирования

**Интерфейс** – это способ общения пользователя с персональным компьютером, пользователя с прикладными программами и программ между собой. Интерфейс служит для удобства управления программным обеспечением компьютера [8].

Ряд важнейших программ, например, все виды DOS, запускаются автоматически при включении компьютера, другие с помощью файлов autoexec.bat или config.sys (различные драйверы). Ряд программ могут при необходимости автоматически загружаться при запуске оболочек Windows.

**Интерфейс** – это в общем случае набор правил, согласно которым взаимодействуют два объекта и более.

- 1. **Виды интерфейса**

Интерфейсы различают по таким характеристикам, как структура связей, способ подключения и передачи данных, принципы управления и синхронизации.

1. Внутримашинный интерфейс – система связи и средств сопряжения узлов и блоков ЭВМ между собой. Внутримашинный интерфейс представляет собой совокупность электрических линий связи (проводов), схем сопряжения с компонентами компьютера, протоколов (алгоритмов) передачи и преобразования сигналов.

Различают два варианта организации внутри машинного интерфейса [12]:

- многосвязный интерфейс, при котором каждый блок ПК связан с другими блоками своими локальными проводами;
- односвязный интерфейс, в результате которого все блоки ПК связаны друг с другом через общую или системную шину.

2. Внешний интерфейс – система связи системного блока с периферийными устройствами ЭВМ или с другими ЭВМ

Здесь можно выделить также несколько типов внешнего интерфейса:

- интерфейс периферийных устройств, подключаемых с помощью шин ввода-вывода (ISA, EISA, VLB, PCI, AGP, USB IEEE 1384 SCSI и др.);
- сетевой интерфейс, типа одноранговой сети или сети клиент-сервер с топологиями типа звезда, кольцевая или шинная.

3. Интерфейс «человек-машина» или интерфейс «человек-компьютер» или пользовательский интерфейс – это способ, которым вы выполняете какую-либо задачу с помощью каких-либо средств (какой-либо программы), а именно совершаемые вами действия и то, что вы получаете в ответ [14].

Интерфейс является ориентированным на человека, если он отвечает нуждам человека и учитывает его слабости.

**Машинная часть интерфейса** – часть интерфейса, реализованная в машине (аппаратно-программной ее части) с использованием возможностей вычислительной техники.

**Человеческая часть интерфейса** – это часть интерфейса, реализуемая человеком с учетом его возможностей, слабостей, привычек, способности к обучению и других факторов.

Наиболее распространенные интерфейсы определены государственными и международными стандартами.

В дальнейшем изложении будет рассматриваться только интерфейс пользователя.

### **Классификация интерфейсов пользователя**

Как было указано выше интерфейс – это, прежде всего набор правил, которые можно объединить по схожести способов взаимодействия человека с компьютером.

Различают три вида интерфейсов пользователя: командный, WIMP и SILK – интерфейсы.

Взаимодействие перечисленных интерфейсов с операционными системами и технологиями показано на рис.1.2:

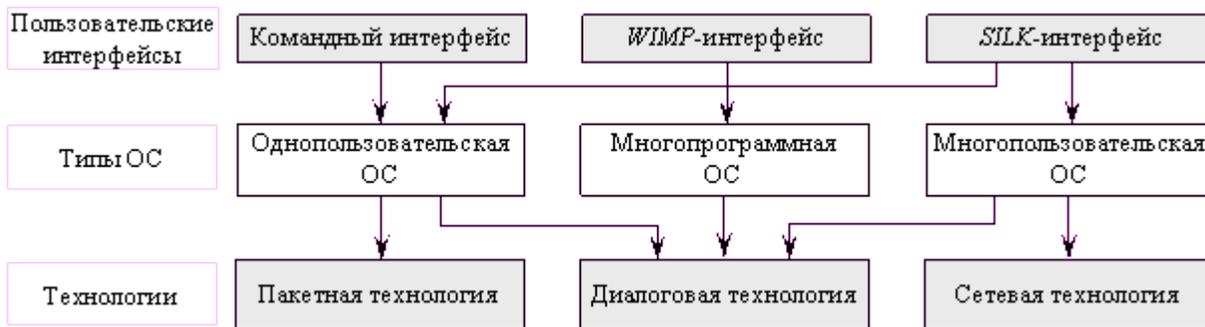


Рис. 1.2. Взаимодействие интерфейсов пользователя их технологий и операционных систем

1. Командный интерфейс, при котором взаимодействие человека с компьютером осуществляется путем подачи компьютеру команд, которые он выполняет и выдает результат пользователю. Командный интерфейс может быть реализован в виде пакетной технологии и технологии командной строки. В настоящее время пакетная технология практически не используется, а технология командной строки можно встретить в виде резервного способа общения человека с компьютером.

2. ООМУ (окно, образ, меню, указатель) WIMP (window, image, menu, pointer) - интерфейс. Характерной чертой этого интерфейса является то, что диалог пользователя с компьютером ведется не с помощью командной строки, а с помощью окон, графических образов меню, курсора и других элементов. Хотя в этом интерфейсе подаются команды машине, но это делается через графические образы.

3. РОЯЗ (речь, образ, язык, знания) SILK (speech, image, language, knowledge) - интерфейс. Этот интерфейс наиболее приближен к обычной человеческой форме общения. В рамках этого интерфейса идет обычный разговор человека и компьютера. При этом компьютер находит для себя команды, анализируя человеческую речь и находя в ней ключевые фразы. Результаты выполнения команд он также преобразует в понятную человеку форму. Этот вид интерфейса требует больших аппаратных затрат, поэтому находится в стадии разработки и совершенствования и используется пока только в военных целях.

SILK- интерфейс для общения человека с машиной использует [15]:

- речевую технологию;
- биометрическую технологию (мимический интерфейс);
- семантический (общественный) интерфейс.

**Речевая технология** появилась в середине 90-х годов после появления недорогих звуковых карт и широкого распространения технологий распознавания речи. При этой технологии команды подаются голосом путем произнесения специальных стандартных слов (команд), которые должны выговариваться четко, в одном темпе с обязательными паузами между словами.

**Биометрическая технология («Мимический интерфейс»)** возникла в конце 90-х годов и в настоящее время находится в стадии разработки. Для управления компьютером используется выражение лица, направление взгляда, размер зрачка и другие признаки человека. Для идентификации пользователя используется рисунок радужной оболочки его глаз, отпечатки пальцев и другая уникальная информация, которая считывается с цифровой камеры, а затем с помощью программы распознавания образов из этого изображения выделяются команды.

**Семантический (общественный) интерфейс** возник еще в конце 70-х годов XX века, с развитием искусственного интеллекта. Его трудно назвать самостоятельным видом интерфейса, так как он включает в себя и интерфейс командной строки, и графический, и речевой, и мимический интерфейсы. Основной его особенностью является отсутствие команд при общении с компьютером. Запрос формируется на естественном языке, в виде связанного текста и образов. По сути - это моделирование общения человека с компьютером. В настоящее время используется для военных целей. Такой интерфейс крайне необходим в обстановке ведения воздушного боя.

- **1. Варианты построения интерфейсов**

Можно рассматривать два совершенно разных метода создания интерфейсов.

1) Вручную. В таком подходе пишется код, который отвечает за создание элементов интерфейса, обрабатывающие пользовательские события. Это не всегда быстро, но такой способ дает максимальный контроль создания интерфейса и обработки событий.

2) Графической редактор. В данном подходе используется программный продукт, который позволяет нарисовать интерфейс, мгновенно наблюдая то, как он будет отображаться. Данный метод гораздо нагляднее, но почти всегда менее гибок.

