

**Колледж космического машиностроения и технологий**

**КУРСОВОЙ ПРОЕКТ**

**По МДК.01.02 «Прикладное программирование»**

**Тема: «Разработка приложения "Компьютерный мастер"»**

Выполнила студентка

Группы П1-17

Фролова Л.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Подпись)

\_\_\_\_\_\_\_\_\_\_\_ (Дата сдачи работы)

Проверил преподаватель

Гусятинер Л.Б.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Оценка)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Подпись)

Королёв 2020 г.

Содержание

[ВВЕДЕНИЕ 3](#_Toc44084218)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc44084219)

[1.1. Анализ предметной области 4](#_Toc44084220)

[1.2. Характеристика аппаратного обеспечения фирмы 7](#_Toc44084221)

[1.5 Характеристика программного обеспечения 9](#_Toc44084222)

[1.6 Обработка заказа фирмой «CompGear» 10](#_Toc44084223)

[2. ПРОЕКТНАЯ ЧАСТЬ 11](#_Toc44084224)

[2.1. Диаграмма прецедентов 11](#_Toc44084225)

[2.2 . Выбор инструментов 12](#_Toc44084226)

[2.3. Проектирование сценария 13](#_Toc44084227)

[2.5. Описание главного модуля 15](#_Toc44084228)

[2.6. Описание спецификаций к модулям 16](#_Toc44084229)

[2.7. Описание модулей программы 17](#_Toc44084230)

[2.8. Описание тестовых наборов модулей 23](#_Toc44084231)

[2.9. Описание применения средств отладки 25](#_Toc44084232)

[2.10. Анализ оптимальности использования памяти и быстродействия 26](#_Toc44084233)

[3. ЭКСПЛУТАЦИОННАЯ ЧАСТЬ 27](#_Toc44084234)

[3.1 . Эксплуатационный раздел 27](#_Toc44084235)

[3.2. Рекомендации по эксплуатации программного продукта 28](#_Toc44084236)

[3.3. Руководство администратора по установке программы 31](#_Toc44084237)

[3.4. Руководство пользователя 32](#_Toc44084238)

[ЗАКЛЮЧЕНИЕ 36](#_Toc44084239)

[Список использованных источников. 37](#_Toc44084240)

[Приложение 1 38](#_Toc44084241)

[Приложение 2 40](#_Toc44084242)

[Приложение 3 43](#_Toc44084243)

[Приложение 4 44](#_Toc44084244)

[Приложение 5 45](#_Toc44084245)

[Приложение 6 46](#_Toc44084246)

# ВВЕДЕНИЕ

Целью курсового проекта является разработка программы для рабочего места компьютерного мастера. Автоматизированная система рабочего места компьютерного мастера по предоставлению услуг в компьютерной сфере должна выполнять следующие функции: сборка и подбор комплектующих, полное обслуживание ПК, диагностика оборудования, восстановление данных.

В теоретической части идет описание о рабочем месте компьютерного мастера, какие задачи он выполняет, какое оборудование для него требуется. Также в теоретической части рассказано о математической постановке задачи – в ней приведены формулы по расчету предоставляемых услуг.

Во второй главе опишем структуру базы данных проекта, а также построим структуру интерфейса пользователей, разработаем алгоритмы основных функций системы. Следующим шагом станет определение требований к автоматизации на основе выделенных бизнес-процессов. Данные требования должны будут предъявляться к функциональным возможностям системы, к документообороту и аналитическим расчетам, к интерфейсу системы.

В третьей части будет рассмотрено руководство для пользователей.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## **Анализ предметной области**

Рабочее место компьютерного мастера – это пространство в специализированном помещении, где установлено оборудование для работы с компьютерной электроникой. Оснащение рабочего места компьютерного мастера таким оборудованием позволяет не только увеличить скорость работы с дорогостоящей техникой, но и позволяет качественнее выявить неисправность. Мастер-приемщик – осуществляет приём клиента, принимает оплату выполненной услуги и выдаёт клиенту бумажный чек.

Далее будет приведено описание фирмы ООО «CompGear».

«CompGear» является коммерческой организацией, созданной в организационно-правовой форме «Общество с Ограниченной Ответственностью» (ООО), в соответствии с действующим законодательством Российской Федерации. Общество является юридическим лицом и действует на основании законодательства РФ и устава.

Основной целью организации является получение прибыли в интересах акционеров Общества. Предметной областью деятельности общества является насыщение потребительского рынка по оказанию услуг в компьютерной сфере. Общество в установленном законом порядке осуществляет следующие виды деятельности:

* ремонт и обслуживание компьютеров и периферийного оборудования компьютеров, такого как настольные компьютеры и ноутбуки, компьютерные терминалы, устройства памяти и принтеры;
* обучение клиентов;
* установка программного обеспечения;
* установка антивирусных программ;
* установка драйверов;



Рисунок 1. Рабочее место компьютерного мастера

Оборудование рабочего места компьютерного мастера может включать:

* компьютер;
* паяльное оборудование;
* измерительное оборудование (мультиметр, осциллограф);
* ручной инструмент (микрокусачки, набор отверток, набор инструментов для пайки);
* регулируемый блок питания;
* бинокулярный микроскоп;
* ПО (дистрибутив Windows, антивирус, драйвера)

Выбор оснащения зависит от специфики предоставляемой услуги и выделенного пространства.

Организация рабочего места компьютерного мастера требует стартовых вложений, однако именно от технической оснащенности зависит быстрота и качество предоставляемых услуг, а также репутация предприятия среди клиентов.

Структура организации соответствует ее отраслевой специфике. При этом для нее характерна потребность в квалифицированном персонале, обладающем специальными навыками. Организационная структура фирмы ООО «CompGear» представляет собой структуру, которая показана на рис.1.1.



Рисунок 2. Структура фирмы ООО «CompGear»

Процесс выполнения заказа начинается с поступления заказа от клиента. В качестве клиента могут выступать как физические, так и юридические лица. Клиент оставляет заявку, которую вносит в таблицу мастер-приемщик. Затем заявка передается сборщику-отладчику, который производит осмотр техники и выявляет сроки выполнения работы. После завершения выполнения работы, мастер-оценщик производит оценку оборудования, выдаваемого клиенту.

## Характеристика аппаратного обеспечения фирмы

В данном пункте будет приведена характеристика аппаратного обеспечения фирмы ООО «CompGear».

Пользователи используют стационарные ПК, характеристики которых удовлетворяют необходимым требованиям выполнения рабочих обязанностей их пользователей. Характеристики компьютеров сотрудников фирмы ООО «CompGear» показаны в таблице 1.1:

Таблица 1.1.

Характеристики ПК пользователей ЛВС

|  |  |  |  |
| --- | --- | --- | --- |
| **Должность пользователя** | **Наименование системного блока ПК** | **Характеристики** | |
| Мастер-приемщик | Компьютер HP Desktop | Процессор | Intel-core i3-6100 |
| Оперативная память | 4GB DDR4 RAM |
| Видеокарта | - |
| Жесткий диск | 120GB SSD |
| Блок питания | 450WT |
| Монитор | 15.6 ASUS VT168N |
| Мышь | Genius DX-125 Black |
| Клавиатура | Defender HB-420 Black |
| Сборщик-отладчик  Компьютер HP Desktop Pro | | Процессор | Intel-core i5 6600k |
| Оперативная память | 16GB DDR4 RAM |
| Видеокарта | NVidia GeForce RTX 2060 Super |
| Жесткий диск | 480GB SSD |
| Блок питания | 750WT |
| Монитор | 15.6 ASUS VT168N |
| Мышь | Genius DX-125 Black |
| Клавиатура | Defender HB-420 Black |
| Мастер-оценщик | Компьютер HP Desktop Extreme | Процессор | Intel-core i5-6400 |
| Оперативная память | 8GB DDR4 RAM |
| Видеокарта | NVidia GeForce GTX 1050TI |
| Жесткий диск | 480GB SSD |
| Блок питания | 600WT |

Продолжение таблицы 1.1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Должность пользователя** | **Наименование системного блока ПК** | **Характеристики** | |
| Мастер-оценщик | Компьютер HP Desktop Extreme | Монитор | 15.6 ASUS VT168N |
| Мышь | Genius DX-125 Black |
| Клавиатура | Defender HB-420 Black |

Также на предприятии имеется следующее оборудование:

* Маршрутизатор Keenetic Giga KN-1010;
* Принтер - лазерный HP Laser 107w.

