



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

ОТЧЕТ

по производственной практике ПП.01.01

по профессиональному модулю ПМ.01

**«Разработка программных модулей программного обеспечения
для компьютерных систем»**

Специальность 09.02.03 «Программирование в компьютерных
системах»

Выполнил студент 3 курса группы П1-16

Миронов Василий Анатольевич

_____ (подпись)

Принял преподаватель

Гусятинер Л.Б.

_____ (подпись)

_____ (оценка)

Королев, 2019

Содержание

Введение.....	3
1. Общие сведения об организации	4
1.1. Структура организации	4
Структура организации представлена на Рис. 1.1.1	4
1.2. Структура IT-отдела	5
1.3. Основные функции отдела	6
1.4. Нормативные документы, которые регламентируют профессиональную деятельность	6
1.5. Документооборот организации	7
1.6. Информационные технологий предприятия.....	9
1.7. Программное обеспечение предприятия.....	9
1.8. Задачи, подлежащие автоматизации	10
2. Разработка отдельных программных модулей и их листинги.....	11
2.1. Создание базы данных.....	11
2.2. Создание проекта в QT creator.....	11
2.3.Подключение БД к программе.....	12
2.4. Создание формы.....	14
2.5. Вывод данных БД в программу.....	14
2.6. Сортировка.....	15
2.7. Поиск.....	16
2.8. Кнопки удалить и статус.....	17
2.9. Код проекта	19
3. Вывод.....	27
4. Дневник практики	28
5. Список использованной литературы.....	29
6. Приложения.....	30

Введение

На 3 курсе обучения в ККМТ, студентом группы П1-16 Мироновым Василием была пройдена производственная практика по модулю ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем» на предприятии ООО «ГлобалТекстиль». Студент получил задание написать программу, анализирующую заказы интернет-магазина, а также изучить структуру предприятия, на котором он проходит практику.

1. Общие сведения об организации

1.1. Структура организации

Структура организации представлена на Рис. 1.1.1

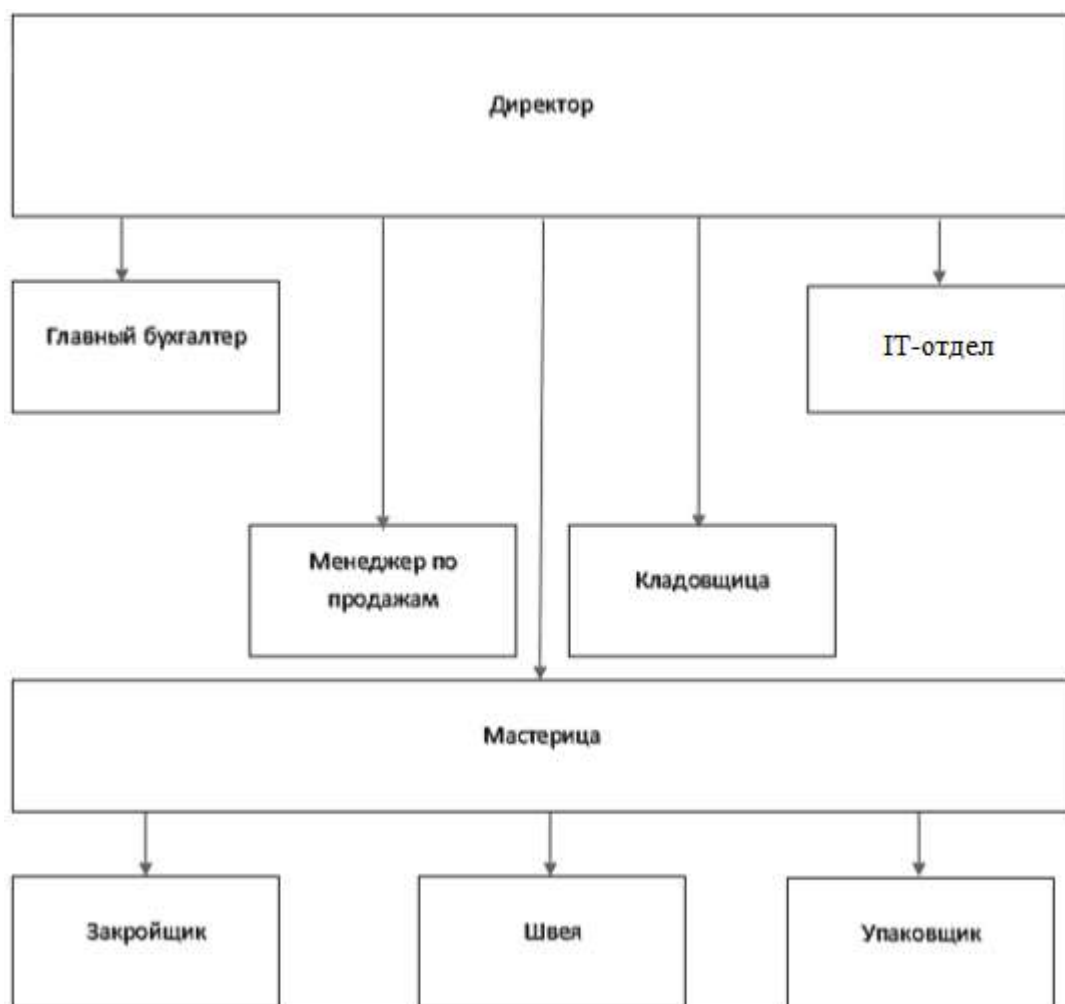


Рис. 1.1.1 Структура организации

1.2. Структура IT-отдела

Во главе стоит руководитель отдела, в его подчинении находятся специалист по поддержке пользователя и системный администратор, техники-программисты выполняют поручения инженера-программиста.

Организационная структура отдела представлена на Рис. 1.2.1.



Рис. 1.2.1. Организационная структура IT-отдела.