На основе собранных пользовательских и иных требований проектировщики создают макеты будущего интерфейса в виде так называемых прототипов, которые графически представляют внешний вид интерфейса. Неотъемлемой частью

прототипа является описание поведения интерфейса, возникающее в процессе взаимодействия пользователя с продуктом, либо эмуляция поведения продукта.

На основе такой спецификации дизайнеры создают графический стиль продукта, а разработчики его реализуют. У каждого участника разработки имеется собственная зона ответственности и компетенции.

Обмен информацией обеспечивается с помощью результатов труда каждого специалиста и происходит по схеме, представленной на Рисунке 1.3.

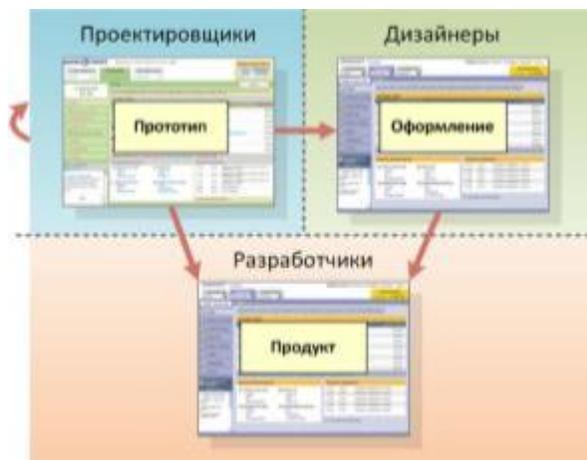


Рис. 1.3. Основные потоки взаимодействия при классическом подходе проектирования пользовательского интерфейса

Данный подход отлично работает для обычных веб-приложений и для классических настольных (GUI) приложений, где давно устоялись свои модели взаимодействия, построенные на основе фиксированного набора элементов управления.

Однако данный подход плохо работает для современных Ajax приложений и особенно для RIA. В настоящий момент происходит стремительное слияние настольных (GUI) и веб-приложений.

Использование специализированных языков описания интерфейса и сред разработки.

Возникло решение, позволяющее отделить внешний вид пользовательского интерфейса от бизнес-логики программы. Решение основано на специальных языках описания интерфейса - XAML от Microsoft, MXML от Adobe, ZUL от Mozilla и так далее. В этих текстовых языках описывается внешний вид элементов (в векторном формате) так, что интерфейс можно создавать в любом редакторе.

В языке XAML можно задавать не только расположение элементов на экранной форме (в векторном виде), но и все трансформации, которые с ними происходят.

Для редактирования XAML лучше всего применять специальные пакеты такие как, например, Microsoft Expression Blend (Blend). Редактирование внутри него происходит в наглядной, визуальной форме, то есть проектировщик может создавать элементы, перемещать их по экрану и описывать поведение.

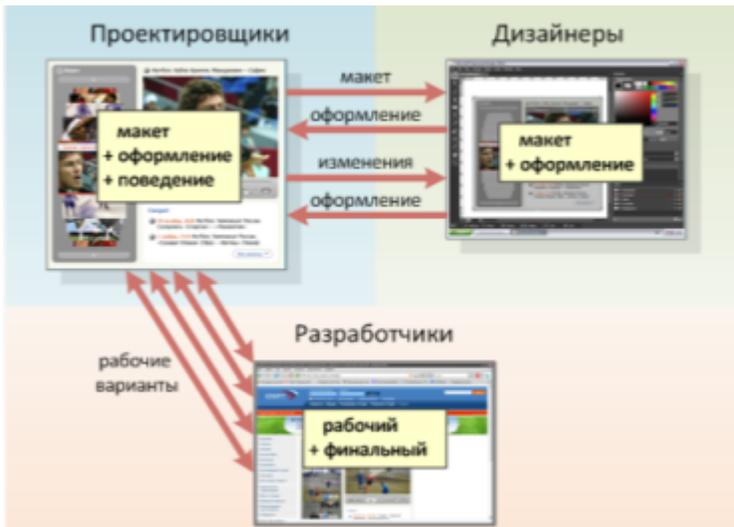


Рис.1.4. Основные потоки взаимодействия при использовании специализированных языков описания интерфейса и сред разработки

Взаимодействие между членами команды теперь происходит следующим образом:

- Проектировщик создает прототип.
- Дизайнер получает код XAML, и прямо в среде Blend или ином XAML редакторе (например, в Microsoft Expression Design) создает совершенно новый файл XAML, содержащий объекты интерфейса в конечном оформлении.
- Проектировщик снова получает код XAML от дизайнера и переделывает его так, чтобы его было удобно поддерживать далее в процессе разработки продукта. Самое главное - проектировщик создает сценарии трансформаций совершаемых с объектами, то есть определяет, как объекты будут реагировать на действия пользователя.
- Разработчик получает от проектировщика код XAML привязывает бизнес-логику и связывает все трансформации (кусочки взаимодействия).

Для эффективной работы системы и ее большого покупательского спроса недостаточно только того, чтобы аппаратура и программы обеспечивали правильные результаты - не менее важным фактором является удобство работы

пользователя. Любая программа, претендующая на то, чтобы быть «удобной для рядового пользователя» должна обладать графическим пользовательским интерфейсом.

Рассмотрим основные подходы к построению «удобного и правильного» GUI.

Работу пользовательского интерфейса программного средства можно описать в виде диаграммы на рисунке 1.5. Каждое действие пользователя ведет к изменению состояния графического интерфейса (визуально - это выставление флажков, нажатие кнопок и т.д., программно идет обращение к соответствующему обработчику данного состояния), что в свою очередь ведет к исполнению определенного модуля программы и дальнейшему изменению состояния графического интерфейса (визуализации ответа программы на действие пользователя).

В случае такого представления логики работы графического интерфейса гораздо проще проводить тестирование отдельных состояний и переходов. Поскольку на диаграмме видны все возможные входные данные, а переходы показывают, как должны развиваться события. При этом получается граф состояний пользовательского интерфейса, который достаточно легко создавать («рисую»).

Также улучшается наглядность представления логики работы графического интерфейса (по сути работы обработчиков событий), появляется возможность применения шаблонов, что, приводит к улучшению качества и скорости создания интерфейсов.



Рис. 1.5. Диаграмма работы пользовательского интерфейса

Следующим подходом к построению пользовательских интерфейсов является концепция User-Centered Design - дизайн вокруг пользователя.

Данный подход основан на концепции выстраивания дизайна на основе нужд конечного пользователя. Это позволяет оптимизировать взаимодействие человека и системы, сделать работу более эффективной. Здесь не человек адаптируется к информационной системе, а информационная система - под человека.

Также существует более традиционный подход к организации информации, базирующийся на логике, диктуемой самим информационным материалом. Этот подход заставляет пользователя подстраивать свои методы работы под организацию интерфейса программы.

1.

## Постановка задачи

1.

# Описание задачи. Цель и задачи построения интерфейса

Компания ООО УК «Логус-групп» занимается производством, хранением и реализацией сельскохозяйственной продукции. Документооборот осуществляется в электронном виде, с помощью компьютерной техники. Для устойчивой работы корпоративной сети системными администраторами поддерживается рабочее состояние компьютерной техники и оргтехники. Ими осуществляется техническое обслуживание раз в 3 месяца, либо ремонт вышедшей из строя компьютерной техники и оргтехники.

В техническое обслуживание входит:

- Чистка системных блоков от пыли;
- Замена термоинтерфейса между процессором и кулером;
- Проверка на работоспособность всех компонентов системного блока.
- Чистка оргтехники от пыли.

Информация о проведении ТО и проверок хранится на бумажных носителях (Рис.2.1), либо такая информация передается словесно, соответственно на сохранность данной информации влияет человеческий фактор, а именно:

- потеря бумажных носителей;
- человек может забыть, у какой техники последний раз делалось ТО;
- при требовании руководства отчета по проведению ТО возможна ситуация, в которой нечем подкрепить словесную информацию.

