**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Колледж экономики и информатики им. А.Н. Афанасьева**

Специальность 38.02.01 Экономика и бухгалтерский учет (по отраслям)

К защите допустить «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

Декан факультета \_\_\_\_\_\_\_\_\_\_\_С.Ю. Прохорова

подпись

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**(дипломная работа)**

**Тема**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Обучающийся:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

подпись инициалы, фамилия

**Руководитель:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

подпись инициалы, фамилия

**Рецензент:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

подпись инициалы, фамилия

**Нормоконтролер:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

подпись инициалы, фамилия

Ульяновск   
2020г.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Колледж экономики и информатики им. А.Н. Афанасьева**

Специальность 38.02.01 Экономика и бухгалтерский учет (по отраслям)

|  |  |
| --- | --- |
| Рассмотрено и одобрено на заседании  предметной (цикловой) комиссии экономических и управленческих дисциплин Протокол № \_\_\_ от «\_\_\_»\_\_\_\_\_\_\_\_\_\_2020 г. Председатель П(Ц)К \_\_\_\_\_\_\_\_\_\_Н.А. Хайбуллова | УТВЕРЖДАЮ: Зам. директора по УМР \_\_\_\_\_\_\_\_\_\_\_\_ И.П. Вершинина «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

**ЗАДАНИЕ**

**на дипломную работу обучающегося**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

1 Тема работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

утверждена приказом по университету \_\_\_\_\_\_\_ от “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

2 Срок сдачи обучающимся законченной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3 Исходные данные к работе \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4 Содержание дипломной работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Консультанты по работе, с указанием относящихся к ним разделов работы:

|  |  |  |  |
| --- | --- | --- | --- |
| Раздел | Консультант | Дата, подпись | |
| задание выдал | задание принял |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6 Дата выдачи задания\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Фамилия, имя, отчество

руководителя (полностью)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Занимаемая должность\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Задание принял к исполнению\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись обучающегося) **(**инициалы, фамилия)

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Колледж экономики и информатики им. А.Н. Афанасьева**

Специальность 38.02.01 Экономика и бухгалтерский учет (по отраслям)

**ОТЗЫВ**

**на дипломную работу**

Тема \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

1. Объем дипломной работы \_\_\_\_\_\_\_\_\_\_\_\_ страниц (листов), в т.ч.

а) количество страниц практической (расчетно-аналитической) части \_\_\_\_\_\_\_

б) количество страниц приложений \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Оценка содержания дипломной работы, ее положительные стороны и недостатки, выводы и предложения

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Мнение руководителя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель дипломной работы \_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

РЕЦЕНЗИЯ

На дипломную работу, выполненную обучающимся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема дипломной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Соответствие дипломной работы по объему и содержанию, выданному заданию \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Возможность использования дипломной работы или отдельных частей в практической деятельности или учебном процессе \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Достоинства дипломной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Недостатки дипломной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Мнение рецензента на основе анализа дипломной работы о степени подготовленности выпускника к работе по специальности \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дипломная работа обучающегося \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

заслуживает \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ оценку

(оценка по пятибалльной системе)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(место работы, должность рецензента)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_

С рецензией ознакомлен(а) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись обучающегося) **(**инициалы, фамилия)

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Колледж экономики и информатики им. А.Н. Афанасьева**

Специальность 38.02.01 Экономика и бухгалтерский учет (по отраслям)

Календарный график

выполнения дипломной работы

Сроки выполнения с\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ по \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ курс\_\_\_ группа №\_\_\_\_\_\_\_\_

Тема дипломной работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание работы | Кол-во часов консуль-таций | Норматив ный срок | Фактичес кий срок | Оценка этапа |
| 1 Выдача задания и проведение консультации по методике написания дипломной работы | 2 |  |  |  |
| 2 Согласование плана. Подбор литературы, нормативных документов | 2 |  |  |  |
| 3 Консультации по отдельным разделам дипломной работы:  Работа с теоретической частью  Работа с практической частью  Работа с заключением  Работа с материалом в электронной форме | 2 |  |  |  |
| 2 |  |  |
| 1 |  |  |
| 2 |  |  |
| 4 Сдача выполненной дипломной работы | - |  |  |  |
| 5 Написание отзыва на дипломную работу | 1 |  |  |  |
| 6 Нормоконтроль | - |  |  |  |
| 7 Предоставление дипломной работы на рецензирование | - |  |  |  |
| 8 Подготовка к защите | 2 |  |  |  |
| 9 Защита дипломной работы | - |  |  |  |
| Итого: | 14 |  |  |  |

Руководитель дипломной работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 8](#_Toc42010640)

[1. Актуальность 9](#_Toc42010641)

[2. Экономическая часть 10](#_Toc42010642)

[2.1 Расчет трудоемкости разработки программного продукта 10](#_Toc42010643)

[2.2 Расчет себестоимости и цены разработки программного продукта 11](#_Toc42010644)

[2.3 Методика расчета статей калькуляции 12](#_Toc42010645)

[3 Анализ требований 18](#_Toc42010646)

[3.1 Изучение предметной области 18](#_Toc42010647)

[4 Разработка программы 19](#_Toc42010648)

[4.1 Разработка базы данных 19](#_Toc42010649)

[4.2 Создание аутентификации и меню 21](#_Toc42010650)

[4.3 Создание формы для работы с таблицей “Склад” 24](#_Toc42010651)

[4.4 Создание формы для работы с таблицей “Работинки” 30](#_Toc42010652)

[4.5 Создание формы для работы с таблице “Заказ” 34](#_Toc42010653)

[4.6 Создание формы таблицы Истории 45](#_Toc42010654)

[ЗАКЛЮЧЕНИЕ 51](#_Toc42010655)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 52](#_Toc42010656)

# **ВВЕДЕНИЕ**

В 21 веке цифровые технологии окутали весь мир, почти в каждом доме присутствует персональный компьютер, а в большинстве компаний присутствует собственные сервера для обработки больших данных и хранения их в мировой сети. Для любого оборудования обязательно требуется программное обеспечение способное выполнять поставленные задачи и облегчая работу сотрудников. Каждая компания на момент своего развития начинает отказываться от бумажных носителей и переносит всё в электронный формат. Происходит это из-за многих факторов в числе которых: отсутствие возможности успевать следить за рабочими, контроль документации, утеря дорогостоящего товара и т.п. В последствии компании начинают нуждаться в вычислительной технике, которая могла бы содержать и контролировать информацию. В ту же стадию вошла компания ООО “Инструмент-Сервис”. Данной компании потребовалась программное обеспечение способное содержать в себе информацию о содержимом склада, информацию о работающих на предприятии людей, и возможности отслеживать выполнения работы людей, берущих материалы со склада.