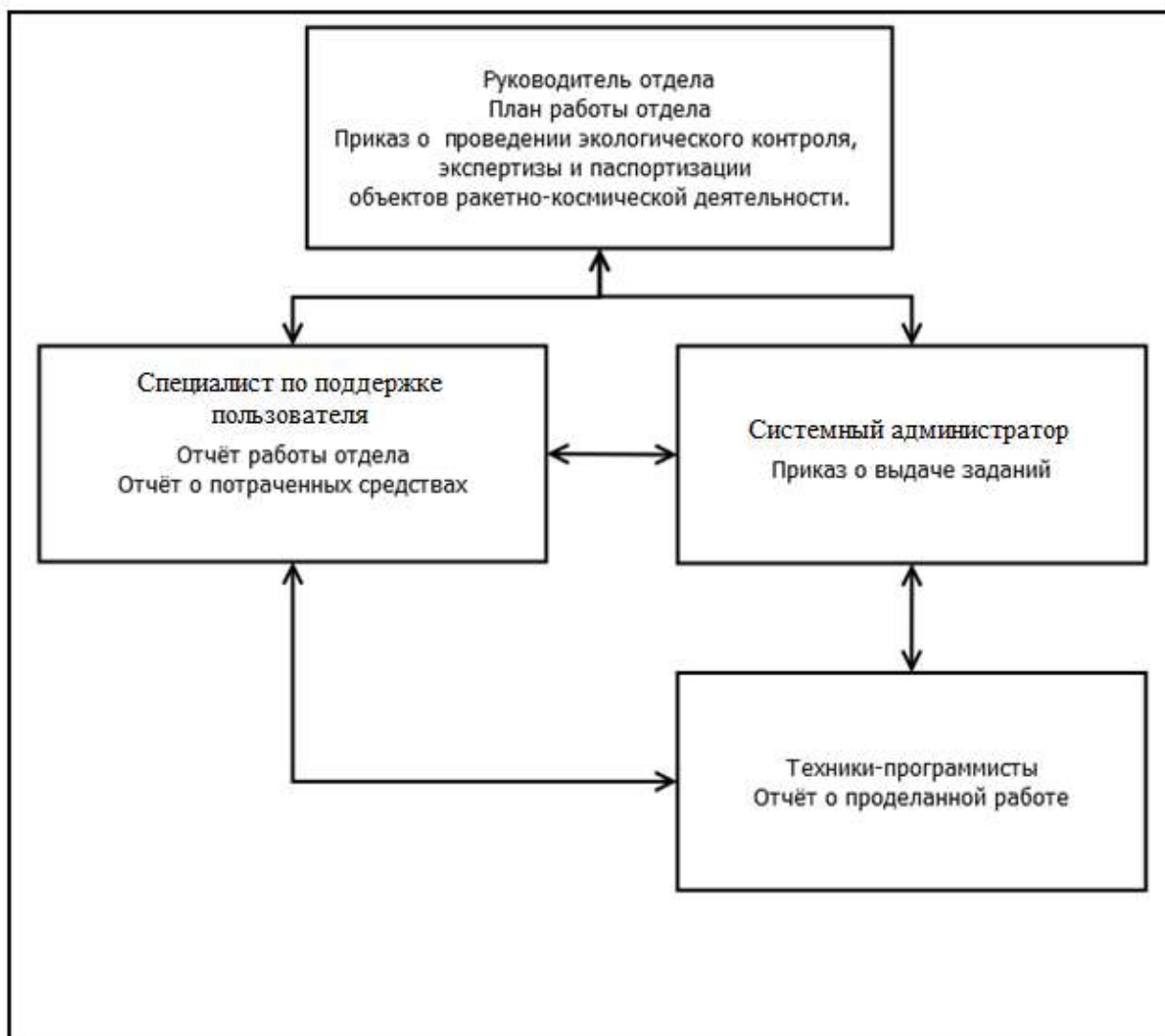


Рис. 1.2.2. Информационная структура IT-отдела

1.3. Основные функции отдела

- Разработка, отладка анализ и оптимизация программного кода на основе готовых спецификаций;
- Интегрирование программных компонентов;

1.4. Нормативные документы, которые регламентируют профессиональную деятельность

- Политика информационной безопасности организации;

- Положение по обеспечению информационной безопасности организации;
- Положение о порядке доступа к информационным ресурсам;
- Положение о коммерческой или служебной тайне;
- Порядок обращения с информацией, подлежащей защите;
- Порядок защиты от несанкционированного доступа к информации и незаконного вмешательства в процесс функционирования информационной системы;
- План обеспечения непрерывной работы IT-сервисов и восстановления в случае возникновения кризисных ситуаций;
- Положение об отделе информационной безопасности;
- Классификация и перечень основных видов угроз информационной безопасности предприятия;
- Требования и рекомендации по обеспечению информационной безопасности предприятия;
- Положение об отделе технической защиты информации;

1.5. Документооборот организации

Основная задача документооборота – документирование текущей деятельности компании. Этот сложный технологический процесс подразумевает создание, прием, передачу, согласование и хранение документации, обеспечивающей работу всей компании. Маршрут движения документа и время его прохождения от одного этапа к другому – ключевые параметры этого процесса.

Важность документооборота для современного делопроизводства выражается в том, он выделен в отдельную сферу деятельности.

Предметной областью этой деятельности являются инстанции движения документов и скорость этого движения. Оптимизация механизма движения деловых бумаг существенным образом повышает эффективность ключевых бизнес-процессов предприятия. От работы этого механизма напрямую зависит эффективность хранения информации, возможность поиска в базе нужных сведений, а также современное обеспечение пользователей системы оперативными данными. Продуктивность административных решений в значительной степени обусловлена актуальностью и релевантностью той информации, на основе которой они принимаются. В этой связи организация документооборота является ключевым компонентом общей системы управления. В ее приоритетные задачи входит контроль исполнения руководящих решений, разработка поисковых систем и информационное обеспечение работы непосредственных исполнителей.



Рис. 1.4.1 Документооборот организации

1.6. Информационные технологий предприятия

- GitHub

1.7. Программное обеспечение предприятия

- Windows 10
- Microsoft office 2016
- Qt Creator
- Notepad++
- SQLite Studio 3.2.1
- PyCharm 2019.1.3

1.8. Задачи, подлежащие автоматизации

- Создание систем управления рисками предприятий, органов исполнительной власти федерального и муниципального уровней.
- Разработка систем управления проектами создания сложных технических систем.
- Оценка рисков проектов создания сложных технических систем.

2. Разработка отдельных программных модулей и их листинги.