**Проведение технического обслуживания  
компьютерной техники, оргтехники, сетевого оборудования  
На территории ООО УК "Логус-групп"**

**Ответственный:** \_\_\_\_\_

**Дата начала ведения журнала:** \_\_\_\_\_

<b>№</b>	<b>ФИО пользователя</b>	<b>Наименование техники</b>	<b>Дата проведения</b>	<b>Примечание</b>

Рис.2.1. Пример документа о проведении ТО

Исходя из вышеперечисленных ситуаций, было предложено решение по оптимизации рабочего процесса по ведению документации ТО и инвентаризации.

Учет оборудования предприятия – ключевой источник получения актуальной и прозрачной информации об активах предприятия и их состоянии. Под оборудованием предприятия в нашем случае понимается компьютерная техника и оргтехника.

Для эффективного управления техническим обслуживанием и ремонтами оборудования крайне необходимо владеть наиболее полной информацией об имеющихся активах и их состоянии. Эта информация нужна не только ремонтным службам, но и другим подразделениям предприятия.

Учет оборудования является залогом своевременного проведения обслуживания и ремонтных работ, что в свою очередь позволяет продлить жизненный цикл оборудования и снизить расходы предприятия в целом.

Автоматизация учета технического обслуживания ООО УК «Логус-Групп» проводится с целью:

- минимизировать затраты на регистрацию обращений в службу технической поддержки;
- обеспечить доступность отчета по заявкам и обращениям в службу технической поддержки практически в режиме «он-лайн» с исключением временного фактора трудозатрат специалиста на его составление;
- исключить рутинные операции, отвлекающие ресурс специалиста, такие как: заполнение форм документов, контроль над комплектностью документов, поиск нужных форм на бумажном носителе, ксерокопирование, пр.;
- повысить оперативность принятия решения по обеспечению функциональности оборудования компании;
- минимизировать время на проверку отчетов по отремонтированному оборудованию и программным средствам;
- сократить издержки на непроизводственные операции;
- содействовать росту клиентов компании и объемов прибыли компании.

Цели автоматизации бизнес-процессов предполагается достичь за счет разработки и внедрения специализированных программных средств.

Назначение программы – оптимизация рабочего процесса и удобства хранения информации в организации ООО УК «Логус-групп».

Минимальные требования для разработки и работы информационной системы учёта технического обслуживания компьютерной техники:

- Процессор: от 1,5ГГц.
- Память: от 1 Гб.
- Жёсткие диски: 200 Гб.

Существующая информационная сеть имеет следующие технические показатели:

- Процессор: 3ГГц.
- Память: 2Гб.
- Жёсткие диски: 500Гб.

Следовательно, необходимости в обновлении КТС нет. Это позволит значительно снизить расходы на разработку и внедрение новой информационной системы.

- 1.

# Выбор варианта построения и средств разработки интерфейса

Кратко расскажем о наиболее важных из существующих в настоящий момент языков программирования.

1. C является одним из первых языков программирования, который стал популярными и используется до сих пор. Этот язык создан в 1972 году Деннисом Ритчи в Bell labs и является основой для многих других языков: C#, JavaScript, C++, Java, Perl. Этот язык обычно первым преподаётся в ВУЗах, так как помогает лучше понимать многие составляющие технологии программирования.

Многие популярные ресурсы в интернете помогают бесплатно освоить язык C.

1. Для своего времени язык C был хорошим, однако с течением времени ему стало не хватать объектно-ориентированности. Потребовалось реализовать эту потребность. Так, в 1983 г. был разработан более мощный язык, который получил название C++. Его часто называют объектно-ориентированной версией языка C. C++ – один из самых часто используемых императивных языков в мире. Разработчики его применяют для создания разного рода приложений: браузеры Mozilla Firefox, Google Chrome, музыкальный проигрыватель Winamp, а также продукты Adobe – все это и много другое написано на языке C++.

Также C++ используют для разработки игр и приложений для смартфонов. На рынке труда разработчики C++ востребованы и число вакансий растёт.

1. Язык Java сочетает в себе возможности C и C++ и является мощным языком создания кроссплатформенных приложений. Java, стал первым полностью объектно-ориентированным языком и функционирует по принципу «Однажды написал, используй везде». Исходный код Java можно скомпилировать однажды и затем запускать на любом устройстве. Требование к устройству – наличие установленной виртуальной машины Java (JVM).
2. C# является представителем семейства языков программирования от компании Microsoft. Он создан в 2000 году в качестве составной части платформы Microsoft .Net Framework. C# весьма близок к Java в использовании, хотя есть мнение о том, что C# сочетает в себе надёжность C++ вместе с

продвинутыми характеристиками языка Java.

На C# можно создавать почти все типы приложений, к тому же его можно использовать для написания ПО для большинства современных платформ.

1. Многие вполне обоснованно полагают, что PHP является самым мощным серверным языком, который может выполнять большой объем работы при незначительном количестве строк кода. PHP широко применяется для создания различных CMS, например, Joomla, Drupal, WordPress и др., а также для разработки статических и динамических веб-сайтов. PHP представляет собой язык с открытым кодом. Это означает, что в открытом доступе существуют тысячи модулей и решений, которые могут быть легко модифицированы для нужных программисту функций.
2. Создателем этого языка является Ларри Уолл, который разработал Perl в 1987 году для упрощения обработки различных отчетов. Неофициальный бэкроним языка – Practical Extraction and Report Language. Perl представляет собой высокоуровневый современный язык программирования общего назначения, который, в принципе, может заменить любой другой язык программирования. В данное семейство входят Perl 5, Perl 6. Языки семейства Perl позаимствовали черты таких языков, как C, AWK, sh, sed.
3. Создателем Python является Гвидо ван Россум, который создал его в 1991 году. Python представляет собой высокоуровневый язык программирования. Он очень прост для изучения благодаря своему минимализму, понятному синтаксису и хорошей читабельностью.
4. Ruby очень похож на Python – это простые в понимании и читабельные языки программирования, которые используются в основном для разработки веб-приложений. Ruby разработан в 1995 году Юкихио Мацумото. Обычно чистый Ruby не используется, а используется его фреймворк RubyOnRails. Это мощный фреймворк веб-разработки. На нем написаны многие сайты, включая Yammer, Shopify, Groupon, Github, Scribd.
5. Delphi вобрал в себя лучшие достоинства Pascal, соединив ее с гибкостью объектно-ориентированного программирования.

Язык имеет высокую читабельность кода, простой и понятный синтаксис, в то же время достаточно жесткий.

Delphi более подвержен сбоям по сравнению, например, с C#.

Язык не такой гибкий, как C++ или C.

1. Borland Delphi – среда визуального программирования, предоставляющий разработчику большой объем возможностей по созданию программ, предназначенных для работы с базами данных. Для работы с базами данных в него включены такие возможности как генератор отчетов, позволяющий осуществлять вывод информации на принтер, а также организована поддержка языка SQL, позволяющего создавать запросы различной сложности с локальными и сетевыми базами данных.

Delphi предоставляет разработчику приложения широкие возможности быстрого и качественного проектирования графического интерфейса пользователя – различных окон, кнопок, меню и т.д.