Цель дипломной работы заключается в создании программного обеспечения по заказу для ООО “Инструмент-Сервис”, в выполняемые задачи которого входит контроль содержимого склада, содержание информации о рабочих и сохранение данных о выполненном заказе.

Задачами дипломной работы являются: изучение предметной области, ознакомление с возможными для создания поставленной задачи языков программирования, выбор наиболее подходящего, ознакомление с инструментарием, подбор наиболее подходящего варианта спроектированной программы, изучение возможностей баз данных и создание программного обеспечения.

1. **Актуальность**

Актуальность темы заключается в постоянной потере дорогостоящего товара, причиной стала невозможность следить за выполнением работы по причине высокого количества работников и большого количества заказов. Поэтому потребовалась программа способная следить за выполнением заказов рабочими и количеством приходящего и уходящего количества продукции на склад. Потеря товара происходит из-за халатности рабочих или же из-за преднамеренных действий, для того чтобы избежать этого требуется программа, хранящая информацию, это предоставляет возможность сравнить приносимую работниками отчётность и хранимую в программе информацию. Из-за высоких потерь и постоянных жалоб, данное программное обеспечение имеет высокую актуальность.

1. **Экономическая часть**

## **Расчет трудоемкости разработки программного продукта**

В рамках этого этапа проводится расчет затрат времени на выполнение работ по созданию программного продукта. Все работы, требующиеся для разработки программы, можно разделить на 2 группы:

*Составление необходимого перечня работ и ее кодирование*

В разработку программного продукта входит следующий перечень работ:

1. Постановка проблемы;
2. Описание предметной области;
3. Программирование и реализация Web-страниц, БД и т.п.;
4. Отладка приложения;
5. Документирование.

*Установка оценок времени продолжительности работ*

Для расчета ожидаемого времени выполнения работ возьмем значения времени выполнения при благоприятных условиях и при неблагоприятных условиях и произведем вычисления, взяв среднее арифметическое от оптимистичной и пессимистичной оценок.

На разработку программного продукта будет затрачено максимально 30 дней (с 1 по 30 мая) при следующих исходных данных:

Таблица 1 – Расчет трудоемкости внедрения программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование работы | Вероятностные оценки, дни | | |
| Tmin | Tmax | Tож |
| Постановка проблемы | 1 | 2 | 1,5 |
| Описание предметной области | 1 | 2 | 1,5 |
| Проектирование и дизайн программного продукта | 15 | 17 | 16 |
| Отладка Web-приложения, БД и т.п. | 2 | 3 | 2,5 |
| Оформление документов | 3 | 4 | 3,5 |
| Продолжение таблицы 1 |  |  |  |
| Внедрение программы | 1 | 2 | 1,5 |
| Общая продолжительность работы составляет: | 19 | 30 | 26,5 |

2

tmin+ tmax

tож = (4.1)

где Тож – ожидаемое время продолжительности работ

Тmin – оценка при наиболее благоприятных условиях

Tmax – оценка при наиболее неблагоприятных условиях

2

19+30

*tож* = = 26,5

Ожидаемое время выполнения работ по разработке программного продукта = 26,5 дней.

* 1. **Расчет себестоимости и цены разработки программного продукта**

Себестоимость разработки информационной системы - это, как правило, совокупность затрат на разработку программного продукта. Затраты на разработку программы подразделяются на следующие статьи расходов:

1. Материальные затраты;
2. Основная заработная плата;
3. Начисления на оплату труда;
4. Накладные расходы.
   1. **Методика расчета статей калькуляции**

Расчет материальных затрат

В статье «Материальные затраты» предусмотрены затраты на материалы, применяемые при использовании данного программного продукта на предприятии. Расчет стоимости материальных затрат произведен в таблице 2.

Таблица 2 – Расчет стоимости материальных затрат

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование материала | Количество  комплектов,  шт. | Цена одного  комплекта,  руб. | Транспортные  затраты,  руб. | Сумма  затрат, руб. |
| USB-флеш-накопитель | 1 | 400 | 0 | 400 |
| Бумага | 1 | 200 | 0 | 200 |
| Картридж для принтера | 1 | 2000 | 0 | 2000 |
| Ручка | 1 | 5 | 0 | 5 |
| Всего | - | - | - | 2605 |

*Расчет заработной платы программиста*

Расчет ЗП программиста производится в соответствии с трудоемкостью разработки программного продукта.

Данные для расчета:

1. Оклад – 20000 руб.;
2. Плановый фонд рабочего времени за месяц – 176 часов (22 дня);
3. Надбавка за интенсивность– 20%;
4. Стимулирующая надбавка – 50%;
5. Тарифная ставка.

Часовая тарифная ставка (Сч) определяется:

Фрв

Оклад

Сч = , (4.2)

где Фрв – плановый фонд рабочего времени за месяц, из расчета 22 рабочих дня по 8 часов.

Сч = = 113,64 руб. в час

176

20000

Основная заработная плата программиста за разработку программы составит:

ЗПосн = Сч ∙ Тож час (4.3)

ЗПосн = 113,64 ∙ (26,5 ∙ 8) = 113,64 ∙ 212 = 24091,68 руб.;

ЗПдоп = ЗПосн ∙ 0,7 (4.4)

ЗПдоп = 24091,68 ∙ 0,7 = 16864,18 руб.