2.1. Создание базы данных.

Для создания базы данных было решено использовать СУБД SQLite. По заданию в таблице должны быть 7 полей:

- Id – номер заказа в таблице
- Number – номер заказа
- Data – дата заказа
- Fio – ФИО заказчика
- Address – адрес заказа
- Sum – сумма заказа
- Status – статус заказа (не обработан, обработан, ожидание, отменен)

Создание таблицы и полей таблицы происходило путем написания запросов в командной строке:

```
sqlite> create table summary_of_orders(id integer, number varchar(255), data varchar(255), fio varchar(255), address varchar(255), sum varchar(255), status varchar(255));
```

Рис.2.1”Создание базы данных с таблицей в СУБД SQLite”

2.2. Создание проекта в QT creator.

Для работы с проектом была выбрана среда QT creator.

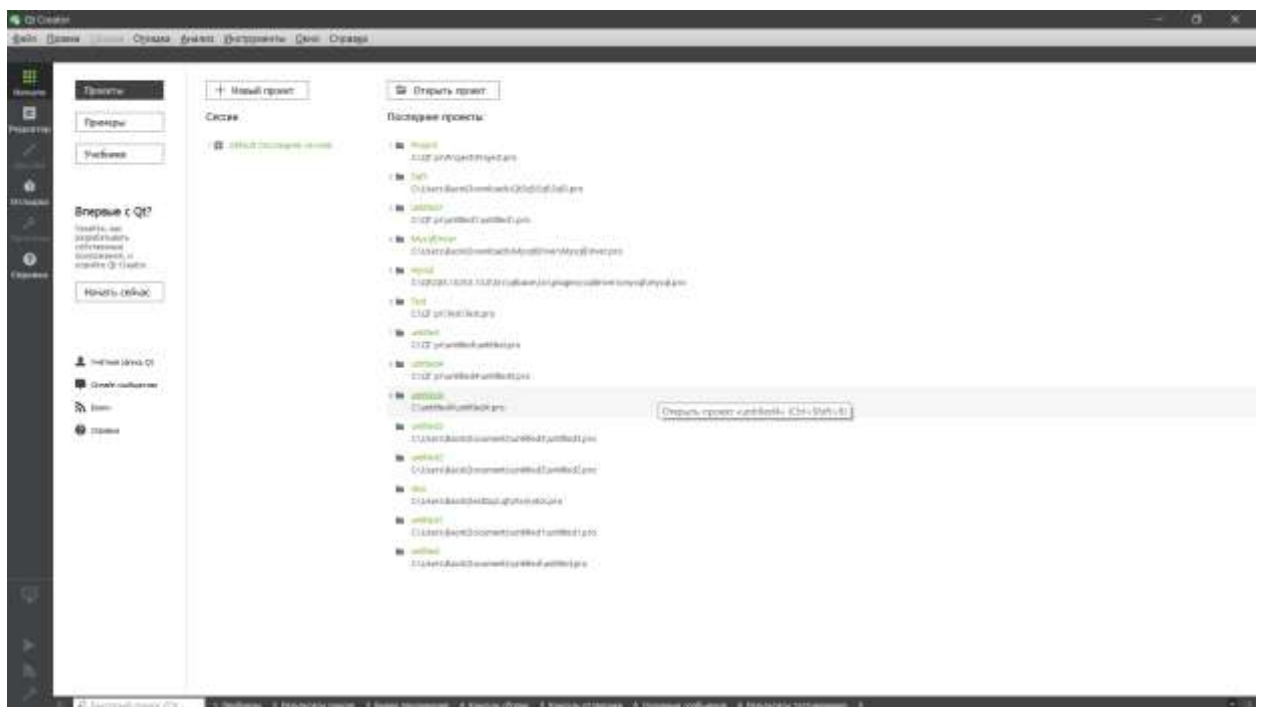


Рис.2.2”Создание проекта”

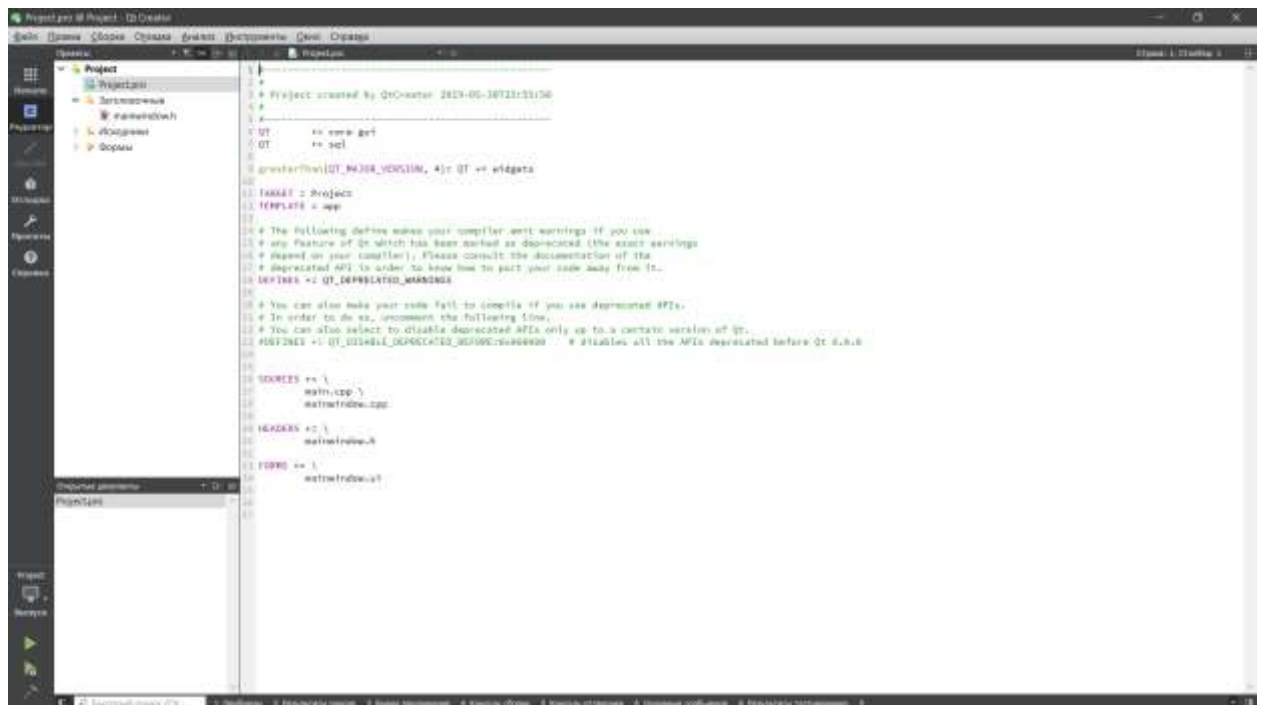


Рис.2.3 “Проект в среде разработки QT”