1. C++Builder – это продукт фирмы Borland, предназначенный для быстрой разработки приложений (RAD – rapid application development) на языке Си++. С помощью C++Builder можно создавать как консольные приложения Win32, так и использовать графический интерфейс пользователя (GUI – graphical user interface). При создании GUI-приложений Win32 с помощью C++Builder доступна вся мощь языка Си++, заключенная в среду RAD. Это означает, что можно создавать интерфейс пользователя (меню, диалоговые окна, главное окно и т.д.), используя технику drag-and-drop. Программист может также помещать в формы элементы управления ОСХ для создания в считанные минуты специализированных приложений, таких как Web-браузеры. При этом не жертвуя скоростью выполнения программ, потому что вся мощь языка Си++ по-прежнему остается в его распоряжении. Стоит отметить, что язык Си++ не является простым даже для профессионалов. Поэтому реализацию программных продуктов на этом языке должны осуществлять только специалисты, имеющие большой опыт работы с данным языком.
2. MS Visual Studio – это решение для разработки, позволяющее командам любого размера проектировать и создавать привлекательные приложения, которые удовлетворят самым взыскательным требованиям заказчиков.

С помощью MS Visual Studio можно использовать инструменты гибкого планирования – такие как планирование объема работ, панели задач и управление невыполненной работой. Их использование позволяет осуществить последовательную разработку и организовать создание программного продукта в удобном темпе.

MS Visual Studio включает средства моделирования, обнаружения и проектирования архитектуры, что позволяет описать создаваемую систему и полностью

реализовать концепцию программного продукта.

Стоит отметить, что в MS Visual Studio присутствуют механизмы, позволяющие получать подробную информацию о возникающих ошибках.

Осуществим сравнение описанных выше языков программирования. Сравнение языков программирования проводилось самостоятельно на базе заявленных производителем технических возможностей языков программирования с использованием метода сравнения.

В качестве языка программирования выбран Delphi, соответственно, средой программирования будет являться Delphi версии 2010 года, т.к. Delphi является императивным, структурированным, объектно-ориентированным языком программирования со строгой статической типизацией используемых переменных.

Delphi 2010 представляет собой среду, предназначенную для визуального проектирования приложений для Windows с использованием различных механизмов повторного использования разработанного программного кода. У языка программирования Delphi, как и у многих других крупных компаний, есть конкуренты. Им является всем знакомая среда для разработки Microsoft Visual C++, которая имеет свои особенные преимущества и некоторые недостатки, но является более популярной, в основном, из-за того, что была разработана корпорацией Microsoft.

Отличительной чертой Delphi является модель компонентной разработки прикладного программного обеспечения. В Delphi Компоненты очень просты для развития и использования. Как результат покрытия большей части той структуры программы, которая является близкой к взаимодействию с операционной системой.

Delphi является одной из первых систем, которая занялась быстрой разработкой приложений (RAD) и технологию визуального конструирования. Технология Visual Design содержит готовые компоненты, из которых будет строиться интерфейс будущей программы.

Меню представляет из себя быстрый и гибкий интерфейс для среды разработки Delphi, которым можно управлять по средствам набора горячих клавиш. Также, можно использовать интерактивное меню для выполнения множества задач; наиболее общих задач вроде управления отладчиком, открытия и закрытия файлов или настройкой среды программирования.

Среда разработки Delphi является средой программирования, в которой сочетаются простота и удобство с мощностью и гибкостью объектно-ориентированного программирования. Она обеспечивает визуальное проектирование пользовательского интерфейса и уникальные по своей простоте и мощности средства доступа к базам данных.

В среде разработки Delphi нет никаких сложных конструкций и при этом она не уступает C++ по своим возможностям. Скорость работы созданных программ на Delphi не меньше, чем у программ, которые создавались на C++. Разница будет заметна при большом объеме математических вычислений, но небольшая. Это объясняется отличной способностью оптимизации программ компилятора C++.

В тоже время, чем лучше оптимизация программ, тем больше это занимает времени, что отражается на процессе разработки программ. Здесь, несомненно, преимущество компилятора у Delphi, который позволяет почти мгновенно внести изменения и проверить результат. Компоненты среды программирования Delphi являются, по существу, просто специализированными объектами, и их функционирование определяется объектно-ориентированной Visual Component Library (архитектурой библиотеки визуальных компонентов). Сокращенно – VCL.

Delphi предлагает пользователю довольно мощный набор компонентов для работы с базами данных, к тому же она очень проста и логична.

У среды разработки Delphi есть еще очень важное преимущество в отличии от остальных коммерчески успешных языков программирования – он отлично подходит для обучения программированию. В связи с этим целесообразно использовать его в качестве языка для разработки.

1.

## **Пример построения интерфейса программы**

1.

### **Этапы построения интерфейса**

В первую очередь необходимо разработать базу данных, а далее пользовательский интерфейс для работы с этой базой данных.

В процессе функционирования предприятия, в эксплуатацию запускается оборудование определенного типа. Данный процесс сопровождается предоставлением документации, на основании которой данное оборудование учитывается в различных отделах.

Для формирования отчетов в данной организации существуют следующие формы отчетности:

1. Учет по производителю, показанный в таблице 3.1.
2. Учет по типу оборудования (таблица 3.2).
3. Учет оборудования (таблица 3.3).

Таблица 3.1

Учет по производителю

№ п/п	Код оборудования	Наименование оборудования	Производитель	Серийный номер
----------	---------------------	------------------------------	---------------	-------------------

Таблица 3.2

Учет по типу оборудования

№ п/п	Код оборудования	Наименование оборудования	Производитель	Серийный номер
----------	---------------------	------------------------------	---------------	-------------------

Таблица 3.3

Учет оборудования

№ п/п	Подразделение	Индекс АТС	Наименование оборудования	Производитель	Серийный номер
----------	---------------	---------------	------------------------------	---------------	-------------------

Во время эксплуатации оборудование может выйти из строя. В таких случаях его оформляют в ремонт и делают запись в журнал учета ремонта оборудования. При возврате отремонтированного оборудования в данном журнале делается отметка о получении, и оно сдается в дальнейшую эксплуатацию.

Все вышеописанное показано в схеме информационных потоков на рисунке 3.1.



Рис.3.1. Схема информационных потоков

Таким образом, разрабатываемая база данных должна решать следующие задачи:

- Учет используемого оборудования;
- Учет вновь устанавливаемого оборудования;
- Учет повреждения и отправки оборудования в ремонт;
- Поиск оборудования одного производителя, для составления договоров гарантийного обслуживания;
- Поиск оборудования одного типа, для анализа использования, возможной замены на более современное;
- Поиск типичных неисправностей, для разработки вариантов снижения сбоев и повреждений;
- Генерация отчета по неисправному оборудованию.
- Формирование сопроводительного документа (акта) при оформлении ремонта оборудования.

ER-модель (Модель сущность-связь) – модель данных, позволяющая описать концептуальные схемы предметной области.

Концептуальная модель базы данных – это отражение предметной области, для которой разрабатывается база данных. Не вдаваясь в теорию, отметим, что это некая диаграмма с принятыми обозначениями элементов. Так, все объекты, обозначающие вещи, обозначаются в виде прямоугольника. Атрибуты, характеризующие объект - в виде овала, а связи между объектами - ромбами. Мощность связи обозначаются стрелками (в направлении, где мощность равна многим – двойная стрелка, а со стороны, где она равна единице - одинарная).[6]

Таким образом, можно выделить следующие объекты необходимые для учета оборудования связи (рисунок 3.2):

- Подразделение;
- Оборудование;
- Производитель;
- Журнал ремонта.

Для формирования сопроводительного документа введем следующие объекты:

- Сотрудник;
- Должность.