Итого затраты на оплату труда:

ЗПобщ = ЗПосн  + ЗПдоп (4.5)

ЗПобщ = 24091,68+16864,18 = 40955,86 руб.

*Начисления на выплаты по оплате труда – 30,2%:*

ПФР – 22%, ФСС – 2,9%, ФФОМС – 5,1%, страховые взносы от

несчастных случаев на производстве – 0,2%

100

ЗПобщ∙30,2

Стр.взн. = , (4.6)

40955,86**.** 30,2

Стр. взн. = 100 = 12368,67 руб.

*Расчет накладных расходов*

В статью включены затраты, связанные с обслуживанием и организацией производства. В данном случае – это расходы на электроэнергию, потребляемую компьютером за время разработки программы и амортизационные отчисления.

Таблица 3 – Затраты на электроэнергию

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вид оборудования | Мощность, кВт | Стоимость, кВт/час | Время работы оборудования, Тож час | Сумма затрат, руб. |
| ПК | 0,7 | 3,77 | 212 | 559,47 |
| Итого | - | - | - | 559,47 |

В таблице 3 выполнены расчеты по затрате ресурсов на электроэнергию по формуле:

Сумма затрат = (М ∙ С) ∙ Т, (4.7)

где М - Мощность, кВт

С - Стоимость 1 кВт/час

Т - Время работы оборудования, Тож час

В таблице 4 рассчитана сумма амортизационных отчислений за период разработки программного продукта в днях.

Таблица 4 – Амортизационные отчисления

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вид оборудования | Первоначальная стоимость, руб. | Количество рабочих месяцев | Норма амортизации % | Сумма амортизационных  отчислений, руб. |
| ПК | 80000 | 1 | 12,5 | 833,33 |
| Итого | - | - | - | 833,33 |

Сумма амортизационных отчислений за период разработки, определяются по следующим формулам:

100

Сп∙ На

Аг = (4.8)

Тмес

∙

12

Аг

Амес = , (4.9)

где Сп – первоначальная стоимость оборудования, рублей;

На – годовая норма амортизации, %;

Тмес – трудоемкость разработки программы, месяцы;

12 – количество календарных месяцев в году;

Прочие накладные расходы – 20% от основной заработной платы:

100

24091,68∙20

4818,34 руб.

=

Сумма накладных расходов = затраты на электроэнергию + амортизационные отчисления + прочие накладные расходы.

Сумма накладных расходов = 559,47 + 833,33+ 4818,34 = 6211,14 руб.

*Расчет калькуляции затрат на разработку программного продукта*

Расчет калькуляции затрат наглядно представлен в таблице 5 «Калькуляция затрат на разработку программного продукта».

Таблица 5 - Калькуляция затрат на разработку программного продукта

|  |  |
| --- | --- |
| Статья затрат | Сумма затрат, руб. |
| Материальные затраты | 2605 |
| Общая сумма з/п | 40955,85 |
| Отчисления на страховые взносы | 12368,67 |
| Накладные расходы, в т.ч. амортизация | 6211,14 |
| Итого: | 62140,66 |

**Определение** цены программного продукта

Процент рентабельности (условно) – 25% ,

Прибыль рассчитывается по формуле:

100

Сполн ∙%Р

П = , (4.10)

где Сполн – себестоимость

Р - процент рентабельности

100

62140,66∙25

П = = 15535,16 руб.

Цена программного продукта равна сумме полной себестоимости и прибыли:

Ц = С*полн* + П (4.11)

Ц =62140,66 + 15535,16 = 77675,82 руб.

Цена программного продукта с НДС (ставка 20%):

НДС = (4.12)

100

Ц ∙20

100

(77675,82 ∙20)

НДС = = 15535,16

Цена = 77675,82 + 15535,16= 93210,98 руб,

Стоимость программного продукта составляет 93210,98 рублей.

Для создания программного обеспечения, для персонального компьютера необходимы затраты в размере 93210,98 рублей. В затраты входят стоимость: затраченной электроэнергии, оборудования, расходных материалов и заработной платы программиста. В общем на разработку программы ушло 30 дней каждый из которых включал в себя 8 часовой рабочей день. Большая часть времени была потрачена на создания программного продукта. Так же много времени заняло создание документации и изучении предметной области.

Эффективность данной разработки оказалось крайне высока. По причине потраченного времени. Расчёт общего времени выполнения работы должен был занять 3 месяца. Потрачено времени было 26,5 дней. Что подтверждает высокую эффективность работы программиста над программным продуктом и документацией.

Качество созданной программы на хорошем уровне. Оно определяется из двух факторов: отзыва заказчика и сравнение с конкурентами. После пользования программным продуктом заказчик отлично отозвался о созданном программном обеспечении, так же он отметил, что программа превзошла ожидания и оказалась более простой в использовании чем ожидалось. В сравнении с конкурентами данная программа выполняет большее количество задач чем её аналоги. То есть в свою очередь является более эффективной для выполнения поставленных целей.

1. **Анализ требований**

## **Изучение предметной области**

ООО “Инструмент-сервис” компания, занимающаяся поставкой деталей по все России. В данной компании работает свыше полусотни людей, каждый из которых занимается поставленной перед ним задачей имеет установленную зарплату и вид выполняемой работы. Компания создаёт более 90 видов продуктов каждый из который имеет собственный артикул, размеры, материалы, стадии разработки. Для того чтобы описать заказ требуется знать артикул детали, стадии выполнения, сырьё, вид сырья, фио рабочего, вид выполняемой им работы. До отправки на склад заказ (состоящий из определённого количества деталей) проходит множество стадий выполнения. В них входят: заготовочная работа, токарная работа, заточная работа предварительная, термообработка, шлифовальная работа, заточная работа окончательная. Максимально количество стадий при выполнении заказа 8, то есть все перечисленные. Каждый рабочий при выполнении своей работы заносит в специальный лист информацию о выполненной работе, по которому в свою очередь выдаётся заработная плата учитывая время выполнения и штрафы, наложенные во время работы.