Схема подключения оборудования показана на рис 3.

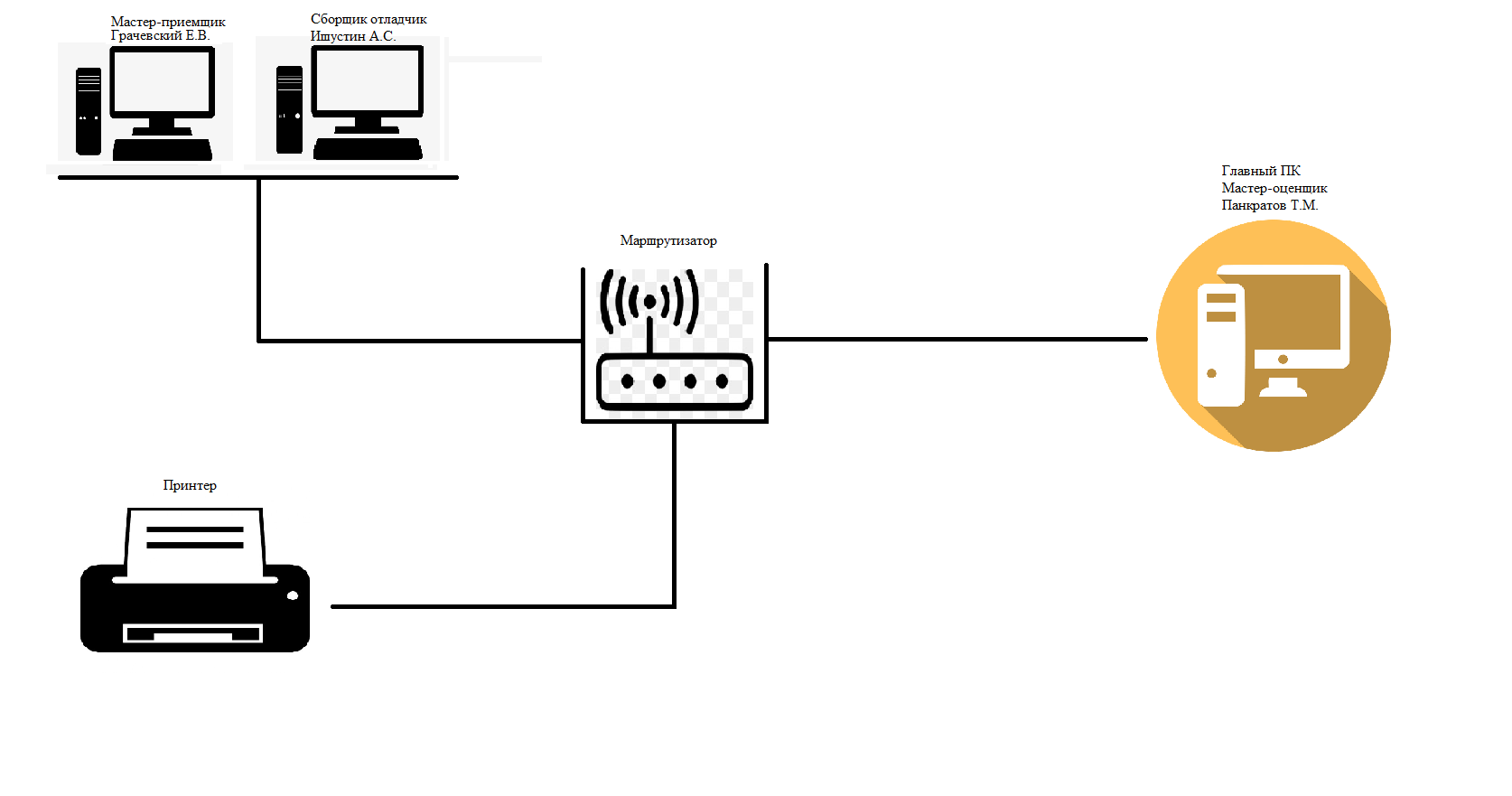


Рисунок 3. Схема аппаратного обеспечения

## . Характеристика программного обеспечения

На компьютеры фирмы «CompGear» установлено программное обеспечение, которое необходимо для работы организации. Программное обеспечение показано в таблице 1.2.

Таблица 1.2.

Имеющееся программное обеспечение на компьютерах фирмы «CompGear».

|  |  |  |
| --- | --- | --- |
| **№** | **Наименование** | **Статус ПО** |
| 1 | MS Windows 10 Pro | платно |
| 2 | MS Office 2016 Pro Plus | платно |
| 3 | 7-zip | бесплатно |
| 4 | Virtual Box | бесплатно |
| 5 | TeamViewer | бесплатно |
| 6 | Visual Studio 2019 | бесплатно |
| 7 | DriverPack Solution | бесплатно |
| 8 | Victoria | бесплатно |
| 9 | CristalDisk | бесплатно |
| 10 | Aida64 | бесплатно |
| 11 | UltraISO | бесплатно |
| 12 | AdobeReader | бесплатно |
| 13 | FPSmonitor | платно |
| 14 | Google Chrome | бесплатно |
| 15 | Антивирус Касперского | платно |
| 16 | Acronis Disk Director | платно |

## . Обработка заказа фирмой «CompGear»

Процесс выполнения заказа начинается с поступления заявки от клиента. В качестве клиента могут выступать как физические, так и юридические лица. Клиент может заказать сборку ПК, установку и настройку персонального компьютера, ремонт и диагностику своего персонального устройства. Затем этот заказ обрабатывается мастером-приемщиком, в случае заказа сборки ПК, заказ перенаправляется мастеру-сборщику, который произведёт оценку объёма данной работы с учетом пожеланий клиента.

В случае заказа сборки нового ПК следует достигнуть договорённости с поставщиками комплектующих. Доставленные комплектующие подвергается тщательному осмотру на наличие дефектов. После чего комплектующие собирают в корпус системного блока. Затем мастер-приемщик договаривается с курьерами организации для доставки изделия и оповещает клиента о выполнении его заказа.

В случае заказа на ремонт или диагностику, курьер отправляется по адресу, который указал клиент, забирает указанную им периферию и доставляет её в мастерскую. Затем оборудование разбирается, её составные части прочищаются, а неисправные детали заменяются исправными. После окончания восстановительных работ, курьер организации забирает периферию и возвращает её клиенту.

# ПРОЕКТНАЯ ЧАСТЬ

## 2.1. Диаграмма прецедентов

В ходе выполнения курсовой работы была построена диаграмма прецедентов для сотрудников предприятия. Данная диаграмма описывает работу сотрудников предприятия.

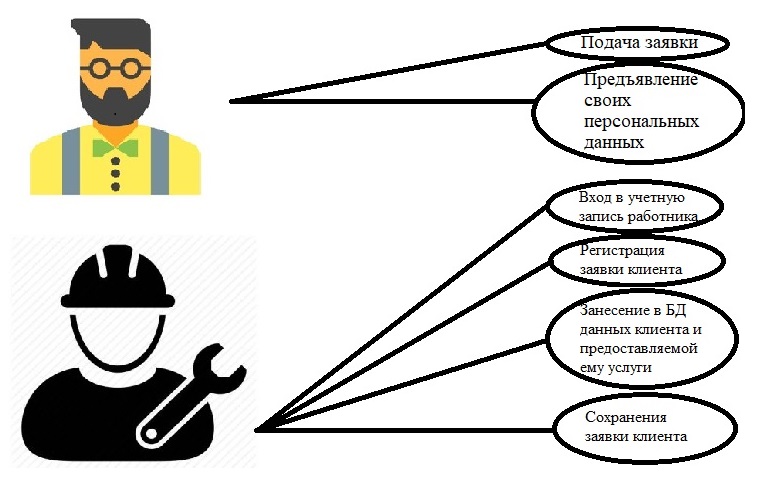


Рисунок 4. Диаграмма прецедентов

## . Выбор инструментов

В рамках курсового проекта было принято решение разработать автоматизированную программу рабочего места компьютерного мастера, включающую базу данных по работе с клиентами. Учитывая возможности имеющегося оборудования и программного обеспечения, необходимо создать современный программный продукт, избегая таких недостатков существующих коммерческих предложений, как высокая стоимость внедрения и сопровождения и слабая ориентированность на пользователя с разной профессиональной подготовкой. Также необходимо уделить особое внимание надежности приложения и простоте его интерфейса.