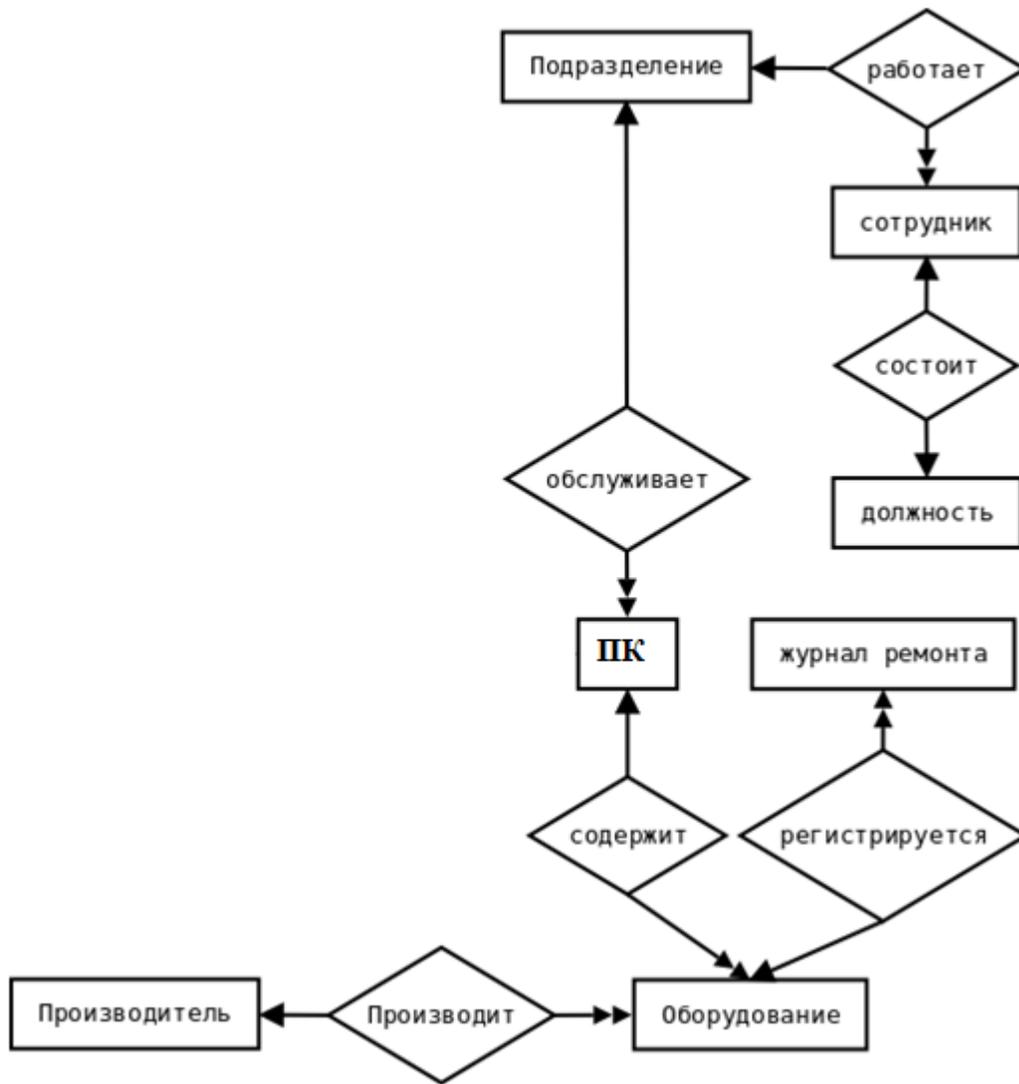


Рис.3.2. Концептуальная модель

При дальнейшем рассмотрении предметной области можно выделить атрибуты указанных выше объектов, показанные в таблице 3.3.

Таблица 3.3

Объекты и атрибуты ER-модели

Объект	Атрибут
Подразделение	Наименование

Код

Наименование

Оборудование

Серийный номер

Тип

Производитель

Производитель

Наименование

Место производства

Неисправность

Дата неисправности

Дата отправки

Способ отправки

Журнал ремонта

Кто отправил

Дата возврата

Способ возврата

Кто принял

Для того чтобы построить реляционную модель, необходимо сформировать набор таблиц.

Таблица 3.4

Виды ТО

Наименование поля Тип данных

Код	Счетчик
Наименование	Текстовый
Тип_работы	Текстовый

Таблица 3.5

Компьютерная техника

Наименование поля Тип данных

Код	Счетчик
Сетевое имя	Текстовый
Месторасположение	Текстовый
Примечание	Текстовый

Таблица 3.6

Оргтехника

Наименование поля Тип данных

Код Счетчик

Название Текстовый

Тип Текстовый

Месторасположение Текстовый

Таблица 3.7

ПО

Наименование поля Тип данных

Код Счетчик

Название Текстовый

Дата покупки Дата/время

Дата окончания Дата/время

Количество лицензий Числовой

Таблица 3.8

Сотрудники

Наименование поля Тип данных

Код	Счетчик
ФИО	Текстовый
Должность	Текстовый

Таблица 3.9

ТО компьютерной техники

Наименование поля Тип данных

Код	Счетчик
Код сотрудника	Числовой
Код_техники	Числовой
Код работы	Числовой
Дата	Дата/время

Таблица 3.10

ТО оргтехники

Наименование поля Тип данных

Код	Счетчик
Код сотрудника	Числовой

Код_оргтехники	Числовой
Код работы	Числовой
Дата	Дата/время

Таблица 3.11

Установленное ПО

Наименование поля Тип данных

Код	Счетчик
Код техники	Числовой
Код ПО	Числовой
Дата установки	Дата/время
Дата удаления	Дата/время

Схема базы данных представлена на рисунке 3.2.

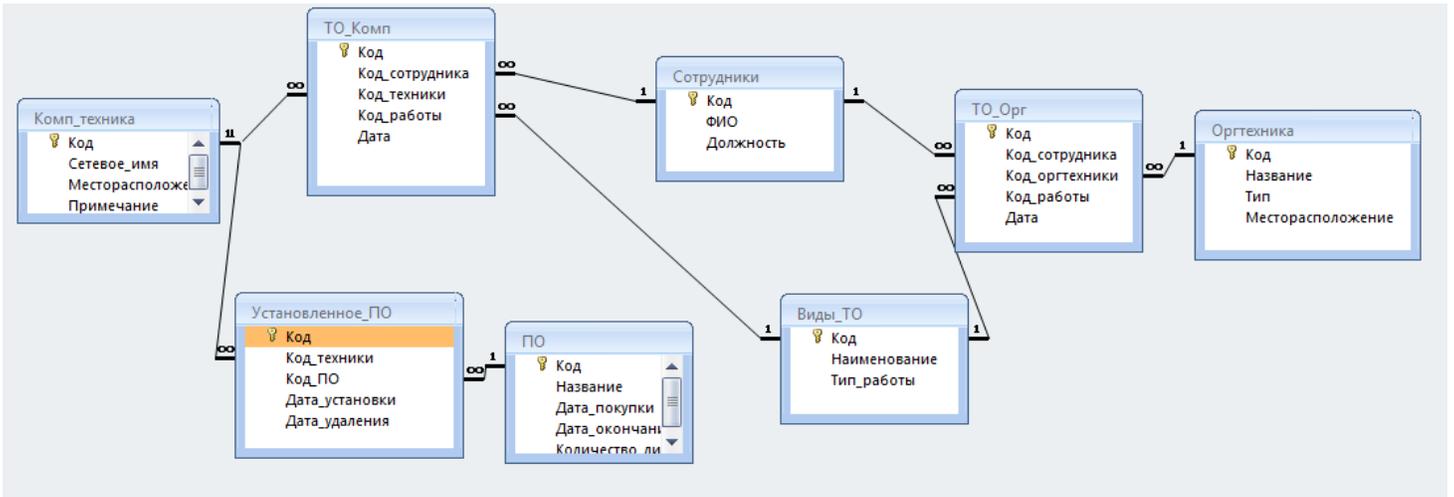


Рис.3.2. Схема базы данных

- 1. Результаты разработки интерфейса

Главное окно программы представлено на рисунке 3.3.