# **Разработка программы**

## **Разработка базы данных**

База данных в данной программе будет состоять из нескольких таблиц. В их число входят: Склад, Рабочие, Заказ и История. Каждая таблица отвечает за объект предметной области.

Склад – отвечает за информацию о каждой детали входящих в производство компании. В данную таблицу входят такие поля как: артикул, название, сырьё, вид сырья, Количество требуемого сырья, количество на складе и количество стадий.

Рабочие – отвечает за информацию о рабочих и о том чем они занимаются. В данную таблицу входят такие поля как: табельный номер, фио, вид работы, номер телефона, зарплата, статус.

Заказ – в этой таблице будет происходить основное действие, она будет включать в себя информацию из двух таблиц. В неё входят такие поля как: номер заказа, артикул, название, сырьё, табельный номер, фио, вид работы, количество, дата.

История – данная таблица существует для того чтобы хранить информацию о проделанной работе, сколько было сделано заказов, кем и когда. В неё входят: номер, номер заказа, фио, артикул, название, количество, дата.

Две таблицы Склад и Рабочие выдают информацию для таблицы заказ, соответственно таблица Заказ содержит внешние ключи, а таблица История принимает информацию из таблицы Заказ.

FK – внешний ключ

PK – первичный ключ

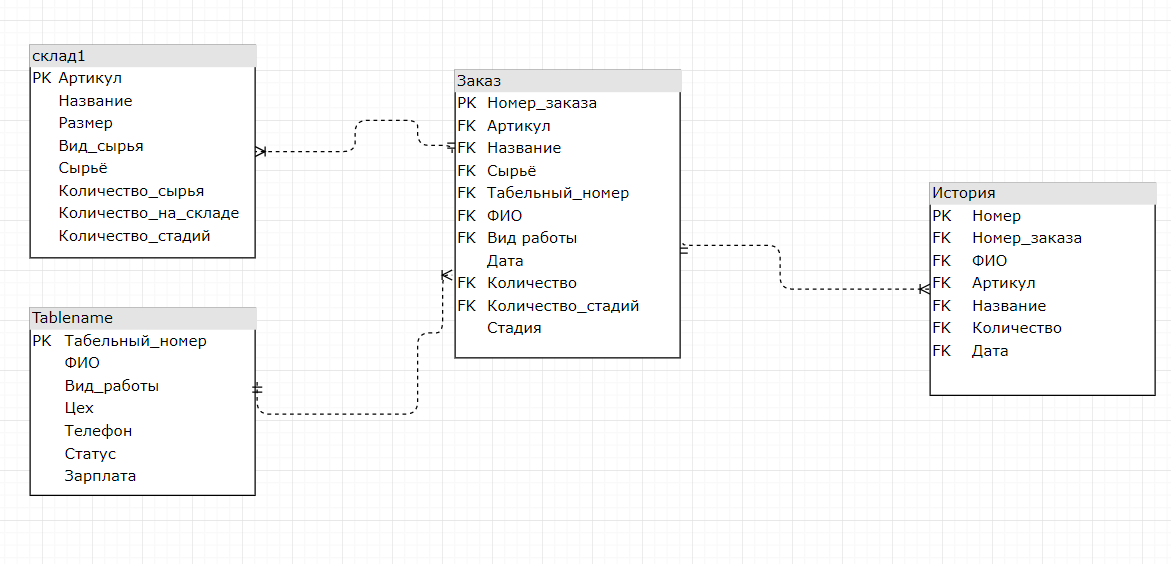


Рисунок 4.1.1 – ER диаграмма базы данных

## **Создание аутентификации и меню**

В самом начале требуется создать форму для аутентификации. Данная форма позволит защитить ценные данные от изменения или кражи, что позволит избежать ненужных неприятностей и огородит информацию о компании от внешнего проникновения.

Для создания было добавлено два текстовых поля textbox в который пользователь должен ввести логин и пароль после чего требуется нажать кнопку ввод, которая запустить проверку введённых данных, если проверка окажется удачной пользователь пройдёт дальше, если окажется не так, то на экране появится сообщение о неправильно введённом логине или пароле.

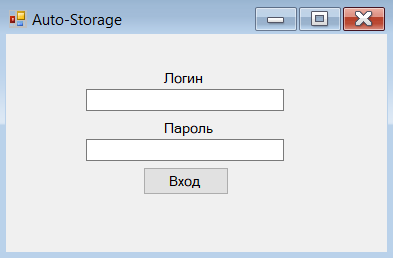


Рисунок 4.2.1 – Форма аутентификации

В первую очередь программа проверит существование файла базы данных, при не соответствии, она выдаст ошибку “Ошибка файла входа”.

Чтобы проверить логин и пароль программа должна свериться с уже имеющимися данными после чего выдать результат. Для содержания данных была выбрана база данных SQLite. Так же были варианты использования через внутренние переменные или XML, но данные варианты не подходят из-за того, что их легко прочесть и изменить. Файл базы данных тяжелее прочесть, что уменьшает вероятность изменить данные. Поэтому в данный файл “enter.sqlite” были внесены данные о пароле и логине.

Чтобы проверить данные программа должна открыть файл и свериться с ними, для работы с базой данных SQLite была подключена стороння библиотека “System.Data.SQLite.Core”, которая содержит в себе основные классы и функции для работы. Чтобы подключиться к базе данных, нужно прописать класс SQLiteconnection, где требуется указать путь к файлу. После подключения следует, создание переменной в которой будут записываться запросы для базы данных. В программе это переменная “cs”. После требуется открыть файл, в котором находится база данных и затем продолжить работу. Для того чтобы получить данные нужно написать запрос Select, который передаст данные для сравнения. Чтобы сравнить экземпляр пароля и логина пользователем, требуется использовать стандартный метод if. После последнего сравнения, при неудаче последует окно “Message.Box” в котором указано о не верном логине или пароле, если же всё верно программа призовёт следующую форму.