2.3.Подключение БД к программе.

```
db = QSqlDatabase::addDatabase("QSQLITE");
db.setDatabaseName("summary_of_orders.db");
db.open();
```

Рис.2.4”Подключение БД”

```
void MainWindow::connectToSql()//проверка открытия бд
{
    if(db.open()){
        qry = new QSqlQuery(db);
        if (ui->comboBox->currentText() == "Все"){
            qry->exec("select * from summary_of_orders");

            conclusion();
        }
    }else {
        QMessageBox::critical(this,"ERROR",model->lastError().text());
    }
}
```

Рис.2.5”Проверка открытия БД”

2.4. Создание формы.

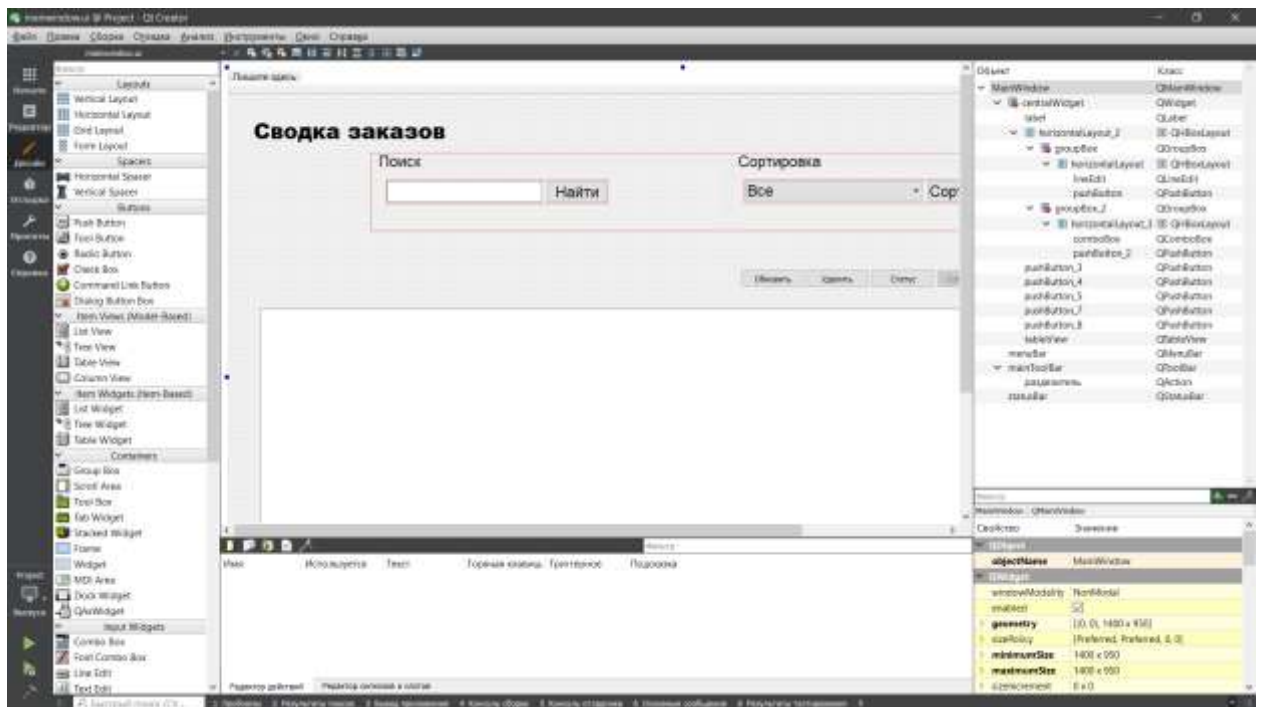


Рис.2.6"Создание формы"

2.5. Вывод данных БД в программу.

```
void MainWindow::conclusion()//вывод данных
{

    model->setQuery(*qry);
    ui->tableView->horizontalHeader()->setSectionResizeMode(QHeaderView::ResizeToContents);
    model->setHeaderData(0,Qt::Horizontal, tr("ID"));
    model->setHeaderData(1,Qt::Horizontal, tr("Номер заказа"));
    model->setHeaderData(2,Qt::Horizontal, tr("Дата заказа"));
    model->setHeaderData(3,Qt::Horizontal, tr("ФИО"));
    model->setHeaderData(4,Qt::Horizontal, tr("Адрес"));
    model->setHeaderData(5,Qt::Horizontal, tr("Сумма"));
    model->setHeaderData(6,Qt::Horizontal, tr("Статус"));
    ui->tableView->horizontalHeader()->setStyleSheet("QHeaderView::section { background-color:silver }");
    ui->tableView->setModel(model);

}
```

Рис.2.7"Вывод данных БД в окто TableView"

2.6. Сортировка.

```
void MainWindow::on_pushButton_2_clicked()//сортировка
{
    if (ui->comboBox->currentText() == "По дате(воз)"){
        qry->exec("select * from summary_of_orders order by data asc");}
    if (ui->comboBox->currentText() == "По дате(уб)"){
        qry->exec("select * from summary_of_orders order by data desc");}
    if (ui->comboBox->currentText() == "По алфавиту"){
        qry->exec("select * from summary_of_orders order by fio asc");}
    if (ui->comboBox->currentText() == "По сумме(воз)"){
        qry->exec("select * from summary_of_orders order by sum asc");}
    if (ui->comboBox->currentText() == "По сумме(уб)"){
        qry->exec("select * from summary_of_orders order by sum desc");}
    if (ui->comboBox->currentText() == "По номеру заказа(воз)"){
        qry->exec("select * from summary_of_orders order by number asc");}
    if (ui->comboBox->currentText() == "По номеру заказа(уб)"){
        qry->exec("select * from summary_of_orders order by number desc");}
    if (ui->comboBox->currentText() == "По периоду"){
        the_choice_of_period();}

    conclusion();
}
```