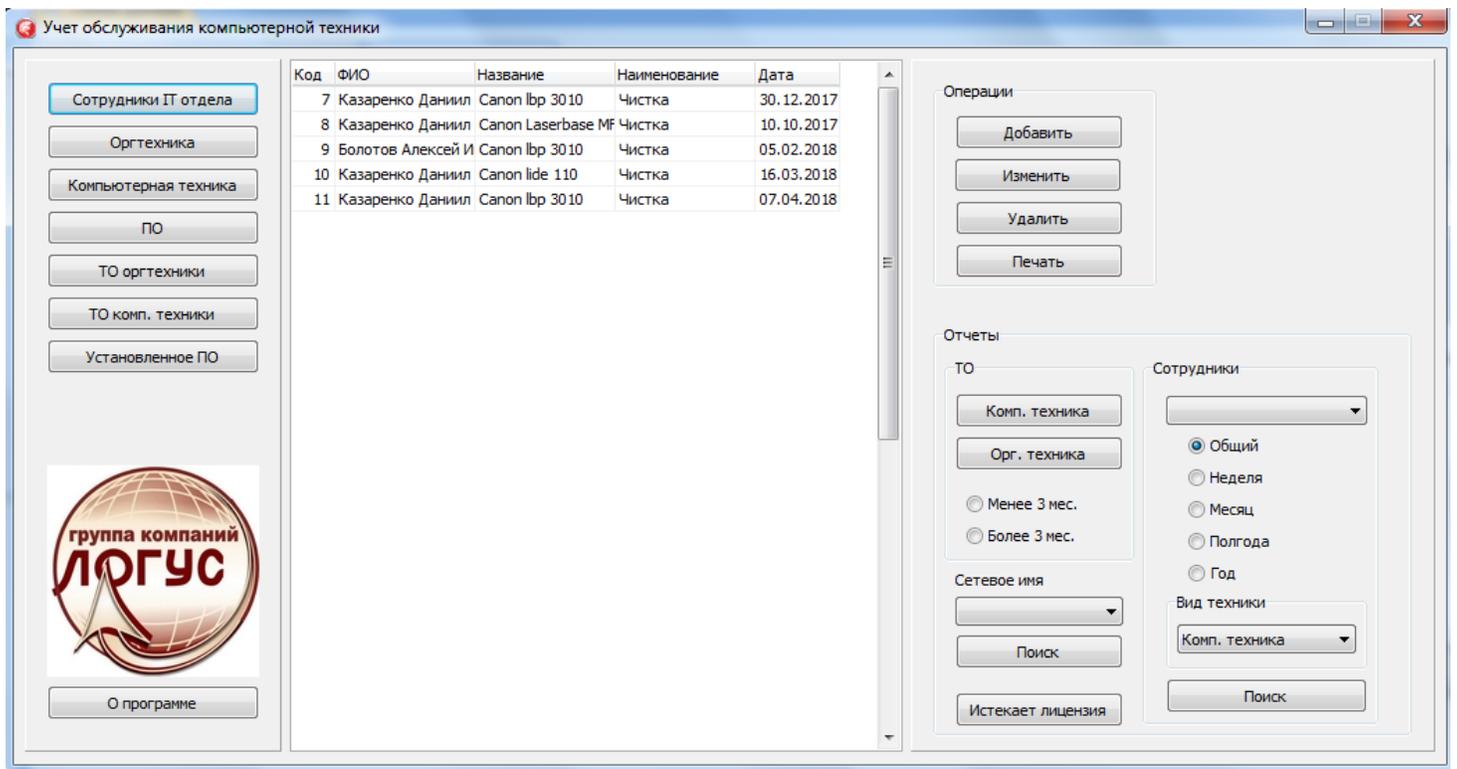


Рис.3.3. Главное окно программы

Рассмотрим каждый пункт меню. На рисунке 3.4 приведен перечень имеющейся оргтехники.

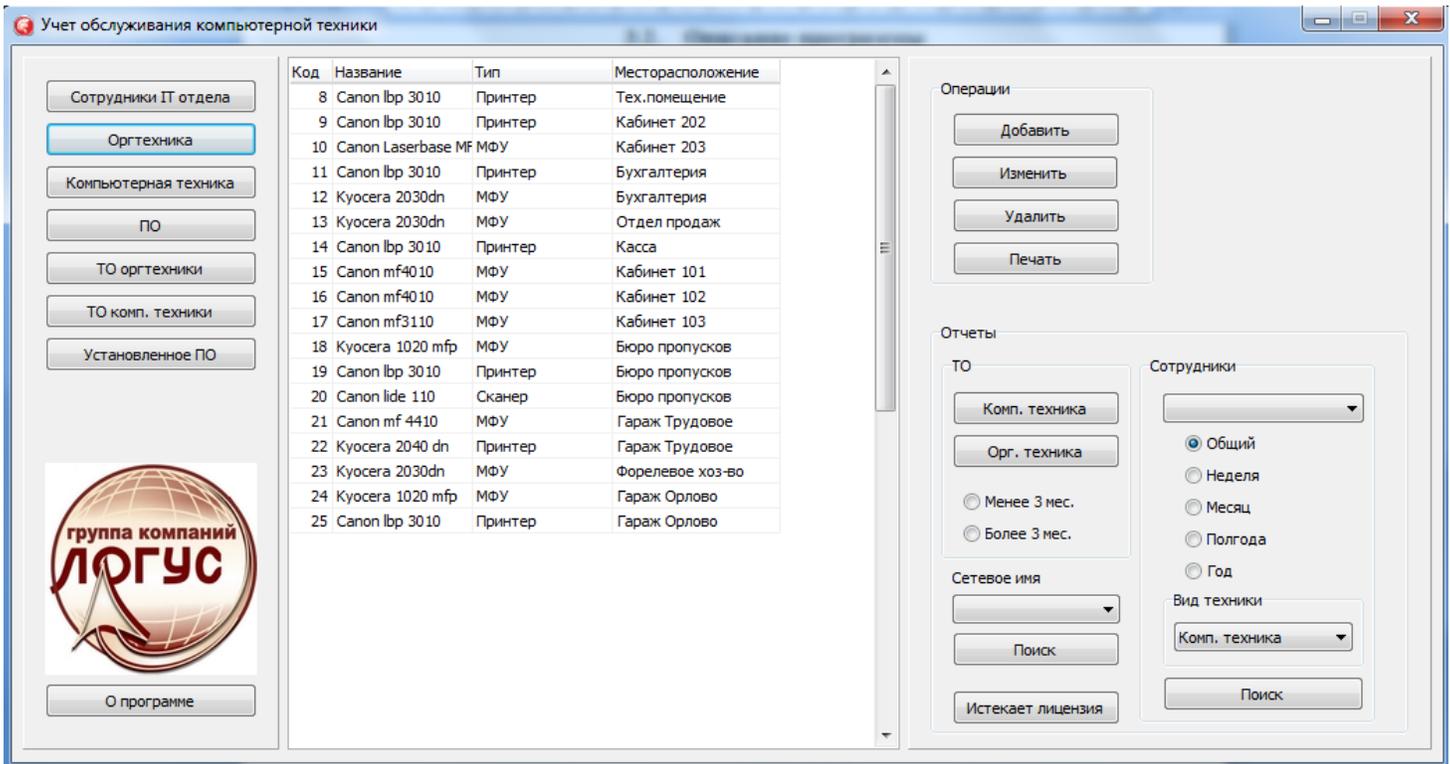


Рис.3.4. Перечень оргтехники

На рисунке 3.5 приведен перечень компьютерной техники.