Рисунок 4.2.2 – код аутентификации

Меню программы основная форма от куда можно получить доступ к каждой таблице. Меню состоит из 4 кнопок: “Склад”, “Заказы”, “Рабочие”, “Создание отчёта”, каждая ведёт к форме для работы с таблицами.

Чтобы получить доступ к другой форме требуется создать экземпляр класса формы, то есть переменную, затем с помощью функции Show призвать её.



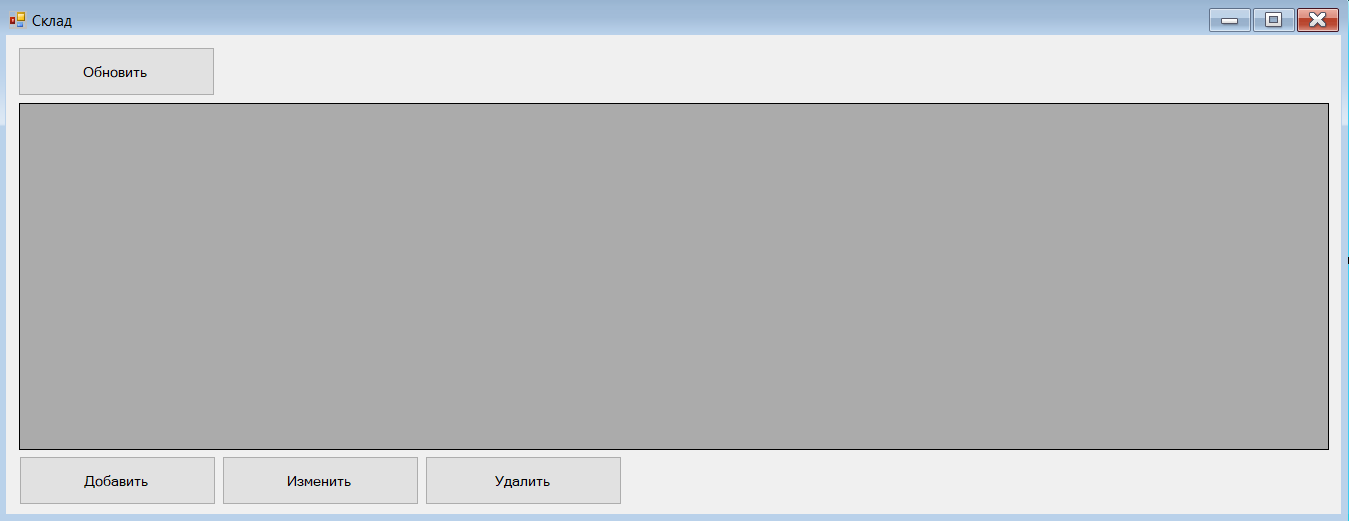
Рисунок 4.2.3 – форма меню



Рисунок 4.2.4– код формы меню

## **Создание формы для работы с таблицей “Склад”**

Первая кнопка приведёт к форме для работы с таблицей.

Рисунок 4.3.1 – форма для работы с таблицей “Склад”.

Форма состоит из 4 элементов: кнопки “Обновить”, “Добавить”, “Изменить”, “Удалить” и отображающий таблицу элемент datagridView.

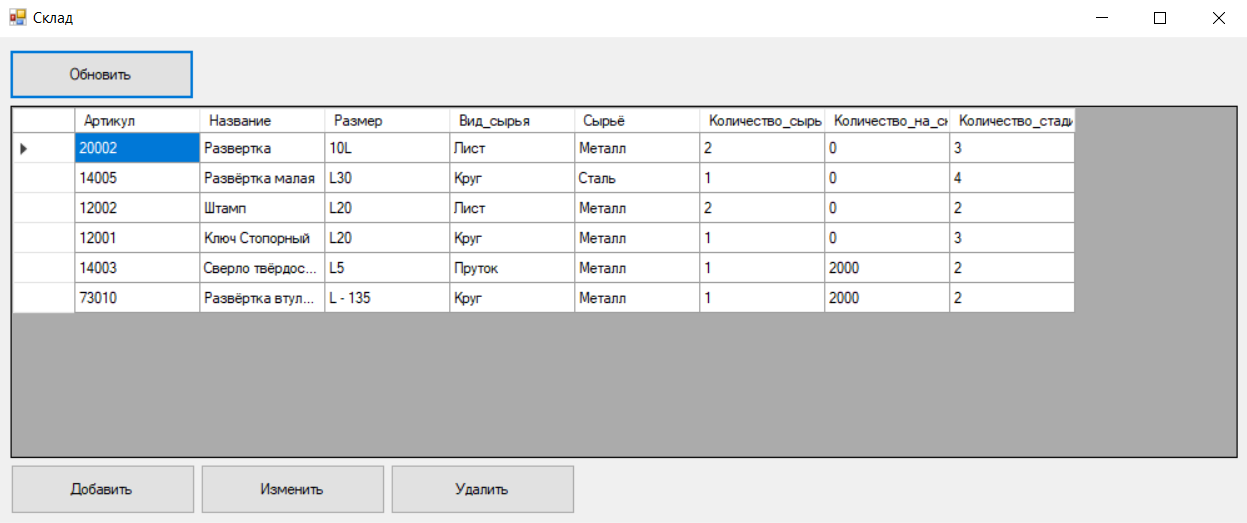


Рисунок 4.3.2 – заполненная форма таблицы “Склад”

Кнопка “Обновить” показывает таблицу Склад в элементе datagridView. После подключения к базе данных нужно описать запрос в переменной класса SQLitecommand. Чтобы вывести таблицу нужно составить запрос, Select \* from склад1. С помощью него возможно получить таблицу Склад. Но чтобы вывести её на экран нужно построить таблицу и отправить туда полученную таблицу Склад. Чтобы сделать это нужно воспользоваться классом SQLDataReader с помощью которого мы считываем таблицу и переводим её в понятный для языка формат. Затем требуется создать таблицу для этого служит встроенный класс DataTable в основной библиотеке System.Data. Переменная данного класса способна принять полученную таблицу и после этого появляется возможность загрузить его в элемент datagridView.