Рис.2.8”Сортировка”

```
void MainWindow::the_choice_of_period()//выбор периода
{
    QStringList years;
    years << QString::fromUtf8("2019") ;

    bool ok;
    QString year = QDialog::getItem(this,
                                    QString::fromUtf8("Выберите год"),
                                    QString::fromUtf8("Год:"), years, 1, false, &ok);

    QStringList items;
    items << QString::fromUtf8("Январь") << QString::fromUtf8("Февраль")
    << QString::fromUtf8("Март") << QString::fromUtf8("Апрель")
    << QString::fromUtf8("Май") << QString::fromUtf8("Июнь")
    << QString::fromUtf8("Июль") << QString::fromUtf8("Август")
    << QString::fromUtf8("Сентябрь") << QString::fromUtf8("Октябрь")
    << QString::fromUtf8("Ноябрь") << QString::fromUtf8("Декабрь");

    bool ok2;
    QString item = QDialog::getItem(this,
                                    QString::fromUtf8("Выберите месяц"),
                                    QString::fromUtf8("Месяц:"), items, 1, false, &ok2);

    QString item2;
    if (item == "Январь"){item = "01"; item2 = "02";}else
    if (item == "Февраль"){item = "02";item2 = "03";}else
    if (item == "Март"){item = "03";item2 = "04";}else
    if (item == "Апрель"){item = "04";item2 = "05";}else
    if (item == "Май"){item = "05";item2 = "06";}else
    if (item == "Июнь"){item = "06";item2 = "07";}else
    if (item == "Июль"){item = "07";item2 = "08";}else
    if (item == "Август"){item = "08";item2 = "09";}else
    if (item == "Сентябрь"){item = "09";item2 = "10";}else
    if (item == "Октябрь"){item = "10";item2 = "11";}else
    if (item == "Ноябрь"){item = "11";item2 = "12";}else
    if (item == "Декабрь"){item = "12";}

    QString period = year + "-" + item + "-" + item2;
    QString period2 = year + "-" + item2 + "-" + item;
    qry->prepare("select * from summary_of_orders "
                "where data >= :period1 "
                "and data < :period2");
    qry->bindValue(":period1", period );
    qry->bindValue(":period2", period2);
    qry->exec();
}
```

Рис.2.9”Реализация выбора периода”

2.7. Поиск.

```
void MainWindow::on_pushButton_clicked()//поиск
{

    if (ui->lineEdit->text() != ""){

        qry->prepare("select*from summary_of_orders where "
                    "number =:serch "
                    "or data =:serch "
                    "or fio =:serch "
                    "or address =:serch "
                    "or sum =:serch" );
        qry->bindValue(":serch", ui->lineEdit->text());
        qry->exec();
    }
    conclusion();
}
```

Рис.2.10”Реализация поиска”

2.8. Кнопки удалить и статус.

```
void MainWindow::on_pushButton_3_clicked()//удаление
{
    QItemSelectionModel *select = ui->tableView->selectionModel();
    if(!select->selectedRows().isEmpty()) //проверяем выделили ли строку (не пустая ли)
    {
        int nr = select->selectedRows().first().row(); //вписываем номер строки
        if (QMessageBox::Yes == QMessageBox::question(this, "Удалить?",
            "Уверены?",
            QMessageBox::Yes|QMessageBox::No))
        {
            QString indid = ui->tableView->model()->data(ui->tableView->model()->index(nr,0)).toString();
            model1->setQuery("DELETE FROM summary_of_orders WHERE id =" + indid);
        }
    }
    else
    {
        //сообщение о том, что строка пуста
        QMessageBox::information(0, "Ошибка", "Строка не выделена.");
    }
    connectToSql();
}

void MainWindow::on_pushButton_4_clicked()//редактировать статус
{
    QItemSelectionModel *select = ui->tableView->selectionModel();
    if(!select->selectedRows().isEmpty()) //проверяем выделили ли строку (не пустая ли)
    {
        int nr = select->selectedRows().first().row(); //вписываем номер строки
        {
            QString indid = ui->tableView->model()->data(ui->tableView->model()->index(nr,0)).toString();
            QStringList items;
            items << QString::fromUtf8("Обработан") << QString::fromUtf8("Не обработан")
            << QString::fromUtf8("Перевосоры") << QString::fromUtf8("Отменен");
            bool oky;
            QString sts = QDialog::getItem(this,
                QString::fromUtf8("Выберите статус"),
                QString::fromUtf8("Статус:"), items, 1, false, &oky);
            if (sts == "Не обработан"){
                model1->setQuery("UPDATE summary_of_orders SET status = 'не обработан' WHERE id =" + indid);}else
            if(sts == "Обработан"){model1->setQuery("UPDATE summary_of_orders SET status = 'обработан' WHERE id =" + indid);}else
            if (sts == "Перевосоры"){
                model1->setQuery("UPDATE summary_of_orders SET status = 'перевосоры' WHERE id =" + indid);}else
            if (sts == "Отменен"){
                model1->setQuery("UPDATE summary_of_orders SET status = 'отменен' WHERE id =" + indid);}
        }
    }
    else
    {
        //сообщение о том, что строка пуста
        QMessageBox::information(0, "Ошибка", "Строка не выделена.");
    }
    connectToSql();
}
```