Перед тем как приступить к написанию программу, следует определиться с инструментарием, на котором она будет разработана, а также его функционал, который мог бы удовлетворить требования программы.

Требования к продукту:

* Поддержка работы с базой данных;
* Высокий уровень производительности.

Для разработки автоматизированной системы были выбраны следующие среды разработки:

1. Lazarus - это специальная программа с интегрированной (свободной) средой для разработки на основе компилятора Free Pascal. Lazarus — программа с полностью открытым исходным кодом, которая направлена на обеспечение библиотек класса Free Pascal (FP). FP — компилятор интегрированной среды разработки соответствует образцу программе Borland — Delphi — и библиотеки Lazarus, которые эквивалентны VCL.
2. Pascal ABC - язык программирования Паскаль нового поколения, включающий классический Паскаль, большинство возможностей языка Delphi, а также ряд собственных расширений. Он реализован на платформе Microsoft.NET и содержит все современные языковые средства, такие как интерфейсы.

## 2.3. Проектирование сценария

Программа рабочего места компьютерного мастера позволяет всем сотрудникам предприятия добавлять и удалять клиентов из таблицы (регистрируя их заявки), редактировать детали заказов:

Для данной программы характерен следующий вид подачи заявки клиента:

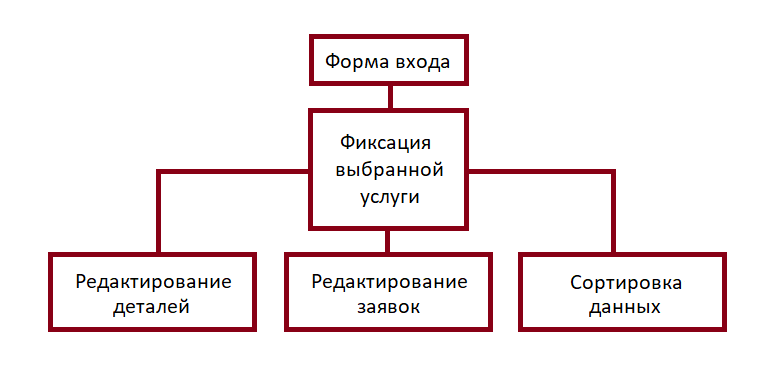
1. Запустить программу;
2. Войти в учётную запись работника;
3. Внесение заявки;
   1. Открыть таблицу с заявками;
   2. Ввести данные нового клиента;
   3. Нажать на кнопку «Сохранить»;
4. Закрыть программу. 

Рисунок 5. Сценарий использования

2.4. **Диаграмма классов**

В приведенной ниже диаграмме классов изображены атрибуты и операции классов. Данная диаграмма является временной и в дальнейшем ходе разработки приложения будет дорабатываться.

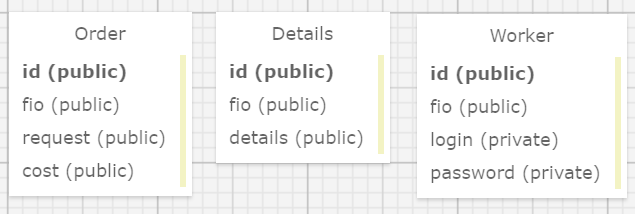


Рисунок 6. Диаграмма классов

Класс «Order» имеет публичные поля id (id заказа), fio (Ф.И.О клиента), request (заявка клиента), cost (стоимость заказа)

Класс «Details» имеет публичные поля id (id заказа), fio (Ф.И.О клиента), details (детали заказа, более подробное описание заявки).

Класс «Worker» имеет публичные поля id (id работника), fio (Ф.И.О работника) и приватные поля login (логин работника для авторизации), password (пароль работника для авторизации).

## 2.5. Описание главного модуля

Главный программный модуль имеет следующую структуру:

1. Форма входа – отвечает за авторизацию пользователя в системе;
2. Окно просмотра заявок;
3. Окно редактирования заявок;
4. Окно просмотра и редактирования деталей заявок;
5. Окно справки о программе.

## 2.6. Описание спецификаций к модулям

В данной главе будут описаны публичные члены модулей курсового проекта.

В главном модуле публичным объектом является DM который представляет из себя главную форму модуля, на которой закреплены объекты, предназначенные для взаимосвязи с базой данных.

В этом объекте так же присутствует процедура DMCreate, которая подсоединяет базу данных к программе.

В модуле отвечающим за заполнение выпадающих списков, и сортировку соответствующих таблиц так же присутствуют публичные методы, которые будут описаны в следующем разделе.

## 2.7. **Описание модулей программы**

Данная программа имеет 5 основных графических окон и 1 модуль данных. Каждое из них имеет свой функционал, который характерен для каждого из них. Далее будут рассмотрены формы программы и их методы и функции

Форма входа:

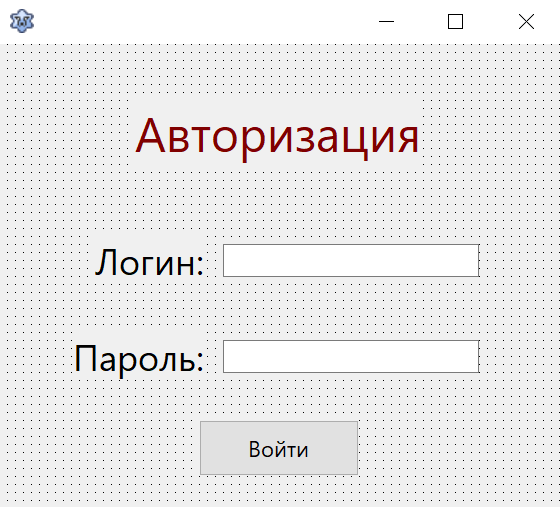


Рисунок 7. Форма входа (FormVhod)

Компоненты:

* ButtonVhod: TButton
* EditPassword: TEdit
* EditLogin: TEdit
* Label1: TLabel
* Label2: TLabel
* Label3: TLabel

Процедуры:

* ButtonVhodClick(Sender: TObject);

Данная форма имеет одну процедуру, вызывающуюся при нажатии на кнопку «Вход». При вызове этой процедуры происходит проверка на корректность введённой связки логина и пароля – если она корректна, то программа разрешит продолжить работу, а в случае, если такие данные в базе данных не найдены, всплывет сообщение об ошибке. Код данной формы представлен в Приложении 1.