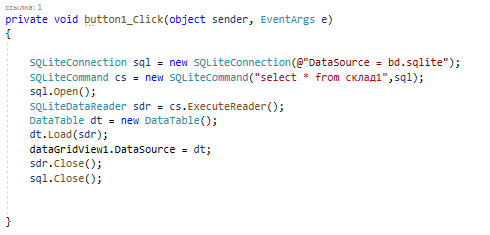


Рисунок 4.3.3 – код кнопки обновить

Кнопка “Добавить” вызывает форму добавления новых записей в таблицу.

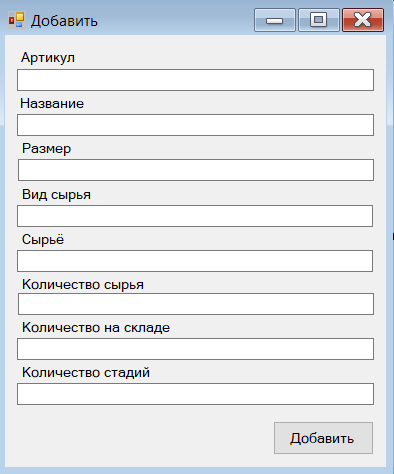


Рисунок 4.3.4 – форма добавления записи

Чтобы добавить запись нужно заполнить каждое поле. Все они имеют определённый тип данных, соответственно числовые и текстовые. После заполнения каждого поля данными, что является обязательным требованием, следует нажать кнопку “Добавить”, тогда начнётся процесс добавления новой записи в таблице. При заполнении первого текстового поля, если такой артикул уже существует, программы выдаст сообщение, предупреждающее о повторе существующего артикула.

Чтобы добавить новую запись в таблицу изначально нужно подключить к базе данных, затем следует создать переменную для хранения и исполнения команды здесь это “cs”. Для продолжения открываем базу данных через файл подключения sql. Следующим действием идёт проверка на существующий артикул. Она создана по причине того, что артикул является первичным ключом в базе данных и задаёт уникальное значение для каждого экземпляра базы данных, если уникальное значение будет нарушено библиотека SQLite выдаст исключение. С помощью запроса “Select Артикул from склад1” мы получаем список существующих артикулов в таблице. Сравнив введённые данные с предоставленными из базы данных, при некорректно введённом артикуле появится окно с предупреждением о уже существующем артикуле.

Затем следует запись в таблицу базы данных. Она осуществляется с помощью запроса. В каждый заранее определённый столбец определяется конкретное поле textbox. После чего команда исполняется с помощью функции “cs.ExecuteScalar”.

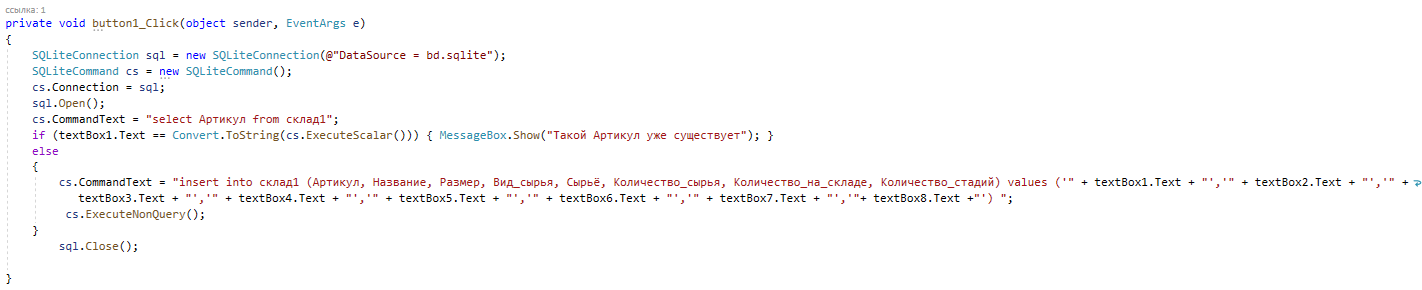


Рисунок 4.3.4 – код добавления записи в форме Склад

Следующей кнопкой в основной форме является “Изменить”. Она призывает форму, в которую заранее загружаются выбранные данные из datagridView.

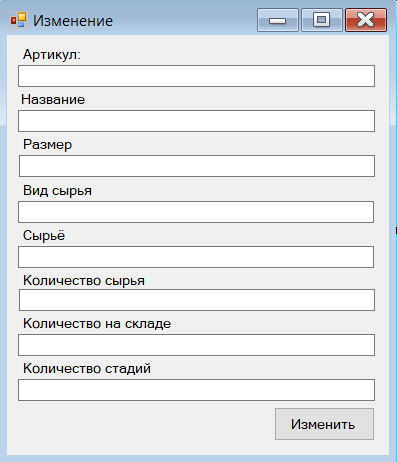


Рисунок 4.3.5 – форма кнопки изменить

Создав экземпляр класса каждому полю в следующей форме присваивается значение, это происходит через запрос Select, после чего каждое полученное поле присваивается своему полю в форме. Так же существует дополнительное поле, целью которого является содержание старого артикула. Это используется для того чтобы при изменении артикула, то есть первичного поля, не произошёл сбой и данные продолжали бы изменять на старом экземпляре таблицы.

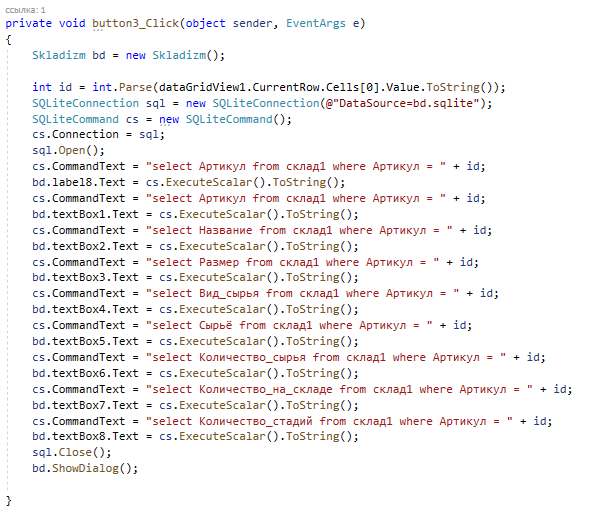


Рисунок 4.3.6 – код отправки данных в форму

После отправки формы открывается форма с уже вставленными данными, где пользователю нужно будет изменить их на новые. После изменения следует этап обновления данных в таблице.