Рис.2.11”Реализация кнопки удаления”

```

void MainWindow::on_pushButton_4_clicked()//редактировать статус
{
    QItemSelectionModel *select = ui->tableView->selectionModel();
    if(!select->selectedRows().isEmpty()) //проверяем выделили ли строку (не пустая ли)
    {
        int nr = select->selectedRows().first().row(); //вписываем номер строки
        {
            QString indid = ui->tableView->model()->data(ui->tableView->model()->index(nr,0)).toString();
            QStringList items;
            items << QString::fromUtf8("Обработан") << QString::fromUtf8("Не обработан")
                << QString::fromUtf8("Переговоры") << QString::fromUtf8("Отменен");
            bool oky;
            QString sts = QDialog::getItem(this,
                                           QString::fromUtf8("Выберите статус"),
                                           QString::fromUtf8("Статус:"), items, 1, false, &oky);

            if (sts == "Не обработан"){
                model->setQuery("UPDATE summary_of_orders SET status = 'не обработан' WHERE id ="+indid);}else
            if(sts == "Обработан"){model->setQuery("UPDATE summary_of_orders SET status = 'обработан' WHERE id ="+indid);}else
            if (sts == "Переговоры"){
                model->setQuery("UPDATE summary_of_orders SET status = 'переговоры' WHERE id ="+indid);}else
            if (sts == "Отменен"){
                model->setQuery("UPDATE summary_of_orders SET status = 'отменен' WHERE id ="+indid);}

        }
    }
    else
    {
        //сообщение о том, что строка пуста
        QMessageBox::information(0, "Ошибка", "Строка не выделена.");
    }

    connectToSql();
}

```

Рис.2.12”Реализация кнопки статуса”

2.9. Код проекта

Листинг 1.(Project.pro)

```
#-----  
#  
# Project created by QtCreator 2019-05-30T23:55:50  
#  
#-----  
QT      += core gui  
QT      += sql  
  
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets  
  
TARGET = Project  
TEMPLATE = app  
  
# The following define makes your compiler emit warnings if you use  
# any feature of Qt which has been marked as deprecated (the exact warnings  
# depend on your compiler). Please consult the documentation of the  
# deprecated API in order to know how to port your code away from it.  
DEFINES += QT_DEPRECATED_WARNINGS  
  
# You can also make your code fail to compile if you use deprecated APIs.  
# In order to do so, uncomment the following line.  
# You can also select to disable deprecated APIs only up to a certain version of  
# Qt.  
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all the APIs  
# deprecated before Qt 6.0.0  
  
SOURCES += \  
    main.cpp \  
    mainwindow.cpp  
  
HEADERS += \  
    mainwindow.h  
  
FORMS += \  
    mainwindow.ui
```

Листинг 2.(mainwindow.h)

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QtSql/QtSqlDatabase>
#include <QtSql/QtSqlQueryModel>
#include <QtSql/QtSqlQuery>
#include <QtSql/QtSqlTableModel>
#include <QItemSelectionModel>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();
    void connectToSql();
    void conclusion();
    void the_choice_of_period();

private slots:

    void on_pushButton_2_clicked();

    void on_pushButton_clicked();

    void on_pushButton_3_clicked();

    void on_pushButton_7_clicked();

    void on_pushButton_8_clicked();

    void on_pushButton_4_clicked();

    void on_pushButton_5_clicked();

private:
    Ui::MainWindow *ui;
    QSqlDatabase db;
    QSqlQueryModel *modell;
    QSqlQuery *qry;
    QSqlTableModel *model;

};

#endif // MAINWINDOW_H
```

Листинг 3.(main.cpp)

```
#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);

    MainWindow w;
    w.show();

    return a.exec();
}
```

Листинг 4.(mainwindow.cpp)

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QMessageBox>
#include <QSqlError>
#include <QString>
#include <QInputDialog>
#include <QProcess>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    modell = new QSqlQueryModel;
    db = QSqlDatabase::addDatabase("QSQLITE");
    db.setDatabaseName("summary_of_orders.db");
    db.open();
    model = new QSqlTableModel(this, db);
    model->setTable("summary_of_orders");
    ui->tableView->setSelectionBehavior( QAbstractItemView::SelectRows );
    connectToSql();
}

MainWindow::~MainWindow()
{
    delete model;
    delete qry;
    delete ui;
}

void MainWindow::connectToSql() //проверка открытия бд
{
    if(db.open()){
        qry = new QSqlQuery(db);
        if (ui->comboBox->currentText() == "Бсе"){
            qry->exec("select * from summary_of_orders");

            conclusion();
        }

    }else {
        QMessageBox::critical(this, "ERROR", model->lastError().text());
    }
}

void MainWindow::conclusion() //ВЫВОД ДАННЫХ
{
    modell->setQuery(*qry);

    ui->tableView->horizontalHeader()->setSectionResizeMode(QHeaderView::ResizeToContents);
    modell->setHeaderData(0, Qt::Horizontal, tr("ID"));
    modell->setHeaderData(1, Qt::Horizontal, tr("Номер заказа"));
    modell->setHeaderData(2, Qt::Horizontal, tr("Дата заказа"));
```

```

        modell->setHeaderData(3,Qt::Horizontal, tr("ФИО"));
        modell->setHeaderData(4,Qt::Horizontal, tr("Адрес"));
        modell->setHeaderData(5,Qt::Horizontal, tr("Сумма"));
        modell->setHeaderData(6,Qt::Horizontal, tr("Статус"));
        ui->tableView->horizontalHeader()->setStyleSheet("QHeaderView::section {
background-color:silver }");
        ui->tableView->setModel(modell);

    }

void MainWindow::the_choice_of_period()//выбор периода
{
    QStringList years;
    years << QString::fromUtf8("2019") ;

    bool oky;
    QString year = QInputDialog::getItem(this,
                                           QString::fromUtf8("Выберите год"),
                                           QString::fromUtf8("Год:"), years, 1,
false, &oky);
    QStringList items;
    items << QString::fromUtf8("Январь") << QString::fromUtf8("Февраль")
        << QString::fromUtf8("Март") << QString::fromUtf8("Апрель")
        << QString::fromUtf8("Май") << QString::fromUtf8("Июнь")
        << QString::fromUtf8("Июль") << QString::fromUtf8("Август")
        << QString::fromUtf8("Сентябрь") << QString::fromUtf8("Октябрь")
        << QString::fromUtf8("Ноябрь") << QString::fromUtf8("Декабрь");

    bool ok;
    QString item = QInputDialog::getItem(this,
                                           QString::fromUtf8("Выберите месяц"),
                                           QString::fromUtf8("Месяц:"), items,
1, false, &ok);
    QString item2;
    if (item == "Январь"){item = "01"; item2 = "02";}else
        if (item == "Февраль"){item = "02";item2 = "03";}else
            if (item == "Март"){item = "03";item2 = "04";}else
                if (item == "Апрель"){item = "04";item2 = "05";}else
                    if (item == "Май"){item = "05";item2 = "06";}else
                        if (item == "Июнь"){item = "06";item2 = "07";}else
                            if (item == "Июль"){item = "07";item2 = "08";}else
                                if (item == "Август"){item = "08";item2 =
"09";}else
                                    if (item == "Сентябрь"){item = "09";item2 =
"10";}else
                                        if (item == "Октябрь"){item = "10";item2
= "11";}else
                                            if (item == "Ноябрь"){item =
"11";item2 = "12";}else
                                                if (item == "Декабрь"){item =
"12";}
    QString period = year + "-" + item + "-01";
    QString period2 = year + "-" + item2 + "-01";
    qry->prepare("select * from summary_of_orders "
                "where data >= :period1 "
                "and data < :period2");
    qry->bindValue(":period1", period );
    qry->bindValue(":period2", period2);
    qry->exec();
}