Окно просмотра заявок:

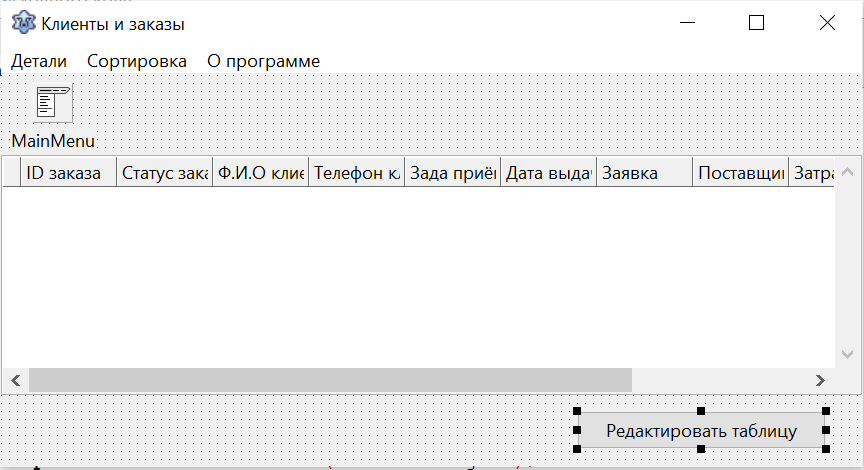


Рисунок 8. Окно просмотра заявок (FormOrdersShow)

Компоненты:

* ButtonEdit: TButton
* DBGrid1: TDBGrid
* MainMenu: TMainMenu
* MenuItem1: TMenuItem
* MenuItem2: TMenuItem
* MenuItem3: TMenuItem
* MenuItem4: TMenuItem
* MenuItem5: TMenuItem
* MenuItem6: TMenuItem
* MenuItem7: TMenuItem
* MenuItem8: TMenuItem

Процедуры:

* procedure ButtonEditClick(Sender: TObject)
* procedure MenuItem1Click(Sender: TObject)
* procedure MenuItem3Click(Sender: TObject)
* procedure MenuItem4Click(Sender: TObject)
* procedure MenuItem5Click(Sender: TObject)
* procedure MenuItem6Click(Sender: TObject)
* procedure MenuItem7Click(Sender: TObject)
* procedure MenuItem8Click(Sender: TObject)

Одной из функций является сортировка данных таблицы:

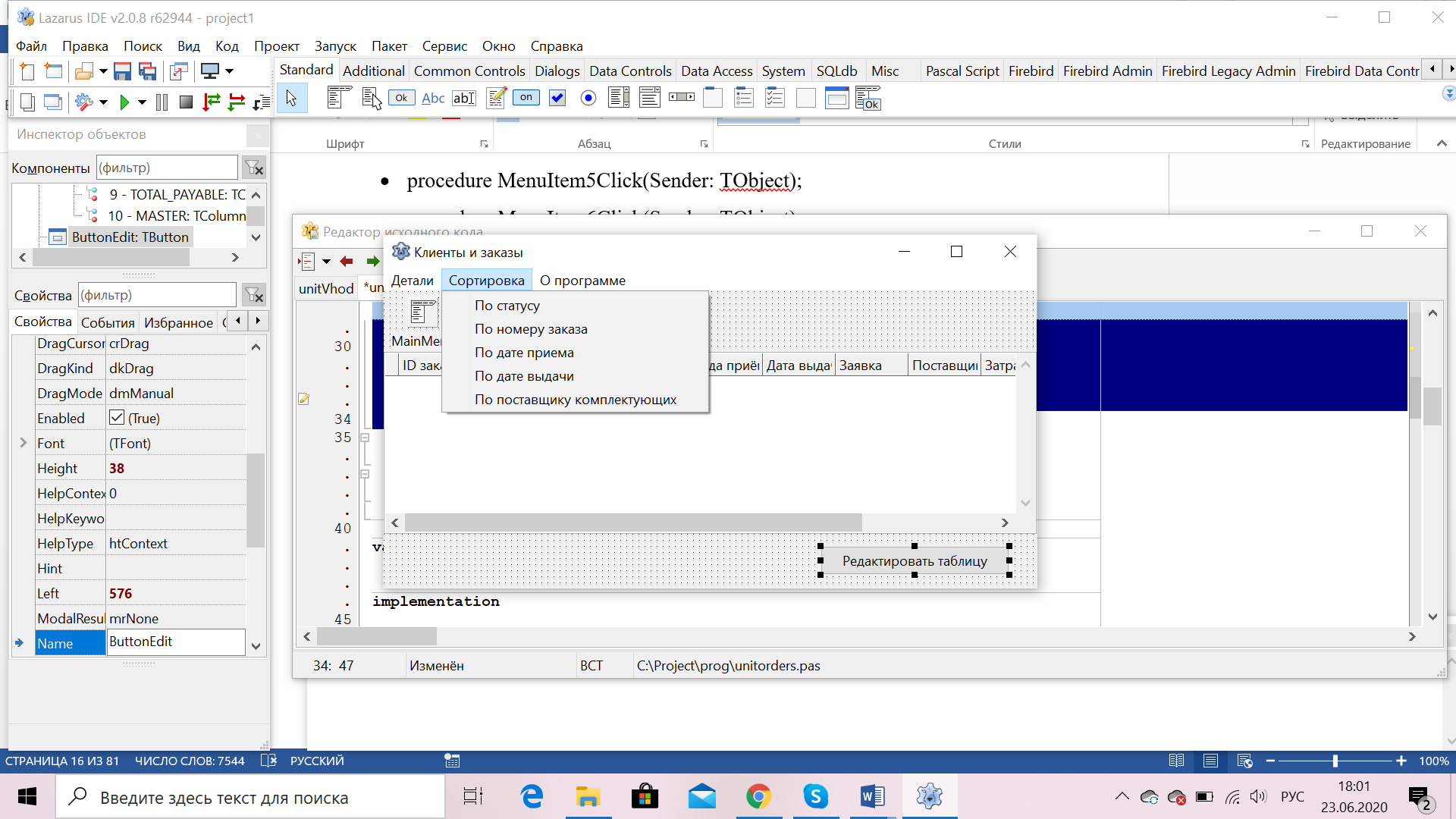


Рисунок 9. Сортировка

Также пользователь имеет возможность открыть окно с деталями, справку о программе и выбрать режим редактирования данных. Код данной формы представлен в Приложении 2.

Окно редактирования заявок:

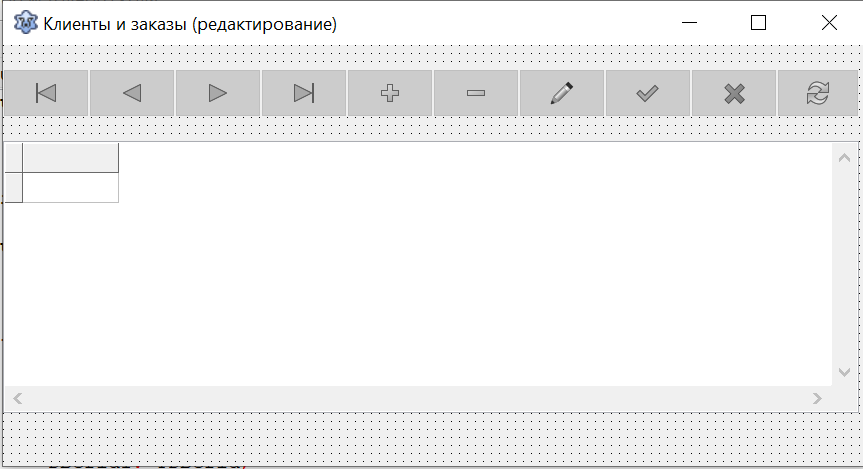


Рисунок 10. Окно редактирования заявок (FormOrdersEdit)

Компоненты:

* DBGrid1: TDBGrid
* DBNavigator1: TDBNavigator

Данная форма не содержит ни одной процедуры, на ней имеется одна таблица, содержащая в себе те же данные о заказе, и панель для редактирования, добавления и удаления заказов. Код данной формы представлен в Приложении 3.