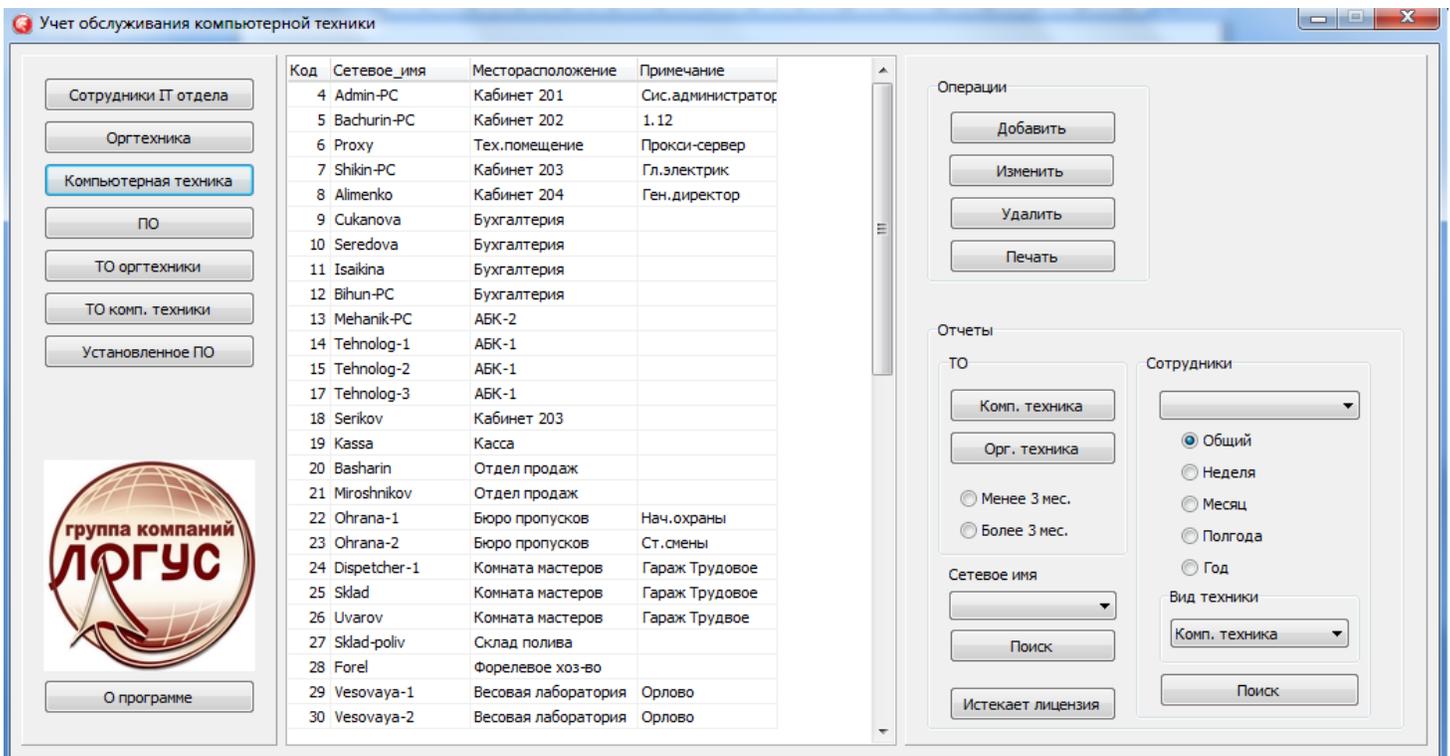
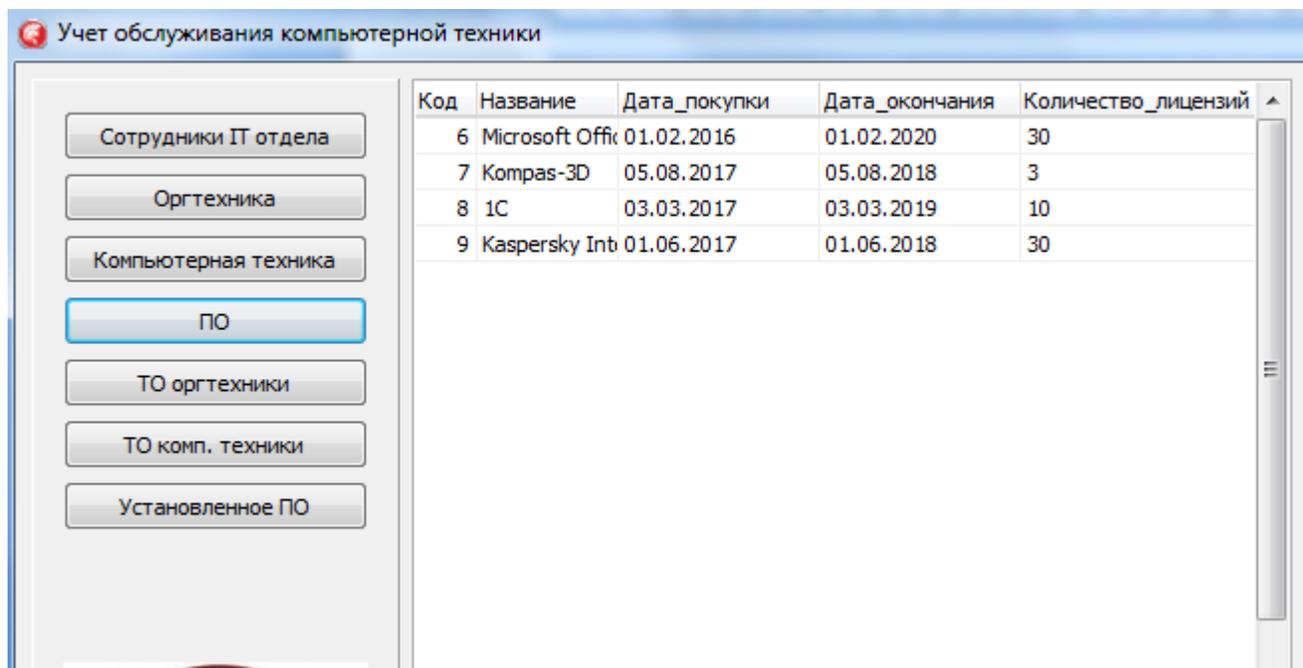


Рис.3.5. Перечень компьютерной техники

Список программного обеспечения приведен на рисунке 3.6.



Код	Название	Дата_покупки	Дата_окончания	Количество_лицензий
6	Microsoft Office	01.02.2016	01.02.2020	30
7	Kompas-3D	05.08.2017	05.08.2018	3
8	1С	03.03.2017	03.03.2019	10
9	Kaspersky Int	01.06.2017	01.06.2018	30

Рис.3.6. Список программного обеспечения

Учет обслуживания компьютерной техники

Сотрудники IT отдела

Оргтехника

Компьютерная техника

ПО

ТО оргтехники

**ТО комп. техники**

Установленное ПО



О программе

Код	ФИО	Сетевое_имя	Наименование	Дата
5	Казаренко Даниил	Admin-PC	Чистка	10.12.2017
7	Казаренко Даниил	Shikin-PC	Чистка	10.12.2017
8	Злодеев Максим Кс	Alimenko	Чистка	20.01.2018
9	Казаренко Даниил	Bachurin-PC	Починка	10.02.2018
10	Казаренко Даниил	Cukanova	Чистка	05.03.2018
11	Казаренко Даниил	Seredova	Чистка	05.03.2018
12	Казаренко Даниил	Isaikina	Чистка	05.03.2018
13	Казаренко Даниил	Bihun-PC	Чистка	05.03.2018
14	Злодеев Максим Кс	Mehanik-PC	Чистка	20.02.2018
15	Злодеев Максим Кс	Tehnolog-1	Чистка	20.02.2018
16	Злодеев Максим Кс	Tehnolog-2	Чистка	20.02.2018
17	Злодеев Максим Кс	Tehnolog-3	Чистка	20.02.2018
18	Болотов Алексей И	Ohrana-1	Чистка	04.11.2017
19	Болотов Алексей И	Ohrana-2	Чистка	04.11.2017
20	Казаренко Даниил	Miroshnikov	Починка	20.12.2017
21	Злодеев Максим Кс	Serikov	Чистка	07.02.2018
22	Злодеев Максим Кс	Kassa	Чистка	07.02.2018
23	Казаренко Даниил	Shikin-PC	Починка	15.03.2018
24	Казаренко Даниил	Forel	Чистка	14.03.2018
25	Казаренко Даниил	Dispetcher-1	Починка	15.04.2018
26	Злодеев Максим Кс	Vesovaya-1	Чистка	01.04.2018
27	Злодеев Максим Кс	Elevator	Чистка	02.04.2018
28	Злодеев Максим Кс	GSM	Чистка	02.04.2018
29	Злодеев Максим Кс	Himsklad	Чистка	02.04.2018
30	Казаренко Даниил	Uvarov	Чистка	16.02.2018
31	Казаренко Даниил	Uvarov	Починка	16.02.2018