```

```

void MainWindow::on_pushButton_2_clicked() //сортировка
{
    if (ui->comboBox->currentText() == "По дате(воз)") {
        qry->exec("select * from summary_of_orders order by data asc");}
    if (ui->comboBox->currentText() == "По дате(уб)") {
        qry->exec("select * from summary_of_orders order by data desc");}
    if (ui->comboBox->currentText() == "По алфавиту") {
        qry->exec("select * from summary_of_orders order by fio asc");}
    if (ui->comboBox->currentText() == "По сумме(воз)") {
        qry->exec("select * from summary_of_orders order by sum asc");}
    if (ui->comboBox->currentText() == "По сумме(уб)") {
        qry->exec("select * from summary_of_orders order by sum desc");}
    if (ui->comboBox->currentText() == "По номеру заказа(воз)") {
        qry->exec("select * from summary_of_orders order by number asc");}
    if (ui->comboBox->currentText() == "По номеру заказа(уб)") {
        qry->exec("select * from summary_of_orders order by number desc");}
    if (ui->comboBox->currentText() == "По периоду") {
        the_choice_of_period();}

    conclusion();

}

void MainWindow::on_pushButton_clicked() //поиск
{

    if (ui->lineEdit->text() != "") {

        qry->prepare("select*from summary_of_orders where "
                    "number =:serch "
                    "or data =:serch "
                    "or fio =:serch "
                    "or address =:serch "
                    "or sum =:serch" );
        qry->bindValue(":serch", ui->lineEdit->text());
        qry->exec();
    }
    conclusion();
}

void MainWindow::on_pushButton_3_clicked() //удаление
{
    QItemSelectionModel *select = ui->tableView->selectionModel();
    if(!select->selectedRows().isEmpty()) //проверяем выделили ли строку (не
пустая ли)
    {
        int nr = select->selectedRows().first().row(); //вписываем номер строки
        if (QMessageBox::Yes == QMessageBox::question(this, "Удалить?",
                                                    "Уверены?",

QMessageBox::Yes|QMessageBox::No))
        {
            QString indid =
ui->tableView->model()->data(ui->tableView->model()->index(nr,0)).toString();
            modell->setQuery("DELETE FROM summary_of_orders WHERE id =" +indid);

```



```

    }

}

else
{
    //сообщение о том, что строка пуста
    QMessageBox::information(0, "Ошибка", "Строка не выделена.");
}

connectToSql();
}

void MainWindow::on_pushButton_4_clicked() //редактировать статус
{
    QTableWidgetItem *select = ui->tableView->selectionModel();
    if(!select->selectedRows().isEmpty()) //проверяем выделили ли строку (не
пустая ли)
    {
        int nr = select->selectedRows().first().row(); //вписываем номер строки
        {
            QString indid =
ui->tableView->model()->data(ui->tableView->model()->index(nr,0)).toString();
            QStringList items;
            items << QString::fromUtf8("Обработан") << QString::fromUtf8("Не
обработан")
                << QString::fromUtf8("Переговоры") <<
QString::fromUtf8("Отменен");
            bool oky;
            QString sts = QDialog::getItem(this,
                                           QString::fromUtf8("Выберите
статус"),
                                           QString::fromUtf8("Статус:"),
items, 1, false, &oky);
            if (sts == "Не обработан"){
                modell->setQuery("UPDATE summary_of_orders SET status = 'не
обработан' WHERE id ="+indid);}else
                if(sts == "Обработан"){modell->setQuery("UPDATE
summary_of_orders SET status = 'обработан' WHERE id ="+indid);}else
                if (sts == "Переговоры"){
                    modell->setQuery("UPDATE summary_of_orders SET status =
'переговоры' WHERE id ="+indid);}else
                    if (sts == "Отменен"){
                        modell->setQuery("UPDATE summary_of_orders SET
status = 'отменен' WHERE id ="+indid);}
                }
            }
        }
    }

else
{
    //сообщение о том, что строка пуста
    QMessageBox::information(0, "Ошибка", "Строка не выделена.");
}

connectToSql();
}

void MainWindow::on_pushButton_7_clicked() //сохранение
{
    model->submitAll();
}

void MainWindow::on_pushButton_8_clicked() //отмена
{
    model->revertAll();
}

```

```
void MainWindow::on_pushButton_5_clicked() //обновить
{
    QProcess *rem = new QProcess(this);
    rem->startDetached("imap.exe");
    connectToSql();
}
```

3. Вывод

В результате прохождения производственной практики на ООО «ГлобалТекстиль» значительно расширены профессиональные навыки.

За время прохождения практики я выполнял поставленные задачи, ознакомился документацией и с новыми методами и средствами разработки.

В ходе практики собраны данные и информация для объективной характеристики предприятия. Также проведен общий анализ работы организации, ее внешних контактов. Дана оценка структуре компании.

Получены умения работы в QT creator. Получены навыки работы с SQL.

4. Дневник практики

Дата	Краткое содержание работ	Отметка о выполнении работы	Подпись руководителя практики
15.04 – 17.04	Ознакомление с поставленной задачей. Решение в какой среде разрабатывать проект.		
20.04 – 27.04	Создание формы.		
28.04 – 7.05	Создание и подключение базы данных.		
8.05 – 9.05	Создание и настройка поиска по базе данных.		
11.05 – 14.05	Создание и настройка сортировки.		
15.05 – 17.05	Создание и настройка сортировки по периоду.		
19.05. – 20.06.	Создание кнопок удалить, статус, обновить. Реализация функций работы кнопок.		
26.06 – 27.06	Интеграция программы парсинга в основную программу. Настройка базы данных. Настройка формы.		