Окно просмотра и редактирования деталей заявок:

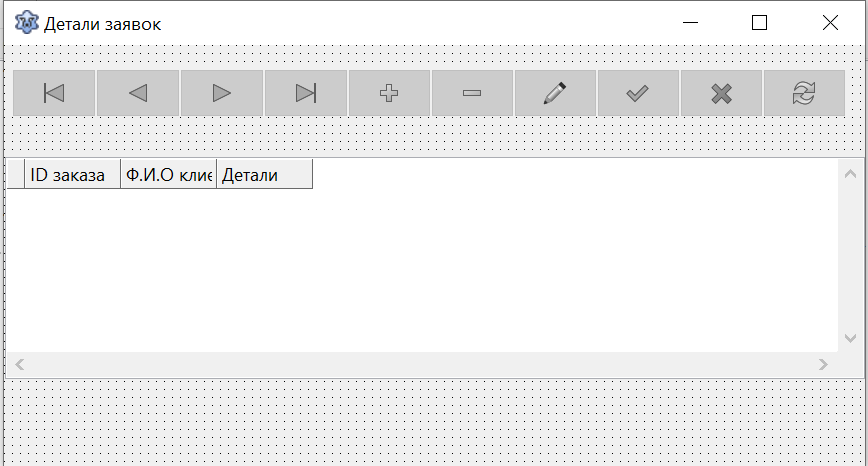


Рисунок 11. Окно просмотра и редактирования деталей заявок (FormDetails)

Компоненты:

* DBGrid1: TDBGrid
* DBNavigator1: TDBNavigator

Данная форма не содержит ни одной процедуры, на ней имеется одна таблица, содержащая в себе данные о деталях заказа, и панель для редактирования, добавления и удаления этих деталей заказов. Код данной формы представлен в Приложении 4.

Окно справки о программе:

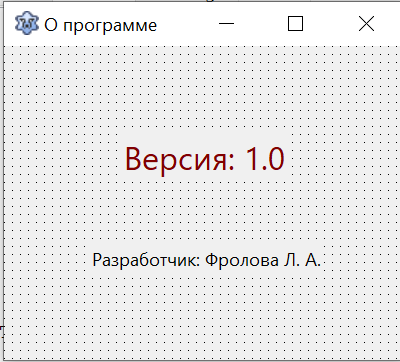


Рисунок 12. Окно справки о программе (FormProg)

Компоненты:

* Label1: TLabel
* Label2: TLabel

Данная форма не имеет процедур, она не даёт пользователю никаких дополнительных возможностей, а носит лишь ознакомительный характер. Код данной формы представлен в Приложении 5.

Модуль данных:

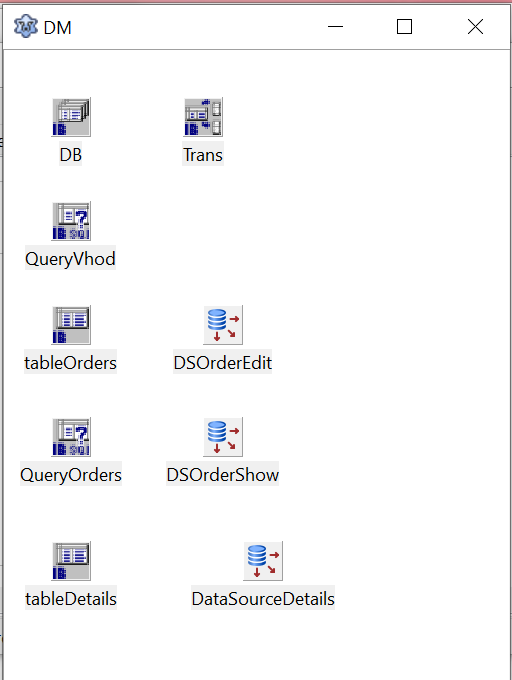


Рисунок 13. Модуль данных (DM)

Компоненты:

* DSOrderShow: TDataSource
* DataSourceDetails: TDataSource
* DSOrderEdit: TDataSource
* DB: TIBDatabase
* QueryOrders: TIBQuery
* tableDetails: TIBTable
* tableDetailsDETAILS: TIBStringField
* tableDetailsFIO\_CLIENT: TIBStringField
* tableDetailsID\_ORDER: TIBIntegerField
* tableOrders: TIBTable
* QueryVhod: TIBQuery
* tableOrdersDATE\_ISSUE: TDateField
* tableOrdersDATE\_RECEIPT: TDateField
* tableOrdersEXPENSES: TIBStringField
* tableOrdersFIO\_CLIENT: TIBStringField
* tableOrdersID\_ORDER: TIBIntegerField
* tableOrdersMASTER: TIBStringField
* tableOrdersPHONE\_CLIENT: TIBStringField
* tableOrdersPROVIDER: TIBStringField
* tableOrdersREQUEST: TIBStringField
* tableOrdersSTATUS\_ORDER: TIBStringField
* tableOrdersTOTAL\_PAYABLE: TIBStringField
* Trans: TIBTransaction

Процедуры:

* DataModuleCreate(Sender: TObject)

Модуль данных можно содержать только невизуальные компоненты. Модуль данных доступен разработчику на этапе разработки. Пользователь не может увидеть модуль данных во время выполнения. Код представлен в Приложении 6.

## Описание тестовых наборов модулей

Тестирование - это проверка соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом, чтобы в ходе пользования программы не возникли проблемы и/или ошибки.

**Автоматический метод:**

В программе «Компьютерный мастер» используются автоматические методы, которые проверяют ту или иную часть программы во время пользования (а также во время запуска и закрытия программы) конечного пользователя, среди них можно выделить следующие методы:

* Проверка на правильность введённых данных – она проверяет такие условия, как:
  + Написано ли слово на латинице;
  + Используются ли числа при вводе чисел;
  + Используются ли буквы при вводе слов;
  + Правильность введенной даты;
  + Заполнены ли все данные заявки.
* Проверка таблиц, строк и столбцов:
  + Имя всех таблиц в базе данных;
  + Имена столбцов для каждой таблицы;
  + Типы столбцов для каждой таблицы;
  + Значение NULL отмечено или нет;
  + Целостность данных.

**Примеры тестирования:**

Для отладки программы были разработаны следующие тестовые вводные данные и сценарии:

1. **Авторизация**

**Константы:** Логин для мастера Абрамова = abramov, Пароль = 123qwd

**Входные данные:** Логин, пароль.

**Тестовый сценарий работы:** Мастер авторизуется под своей учетной записью, вводя логин и пароль. Логин и пароль могут состоять только из латинских букв и цифр. В случае если мастер включит русскую раскладку или допустит ошибку в логине или пароле, то программа выдаст сообщение о неправильном вводе. А если мастер введет все правильно, то программа откроет таблицу с данными клиентов.

**Ожидаемый результат:** Сообщение об успешном входе.

**Тестовые наборы:**

1) Ввожу логин: abramov

Ввожу пароль: 123qwd

Результат: будет произведен вход.

2) Ввожу логин: абрамов

Ввожу пароль: 123qwd

Результат: неверный логин или пароль.

3) Ввожу логин: abramov

Ввожу пароль: 123йцв

Результат: неверный логин или пароль.

## Описание применения средств отладки

При компиляции проекта были обнаружены ошибки.

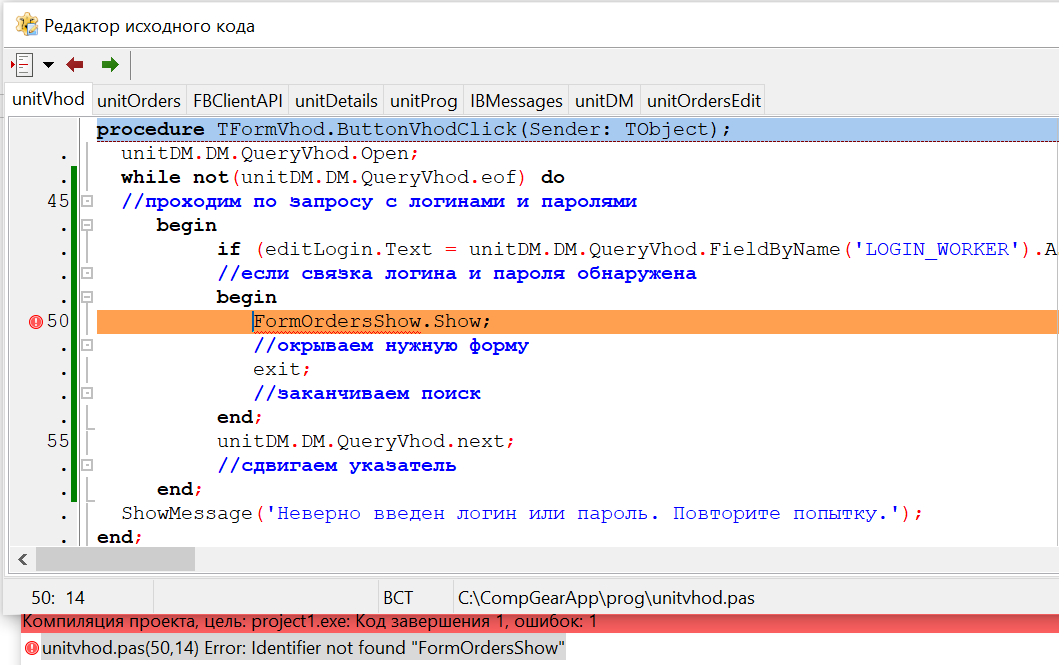


Рисунок 14. Ошибка компиляции

Решением было подключить файл unitOrders, ибо иначе форма с названием FormOrdersShow не могла найтись.