Проверка на уже существующий артикул здесь не требуется, причиной тому служит то что при изменении каких-либо данных кроме артикула проверка не позволит изменить уже готовые данные ведь у экземпляра новых данных артикул будет тот же что и у старого.

Дальше следует изменение данных. В базе данных SQLite оно происходит с помощью запроса “Upgrade” который изменяет существующий экземпляр данных в таблице на новый. Данные полученные из каждого поля отправляются с помощью запроса в таблицу. Каждое поле соответствует своему в запросе, при этом сохраняя старый артикул с помощью условия.



Рисунок 4.3.7 – код кнопки изменить в форме Склад

Последней кнопкой в данной форме является удаление. С помощью неё можно удалить экземпляр записи в базе данных.

Она работает с помощью запроса Delete. Данный запрос удаляет данные опираясь на приведённый шаблон, в данном случае это артикул взяты из элемента datagridView.

В первую очередь пользователь выделяет строку которую требуется удалить. При нажатии на кнопку происходит подключение к файлу базы данных и создание переменной для команд. Затем открывается файл и с помощью запроса Delete, где в условие вставлен Артикул выделенной строки, взятый с помощью функции CurrentRow удаляется экземпляр таблицы.



Рисунок 4.3.8 – код кнопки удаления

## **Создание формы для работы с таблицей “Работинки”**

Следующей кнопкой в меню является Работники. Она призывает основную форму для работы с таблицей из базы данных.

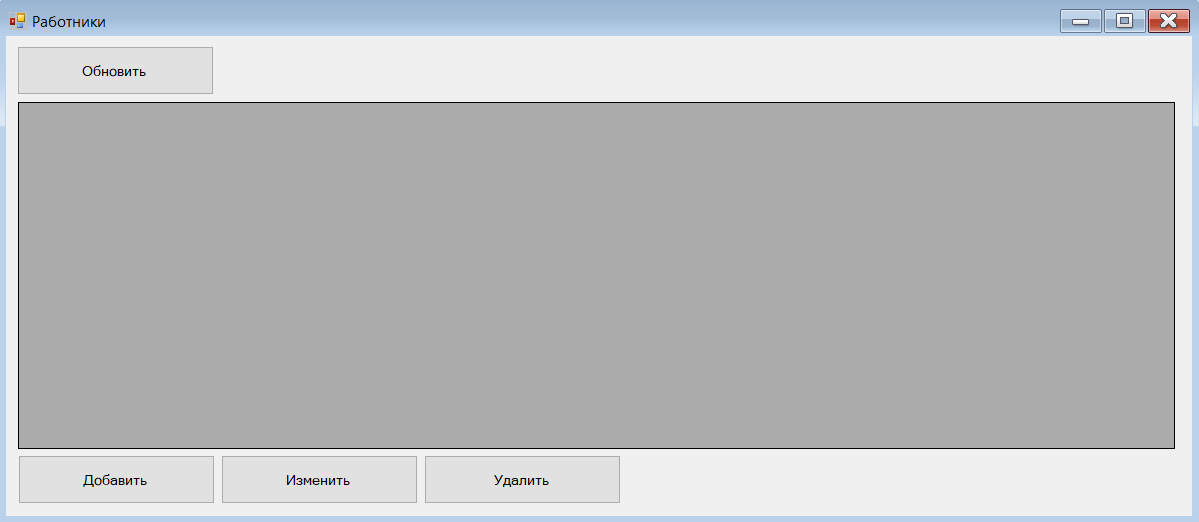


Рисунок 4.4.1 – форма таблицы “Рабочие”

Эта форма так же, как и форма для работы с таблицей склад содержит 4 кнопки и элемент datagridView.

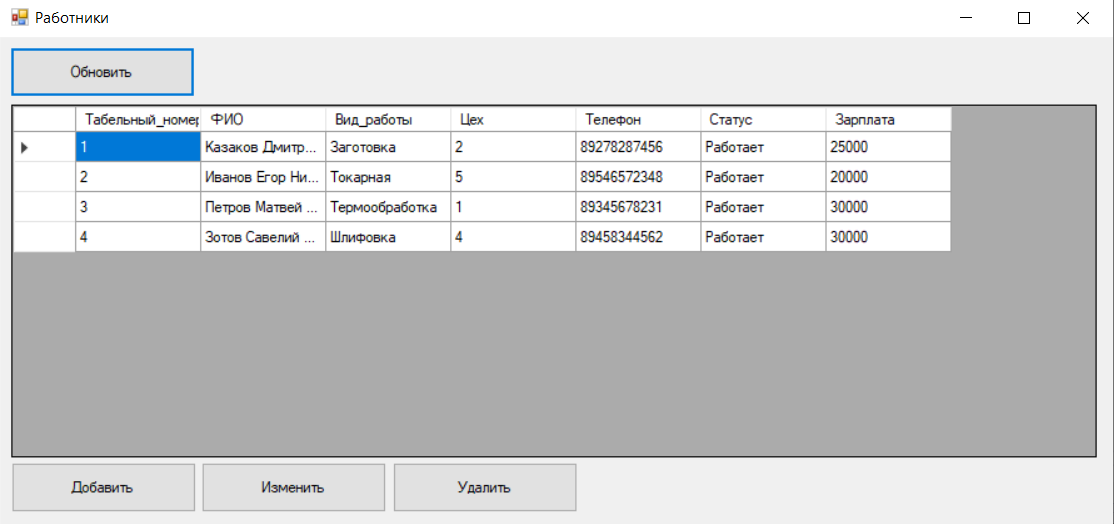


Рисунок 4.4.2 – заполненная форма таблицы “Рабочие”

При вызове кнопки добавить открывается изменённая форма кнопки из таблицы склад. Главным отличием от предыдущей формы является замена двух текстовых полей на поля combobox или селектора. С помощью данного селектора появляется возможность выбрать текстовые значение заранее занесённые в программный код, данное решение позволяет избежать ошибочных записей в базе данных, что увеличит правильность данных при создании экземпляров в таблицах.