5. Список использованной литературы

1. <http://www.cyberforum.ru/>
2. <https://doc.qt.io/>
3. <https://ru.stackoverflow.com/questions/591367/Редактирование-qtableview>
4. http://it-tasks.blogspot.com/2013/08/qtableview_30.html
5. <http://forundex.ru/yap/Redaktirovanie-dannih-cherez-QSqlQueryModeltableView-40939>
6. <http://forum.vinograd.ru/topic-184964.html>
7. <https://habr.com/ru/post/51650/>

6. Приложения

Приложение 1

Дневник прохождения производственной практики Записи выполненных работ.

Дата	Краткое содержание работ	Отметка о выполнении работы	Подпись руководителя практики
27.05 – 28.05	Ознакомление с поставленной задачей. Решение в какой среде разрабатывать проект.		
20.04 – 27.04	Создание формы.		
28.04 – 7.05	Создание и подключение базы данных.		
8.05 – 9.05	Создание и настройка поиска по базе данных.		
11.05 – 14.05	Создание и настройка сортировки.		
15.05 – 17.05	Создание и настройка сортировки по периоду.		
19.05. – 20.06.	Создание кнопок удалить, статус, обновить. Реализация функций работы кнопок.		
26.06 – 27.06	Интеграция программы парсинга в основную программу. Настройка базы данных. Настройка формы.		

Указание о заполнении дневника практики

1. В колонке «Дата» указывается период выполнения работы, изучения материала.
2. В колонке «Содержание работ» записываются виды выполняемых студентом работ, наименование тем изучаемого материала.
3. Отметку о выполнении работ ставит руководитель практики от предприятия.

Отзыв-характеристика

**на студента, обучающегося на базовом уровне по специальности
09.02.03 «Программирование в компьютерных системах»**

(заполняется на рабочем месте непосредственным руководителем)

Ф.И.О. Миронов Василий Анатольевич

Студент обучается на 3 курсе в группе П1-16

Для заполнения отзыва ответьте, пожалуйста, на следующие вопросы:

(Выбранные ответы отметьте в таблице любым доступным способом. Ответов может быть несколько, но дополняющих друг друга)

1. Понимает ли студент-практикант сущность и социальную значимость своей будущей профессии?

Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво

2. Проявляет ли студент-практикант к своей профессии устойчивый интерес?

Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво

3. Способен ли студент-практикант организовать собственную деятельность?

Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво

4. Выбирает ли студент-практикант типовые методы и способы выполнения профессиональных задач?

Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво

5. Оценивает ли студент-практикант эффективность и качество решения различных задач?

Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво

6. Принимает ли студент-практикант решения в стандартных и нестандартных ситуациях?

Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво

7. Может ли студент-практикант нести ответственность за принятые решения?

Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво

8. Осуществляет ли студент-практикант поиск необходимой информации, необходимой для эффективного выполнения профессиональных задач?

Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво

9. Может ли студент-практикант применить необходимую информацию, для эффективного выполнения профессиональных задач?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
10. Повышает ли студент практикант своё личностное и профессиональное развитие?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
11. Владеет ли студент-практикант информационной культурой?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
12. Может ли анализировать студент-практикант информацию с использованием информационно-коммуникационных технологий?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
13. Может ли оценивать студент-практикант информацию с использованием информационно-коммуникационных технологий?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
14. Работал ли успешно студент-практикант в коллективе и в команде?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
15. Как эффективно студент-практикант общался с коллегами, руководством, потребителями?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
16. Берет ли студент-практикант на себя ответственность за работу членов команды (подчиненных) и за результат выполнения заданий?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
17. Может ли студент-практикант самостоятельно определять задачи профессионального и личностного развития?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
18. Может ли студент-практикант заниматься самообразованием?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
19. Может ли студент-практикант осознанно планировать повышение квалификации?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво
20. Ориентируется ли студент-практикант в условиях частой смены технологий в профессиональной деятельности?
Ответ: да/нет/интереса не проявляет/проявляет интерес настойчиво/инициативу не проявляет/проявляет инициативу настойчиво

№№ вопросов	Ответ: да	Ответ: нет	Ответ: интереса не проявляет	Ответ: проявляет интерес настойчиво	Ответ: инициативу не проявляет	Ответ: проявляет инициативу настойчиво
1	+					
2	+					
3	+					
4				+		
5				+		
6	+					
7	+					
8	+					
9	+					
10	+					
11	+					
12	+					
13	+					
14	+					
15						+
16	+					
17	+					
18	+					
19	+					
20	+					

Приложение 3

АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРАКТИКЕ

Студент Миронов Василий Анатольевич, обучающийся на 3 курсе по специальности 09.02.03 «Программирование в компьютерных

системах»

успешно прошел производственную практику в период с 27.04.2019 по 28.06.2019 в организации ООО «ГлобалТекстиль».

Виды и качество выполнения работ

Виды и объемы работ, выполненных обучающимся во время практики	Качество выполнения работ в соответствии с технологией и/или требованиями организации, в которой проходила практика
Знакомство с составом компании. Инструктаж. Изучение устава. 10Ч.	Ознакомился с документами.
Создание базы данных 70Ч.	Создал базу данных и подключил ее к программе.
Разработка программного интерфейса. 40Ч.	Разработал интерфейс программы.
Разработка программы и интеграция программы парсинга. 47Ч.	Успешно интегрировал и разработал программу.
Тестирование и отладка программного продукта. 60Ч.	Успешно провел тесты программы.
Составление отчета. 25Ч.	Составил отчет.

Характеристика учебной и профессиональной деятельности обучающегося во время производственной практики

По итогам прохождения практики студент заслуживает оценку -

Руководитель практики

Туманова Н.А.

инициалы, фамилия

28.06.2019

подпись, дата

Студент группы П1-16

код группы

Миронов В.А.

инициалы, фамилия

28.06.2019

дата

подпись

Королёв, 2019