Также удалились некоторое пустые процедуры, которые создались нечаянным нажатием на компоненты форм, например:

procedure TFormVhod.EditLoginChange(Sender: TObject);

begin

end;

procedure TFormVhod.EditPasswordChange(Sender: TObject);

begin

end;

procedure TFormVhod.Label3Click(Sender: TObject);

begin

end;

## Анализ оптимальности использования памяти и быстродействия

В данном разделе будет проведен анализ оптимальности использования памяти и быстродействия программы.

Список принятых оптимальных решений:

* Было принято решение об использовании многотабличных запросов по индексам, что позволило сократить время поиска так как был использован двоичный или бинарный поиск, так же это значимо сыграло на объеме памяти, так как вместо нескольких весомых палей использовалось только одно поле индивидуального номера;
* Для уменьшения размера файла необходимо отключить «Генерировать отладочную информацию» и включить «Вырезать символы из исполняемого файла».
* Так же необходимо включить «Умная компоновка» в разделе «Cтиль модуля» и «Компоновка».

# ЭКСПЛУТАЦИОННАЯ ЧАСТЬ

## . Эксплуатационный раздел

Эксплуатационный раздел содержит сведения о назначении, составе, принципе действия и конструкции программно-технического комплекса «CompGearApp», его технические характеристики, а также другие сведения, необходимые для правильной эксплуатации программного обеспечения. Комплект поставки «CompGearApp» включает в себя непосредственно программу и эксплуатационную документацию в составе настоящего руководства по эксплуатации, руководства оператора и правил техники безопасности.

Для внедрения и последующей эксплуатации программы персонал должен изучить эксплуатационную и другую техническую документацию на программное обеспечение «CompGearApp», а случае возникновения вопросов персонал может проконсультироваться с разработчиком ПО «CompGearApp».

## Рекомендации по эксплуатации программного продукта

Данная программа обладает относительно низкими требованиями к аппаратному и программному обеспечению оборудования для своего выполнения с адекватной скоростью. Минимальные требования программного продукта приведены в таблице 3.1.

Таблица 3.1.

Тактико-технические характеристики пользователя

|  |  |  |
| --- | --- | --- |
| **Тип оборудования** | **Наименование характеристики** | **Значение** |
| Процессор | Количество, не менее | 1 |
| Количество ядер, не менее | 1 |
| Частота, не менее | 1000 МГц |
| Память | Тип, не хуже | DDR3 |
| Количество слотов, не меньше | 1 DIMM |
| Частота, не менее | 1600 МГц |
| Объем, не менее | 4 (2 × 2) ГБ |
| Сетевой контроллер | Пропускная способность, не хуже | 1Гбит/с Ethernet |
| Жесткие диски | Тип, не хуже | SATA–III |
| Количество, не менее | 1 |
| Объем, не менее | 200 ГБ |
| Видеопамять | Тип, не хуже | GDDR3 |
| Частота, не менее | 1200 МГц |
| Объем, не менее | 512 Мб |
| Оптический привод | Тип, не хуже | DVD-RW |
| Операционная система | Версия, не хуже | Windows 7 (SP1) |
| Монитор | Тип, не хуже | LCD |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
|  | Разрешение, не менее | 24″,1920×1080 |
| Клавиатура | Тип, не хуже | Мембранная |
| Мышь | Тип, не хуже | Лазерная |
| Источник бесперебойного электропитания | Тип, не хуже | UPS |
| Мощность, не менее | Не должна быть меньше потребляемой мощности компьютера, на котором запущена программы. |

Другие выполняемые программы не должны препятствовать нормальной работе программы «CompGearApp», и не должны блокировать работу пользователя с графическим интерфейсом.

Программа рассчитана на применение сотрудниками профильных подразделений объектов заказчика. Программа подразумевает использование тремя единицами персонала: мастером-оценщиком, который наделён расширенными правами в программе, мастером-приемщиком, который будет работать с программой большую часть рабочего времени и мастер-сборщик, он будет узнавать подробности заказа.

Системный администратор должен иметь профильное образование и сертификаты компании-производителя операционной системы. В перечень задач, выполняемых системным администратором, должны входить:

* Задача поддержания работоспособности технических средств;
* Задачи установки (инсталляции) и поддержания работоспособности системных программных средств — операционной системы;
* Задача установки (инсталляции) программы.

Программа должна быть рассчитана на эксплуатацию в условиях, соответствующих требованиям ГОСТ 15543.1-89. Требования к маркировке и упаковке не предъявляются. Требования к транспортировке упаковки не предъявляются. Специальные требования:

* Наличие сервера базы данных Firebird версии 2.5.

## Руководство администратора по установке программы

Для установки программы следует распаковать архив программы «CompGearApp.rar». Распаковку можно произвести с помощью программы архиватора «WinRar», «7Zip» или с помощью других программ, которые позволяют выполнить данную операцию.

## Руководство пользователя

При запуске программы необходимо авторизоваться. Для это нужно указать логин и пароль.

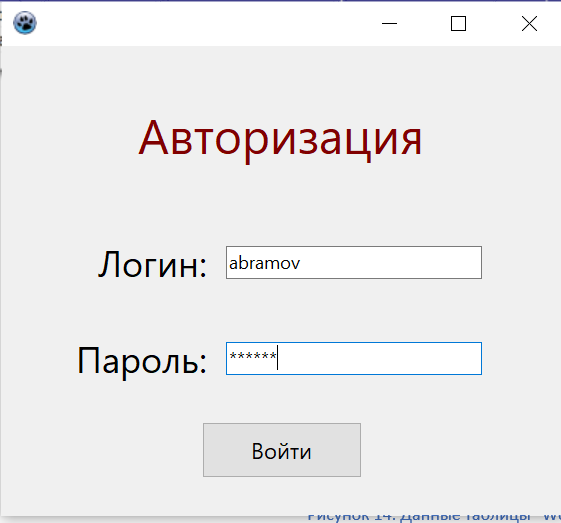


Рисунок 15. Окно авторизации

Каждому работнику выдаётся индивидуальная связка логина с паролем.