Рис.3.7. ТО компьютерной техники

В каждом окне имеется список операций, которые можно совершить (добавление, изменение, удаление, печать), список отчетов, возможность фильтра, поиска.

Таким образом, приведенные окна программы показывают, что она имеет удобный пользовательский интерфейс.

## Заключение

В работе была спроектирован интерфейс для системы учета технического обслуживания компьютерной техники в ООО УК «Логус-Групп».

На этапе обследования была рассмотрена общая характеристика объекта автоматизации, организация работы. На основе анализа сформированы и обоснованы требования к работе системы.

На стадии проектирования разработана общая структура информационной системы в целом, а также по каждой отдельной ее задаче. Определены основные проектные решения, что стало основанием для разработки, отладки программной части и для конструирования эксплуатационной документации.

В качестве средства для разработки программного проекта был выбран язык программирования Delphi.

Система учета компьютерной техники ООО УК «Логус-Групп» позволит повысить производительность труда менеджера, увеличит объем продаваемой продукции на 25%, увеличит прибыль фирмы на 10%, сократить время работы менеджера с документами, и поиск заявок в среднем на 25% за счет автоматического анализа информации, имеющейся в базе данных.

Использование информационной системы позволит более глубоко и в полном объеме собирать и анализировать необходимую информацию о заявках на техническое обслуживание и ремонт.

Отмечено также повышение эффективности учета движения заявок. Для быстрой и полной адаптации пользователя к системе был разработан удобный дружественный интерфейс пользователя и подробное описание работы с системой в контрольном примере.

Считаю, что созданная в работе система учета заявок в отделе технической поддержки ООО «Логус-Групп» полностью соответствует информационным требованиям предприятия и сможет поддерживать это соответствие в течение всего жизненного цикла системы.

## **Список использованной литературы**

1. Аткинсон Л. MySQL. Библиотека профессионала - СПб: Вильямс, 2014. - 624 с.
2. Бритов Г., Осипова Т. Моделирование бизнес-процессов. - М.: LAP, 2014. - 124 с.
3. Варфоломеева Е.В. Информационные системы в экономике: Учебное пособие / Е.В. Варфоломеева, Т.В. Воропаева и др.; Под ред. Д.В. Чистова - М.: НИЦ ИНФРА-М, 2015. - 234 с.

4. Вдовенко Л.А. Информационная система предприятия: Учебное пособие/Вдовенко Л. А. - 2 изд., перераб. и доп. - М.: Вузовский учебник, НИЦ ИНФРА-М, 2015. - 304 с.
5. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. - СПб.:Питер, 2015. - 368 с.
6. Гвоздева В.А. Базовые и прикладные информационные технологии: Учебник / Гвоздева В. А. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2015. - 384 с.
7. Гвоздева В.А. Информатика, автоматизированные информационные технологии и системы: Учебник / В.А. Гвоздева. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2015. - 544 с.
8. Дейт К.Дж. Введение в системы баз данных. - К.: Диалектика, 2012. - 360 с.
9. Дунаев В.В. Базы данных. Язык SQL для студента - Издательство: БХВ, 2013. - 196 с.
10. Душин, В.К. Теоретические основы информационных процессов и систем : учебник / В.К. Душин .— 5-е изд. — М. : ИТК "Дашков и К", 2014 .— 348с.
11. Заботина Н.Н. Проектирование информационных систем: Учебное пособие / Н.Н. Заботина. - М.: ИНФРА-М, 2011. - 331 с.
12. Затонский А.В. Информационные технологии: разработка информационных моделей и систем: Учеб. пос. / А.В.Затонский - М.: ИЦ РИОР: НИЦ ИНФРА-М, 2014 - 344с.
13. Илющечкин В. Основы использования и проектирования баз данных. Учебник. - М.:Юрайт, 2014. - 214с.
14. Информационные системы в экономике: Учебник / К.В. Балдин, В.Б. Уткин. - 7-е изд. - М.: Дашков и К, 2012. - 395 с.
15. Исаев Г. Проектирование информационных систем. Учебное пособие. - М.: Омега-Л, 2015. - 432с.
16. Карминский А.М. Методология создания информационных систем: Учебное пособие / А.М. Карминский, Б.В. Черников. - 2-е изд., перераб. и доп. - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 320 с.
17. Коваленко В.В. Проектирование информационных систем: Учебное пособие / В.В. Коваленко. - М.: Форум: НИЦ ИНФРА-М, 2014. - 320 с.
18. Кузин, А.В. Базы данных: Учебное пособие для студ. высш. учеб. заведений / А.В. Кузин, С.В. Левонисова. - М.: ИЦ Академия, 2014. - 320 с.
19. Марков А.С., Лисовский К.Ю. Базы данных. Введение в теорию и методологию: Учебник. - М.: Финансы и статистика, 2015. - 512 с.
20. Наумов А.Н., Вендров А.М., Иванов В.К. и др. Системы управления базами данных и знаний - М.: Финансы и статистика, 2014. - 352 с.

21. Пирогов, В.Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В.Ю. Пирогов. - СПб.: БХВ-Петербург, 2014. - 528 с.
22. Редько В.Н., Бассараб И.А. Базы данных и информационные системы. - М.: Знание, 2015. - 602 с.
23. Советов, Б.Я. Базы данных: теория и практика: Учебник для бакалавров / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. - М.: Юрайт, 2013. - 463 с.
24. Тоу Дэн Настройка SQL - СПб: Питер, 2013. - 539 с.
25. Уткин В., Балдин К. Информационные системы в экономике. - М.: Academia, 2014. - 288с.
26. Федотова Е. Информационные технологии и системы: Учебное пособие / Е.Л. Федотова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. - 352 с
27. Федотова Е.Л. Информационные технологии в профессиональной деятельности: Учебное пособие / Е.Л. Федотова. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2015. - 368 с.
28. Фуфаев, Э.В. Базы данных: Учебное пособие для студентов учреждений среднего профессионального образования / Э.В. Фуфаев, Д.Э. Фуфаев. - М.: ИЦ Академия, 2014. - 320 с.
29. Хаббард Дж. Автоматизированное проектирование баз данных - М.: Мир, 2014. - 453 с.
30. Черников Б.В. Информационные технологии управления: Учебник / Б.В. Черников. - 2-е изд., перераб. и доп. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. - 368 с.
31. Шаймарданов Р.Б. Моделирование и автоматизация проектирования структур баз данных - М.: Юнити, 2016. - 469 с.
32. Шишов О.В. Современные технологии и технические средства информатизации: Учебник / О.В. Шишов. - М.: НИЦ Инфра-М, 2012. - 462 с.