Для авторизации можно воспользоваться связками, представленными ниже (на рисунке 18):

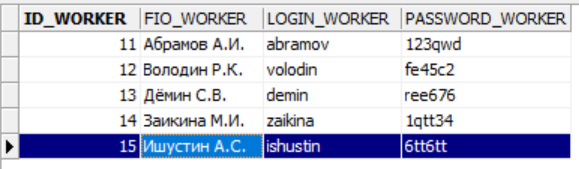


Рисунок 16. Связки логина - пароля

После входа на экране появится главное окно программы, в котором расположено главное меню программы и основная таблица с заявками клиентов:

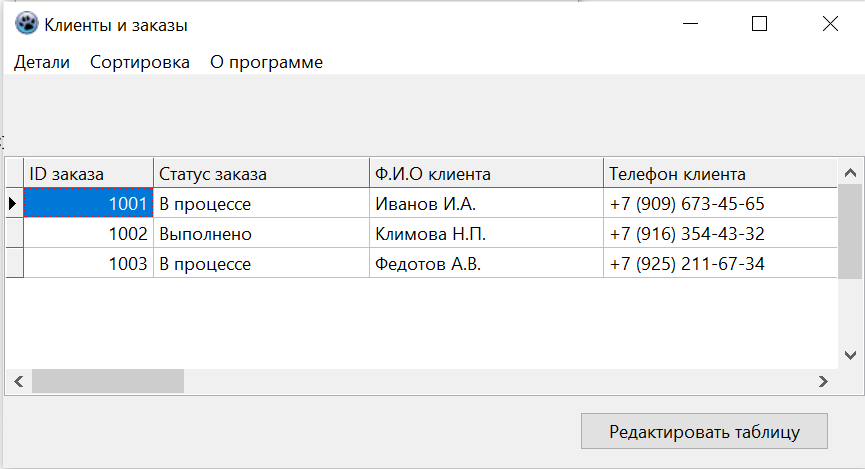


Рисунок 17. Главное окно

Пункт «Сортировка», содержит подпункты:

* По статусу;
* По номеру заказа;
* По дате приёма;
* По дате выдачи;
* По поставщику комплектующих.

При выборе одного из пунктов произойдёт сортировка данных по нужному критерию.

В случае выбора подпункта «Детали» на экране появится окно с таблицей деталей заявок клиентов:

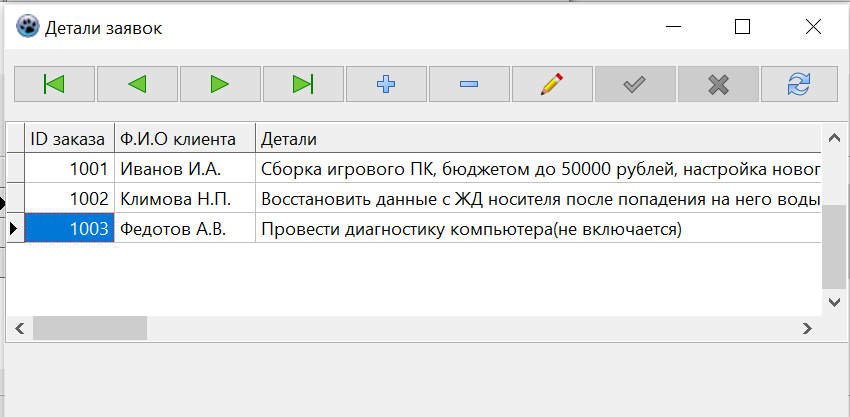


Рисунок 18. Окно деталей

Эта панель даёт возможность работать с данными таблицы.

Кнопка  даёт возможность переместиться в начало таблицы, соответственно, кнопка  перемещает Вас в конец таблицы.

Кнопки  позволяют переходить на предыдущий или следующий заказ.

Эти кнопки  позволяют добавлять новую и удалять выполненную заявку.

Кнопка  позволяет начать редактирование данных. Но на практике её можно не использовать, так как режим редактирования включается двойным нажатием на необходимую ячейку.

Кнопки  позволяют сохранить или отменить внесённые Вами изменения.

Кнопка  обновляет данные таблицы.

При нажатии на кнопку «Редактирование таблицы» откроется таблица заявок с аналогичной панелью управления:

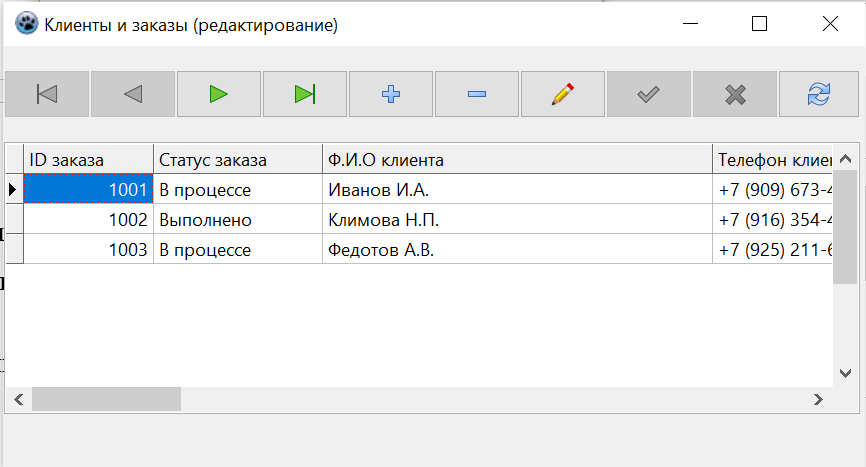


Рисунок 19. Редактирование заявок

В случае выбора пункта меню «О программе», на экране появится окно с кратким описанием программы:

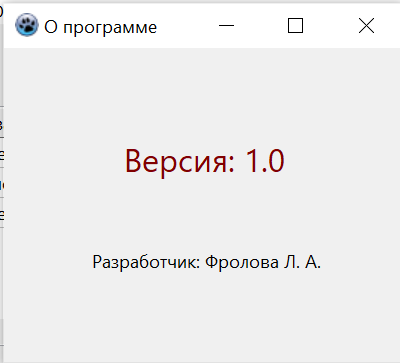


Рисунок 20. Краткое описание программы

# ЗАКЛЮЧЕНИЕ

Итогом выполнения задания на курсовой проект стала программа под названием «CompGearApp», которая позволит автоматизировать расчёт стоимости услуг в компьютерной мастерской «CompGear». Разработка программы была обусловлена необходимостью автоматизации расчёта стоимости оказываемых компьютерной мастерской услуг.

Программный продукт создан с использованием следующих программных инструментов:

* Язык программирования Object Pascal;
* Среда разработки программного обеспечения «Lazarus»;
* Программная библиотека для работы с базой данный «Firebird» «IBX».

Программа работает под управлением операционной системы Windows 7 и новее. Обязательным условием для работы программы является наличие установленного в системе сервера базы данных «Firebird».

# Список использованных источников.

1. Программа для сервисного центра // URL: <https://www.itproportal.ru/programma-uchet-remonta-v-servis-centre.html>;;
2. Lazarus // URL: <https://ru.wikipedia.org/wiki/Lazarus>;
3. PascalABC.NET // URL: <https://ru.wikipedia.org/wiki/PascalABC.NET>;
4. Автоматизация учета услуг сервисного центра по ремонту компьютерной техники // URL: <https://studbooks.net/2320637/informatika/analiz_predmetnoy_oblasti>;
5. Организация рабочего места мастера по ремонту // URL: <https://allrefrs.ru/3-52275.html>;
6. Снаряжение мастера по ремонту компьютеров для работы на выезде // URL: <http://www.1st.rv.ua/snaryazhenie-mastera-remont-kompyuterov/>;
7. Оборудование и инструменты для создания сервисного центра // URL: <https://toolboom.com/ru/articles-and-video/tools-and-equipment-for-setting-up-a-service-center/>;
8. Свой бизнес по обслуживанию и ремонту компьютеров // URL: <https://www.openbusiness.ru/biz/business/svoy-biznes-po-kompyuternomu-obsluzhivaniyu-i-remontu/>;
9. Firebird // URL: <https://ru.wikipedia.org/wiki/Firebird>;
10. Firebird: The true open source database for Windows, Linux, Mac OS X and more // URL: <https://firebirdsql.org/>;
11. Общие принципы ремонта компьютерной техники // URL: <http://krotek.ru/poleznaya-informatsiya/116-obshie-princepi-remonta-komp-tehniki>;
12. Как осуществляется ремонт компьютеров и ноутбуков // URL: <https://system-blog.ru/kak-osushhestvlyaetsya-remont-kompyuterov-i-noutbukov>;

Приложение 1 (unitVhod.pas)

unit unitVhod;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, unitOrders;

type

{ TFormVhod }

TFormVhod = class(TForm)

ButtonVhod: TButton;

EditPassword: TEdit;

EditLogin: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

procedure ButtonVhodClick(Sender: TObject);

private

public

end;

var

FormVhod: TFormVhod;

implementation

{$R \*.lfm}

uses unitDM;

{ TFormVhod }

procedure TFormVhod.ButtonVhodClick(Sender: TObject);

//при нажатии на кнопку "вход"

begin

unitDM.DM.QueryVhod.Close;

unitDM.DM.QueryVhod.Open;

while not(unitDM.DM.QueryVhod.eof) do

//проходим по запросу с логинами и паролями

begin

if (editLogin.Text = unitDM.DM.QueryVhod.FieldByName('LOGIN\_WORKER').AsString) and (editPassword.Text = unitDM.DM.QueryVhod.FieldByName('PASSWORD\_WORKER').AsString) then

//если связка логина и пароля обнаружена

begin

FormOrdersShow.Show;

//окрываем нужную форму

exit;

//заканчиваем поиск

end;

unitDM.DM.QueryVhod.next;

//сдвигаем указатель

end;

ShowMessage('Неверно введен логин или пароль. Повторите попытку.');

end;

end.

Приложение 2 (unitOrders.pas)

unit unitOrders;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, DBGrids, ComCtrls,

DBCtrls, Menus, StdCtrls, unitProg, unitDetails, unitOrdersEdit, unitDM;

type

{ TFormOrdersShow }

TFormOrdersShow = class(TForm)

ButtonEdit: TButton;

DBGrid1: TDBGrid;

MainMenu: TMainMenu;

MenuItem1: TMenuItem;

MenuItem2: TMenuItem;

MenuItem3: TMenuItem;

MenuItem4: TMenuItem;

MenuItem5: TMenuItem;

MenuItem6: TMenuItem;

MenuItem7: TMenuItem;

MenuItem8: TMenuItem;

procedure ButtonEditClick(Sender: TObject);

procedure MenuItem1Click(Sender: TObject);

procedure MenuItem3Click(Sender: TObject);

procedure MenuItem4Click(Sender: TObject);

procedure MenuItem5Click(Sender: TObject);

procedure MenuItem6Click(Sender: TObject);

procedure MenuItem7Click(Sender: TObject);

procedure MenuItem8Click(Sender: TObject);

private

public

end;

var

FormOrdersShow: TFormOrdersShow;

implementation

{$R \*.lfm}

{ TFormOrdersShow }

procedure TFormOrdersShow.MenuItem3Click(Sender: TObject);

//вызов формы справки о программе

begin

FormProg.Show;

end;

procedure TFormOrdersShow.MenuItem4Click(Sender: TObject);

//сортировка таблицы по статусу заказа

begin

with unitDM.DM.QueryOrders do

begin

Close;

SQL.Clear;

//очищаем sql-запрос

SQL.Add('select \* from CLIENTS\_AND\_ORDERS order by STATUS\_ORDER');

//добавляем новый sql-запрос с сортировкой

Open;

end;

end;

procedure TFormOrdersShow.MenuItem5Click(Sender: TObject);

//сортировка таблицы по ID заказа

begin

with unitDM.DM.QueryOrders do

begin

Close;

SQL.Clear;

//очищаем sql-запрос

SQL.Add('select \* from CLIENTS\_AND\_ORDERS order by ID\_ORDER');

//добавляем новый sql-запрос с сортировкой

Open;

end;

end;

procedure TFormOrdersShow.MenuItem6Click(Sender: TObject);

//сортировка таблицы по дате приёма заказа

begin

with unitDM.DM.QueryOrders do

begin

Close;

SQL.Clear;

//очищаем sql-запрос

SQL.Add('select \* from CLIENTS\_AND\_ORDERS order by DATE\_RECEIPT');

//добавляем новый sql-запрос с сортировкой

Open;

end;

end;

procedure TFormOrdersShow.MenuItem7Click(Sender: TObject);

//сортировка таблицы по дате выдачи заказа

begin

with unitDM.DM.QueryOrders do

begin

Close;

SQL.Clear;

//очищаем sql-запрос

SQL.Add('select \* from CLIENTS\_AND\_ORDERS order by DATE\_ISSUE');

//добавляем новый sql-запрос с сортировкой

Open;

end;

end;

procedure TFormOrdersShow.MenuItem8Click(Sender: TObject);

//сортировка таблицы по поставщику комплектующих

begin

with unitDM.DM.QueryOrders do

begin

Close;

SQL.Clear;

//очищаем sql-запрос

SQL.Add('select \* from CLIENTS\_AND\_ORDERS order by PROVIDER');

//добавляем новый sql-запрос с сортировкой

Open;

end;

end;

procedure TFormOrdersShow.MenuItem1Click(Sender: TObject);

//вывоз формы с деталями заявок

begin

FormDetails.Show;

end;

procedure TFormOrdersShow.ButtonEditClick(Sender: TObject);

//вызов формы с возможностью редактирования таблицы

begin

FormOrdersEdit.Show;

end;

end.

Приложение 3 (unitOrdersEdit.pas)

unit unitOrdersEdit;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, DBGrids, DBCtrls,

IBTable;

type

{ TFormOrdersEdit }

TFormOrdersEdit = class(TForm)

DBGrid1: TDBGrid;

DBNavigator1: TDBNavigator;

private

public

end;

var

FormOrdersEdit: TFormOrdersEdit;

implementation

{$R \*.lfm}

end.

Приложение 4 (unitDetails.pas)

unit unitDetails;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, DBGrids, DBCtrls;

type

{ TFormDetails }

TFormDetails = class(TForm)

DBGrid1: TDBGrid;

DBNavigator1: TDBNavigator;

private

public

end;

var

FormDetails: TFormDetails;

implementation

{$R \*.lfm}

end.

Приложение 5 (unitProg.pas)

unit unitProg;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls;

type

{ TFormProg }

TFormProg = class(TForm)

Label1: TLabel;

Label2: TLabel;

private

public

end;

var

FormProg: TFormProg;

implementation

{$R \*.lfm}

end.

Приложение 6 (unitDM.pas)

unit unitDM;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, db, IBDatabase, IBQuery, IBTable, IBCustomDataSet, Dialogs;

type

{ TDM }

TDM = class(TDataModule)

DSOrderShow: TDataSource;

DataSourceDetails: TDataSource;

DSOrderEdit: TDataSource;

DB: TIBDatabase;

OpenDialog: TOpenDialog;

QueryOrders: TIBQuery;

tableDetails: TIBTable;

tableDetailsDETAILS: TIBStringField;

tableDetailsFIO\_CLIENT: TIBStringField;

tableDetailsID\_ORDER: TIBIntegerField;

tableOrders: TIBTable;

QueryVhod: TIBQuery;

tableOrdersDATE\_ISSUE: TDateField;

tableOrdersDATE\_RECEIPT: TDateField;

tableOrdersEXPENSES: TIBStringField;

tableOrdersFIO\_CLIENT: TIBStringField;

tableOrdersID\_ORDER: TIBIntegerField;

tableOrdersMASTER: TIBStringField;

tableOrdersPHONE\_CLIENT: TIBStringField;

tableOrdersPROVIDER: TIBStringField;

tableOrdersREQUEST: TIBStringField;

tableOrdersSTATUS\_ORDER: TIBStringField;

tableOrdersTOTAL\_PAYABLE: TIBStringField;

Trans: TIBTransaction;

procedure DataModuleCreate(Sender: TObject);

private

public

end;

var

DM: TDM;

implementation

{$R \*.lfm}

{ TDM }

procedure TDM.DataModuleCreate(Sender: TObject);

begin

if(OpenDialog.Execute) then

//если выбран файл

DB.DatabaseName:='127.0.0.1:' + ExtractFileDir(OpenDialog.FileName)+'\DB.FDB';

//подставляет в базу данных выбранный путь

DB.Connected:=True;

//связываемся с бд

Trans.StartTransaction;

//активируем транзакцию

tableOrders.Open;

QueryOrders.Open;

tableDetails.Open;

//активируем таблицы

end;

end.