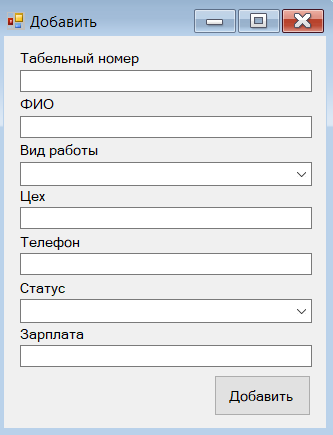


Рисунок 4.4.3 – кнопка добавить для таблицы Рабочие

Combobox это список в котором можно выбрать нужное значение. Чтобы использовать данный элемент требуется либо подключить его через сервер к базе данных, либо занести данные заранее. Данные требуемые в данной таблице будут занесены заранее. Этими данными будут являться вид работы и статус рабочего.

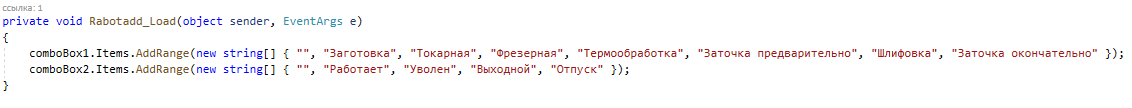


Рисунок 4.4.4 – код добавление экземпляров в combobox

Первый включается в себя разновидности работы, на предприятии которые описаны в изучении предметной области.

Второй описывает статус рабочего. Он включает в себя: работает, выходной, отпуск, уволен.

Чтобы добавить новые данные, после подключения к файлу базы данных и создания переменной для запросов, нужно сначала проверить на существование табельный номер нового экземпляра. Соответственно при несовпадении появляется окно с предупреждением, что такой табельный номер уже существует. Если же всё в порядке программа продолжает действие. Следующим будет добавление в базу данных с помощью insert. В данном случае добавление будет изменено в местах где нужно брать данные из combobox. Для этого существует функция selectedItem. Она позволяет взять выбранной пользователем экземпляр списка.

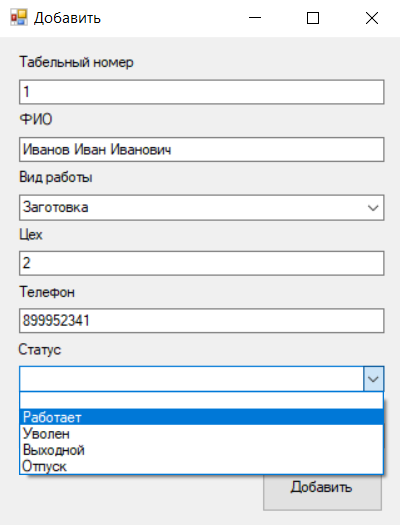


Рисунок 4.4.5 – пример работы combobox

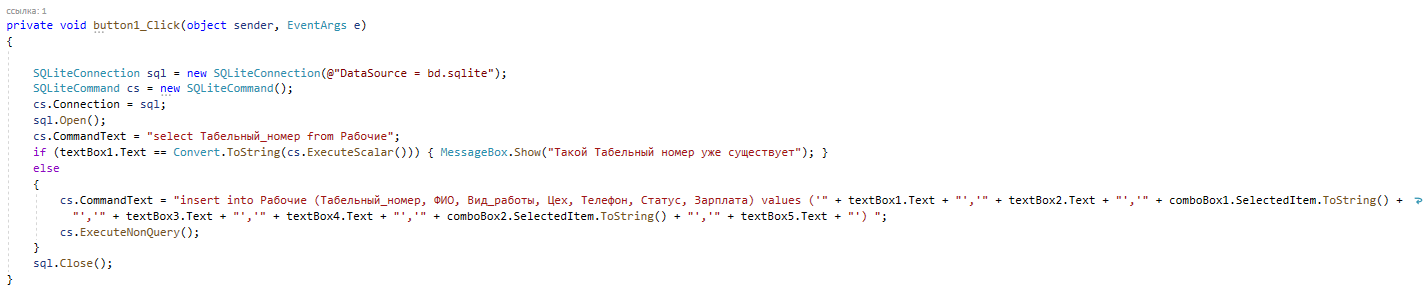


Рисунок 4.4.6 – код формы добавления таблицы Рабочие

Следующая кнопка в форме — это изменение. Принцип её действия схож с кнопкой изменения в форме Склада. Главное отличие заключается в изменённых элементах. Изначально чтобы загрузить туда выбранный до этого пользователем экземпляра нужно загрузить туда все возможные экземпляры, для этого нужно использовать функцию AddRange в совместительстве с созданием массива строк new string[]. После перечисления всех данных, дальше следует добавление в каждое поле экземпляр из выделенной области пользователем. Чтобы выбрать подходящий пункт в combobox используется SelectedItem. Когда всё будет выполнено откроется заполненная форма изменения. Здесь пользователю так же предстоит изменить текст в полях. Затем нажать на кнопку и с помощью запроса данные будут обновлены в таблице.



Рисунок 4.4.7 – код заполнения формы изменения для таблицы Рабочие



Рисунок 4.4.8 – код изменения данных для таблицы Рабочие

Кнопка удаления для данной таблицы идентична той же в таблице Склад.

## **Создание формы для работы с таблице “Заказ”**

Следующей формой будет таблица Заказы. Чтобы вызвать её нужно нажать на кнопку в меню. После вызова можно увидеть ранее использованную форму. Эта форма подходит для таблицы Заказов.

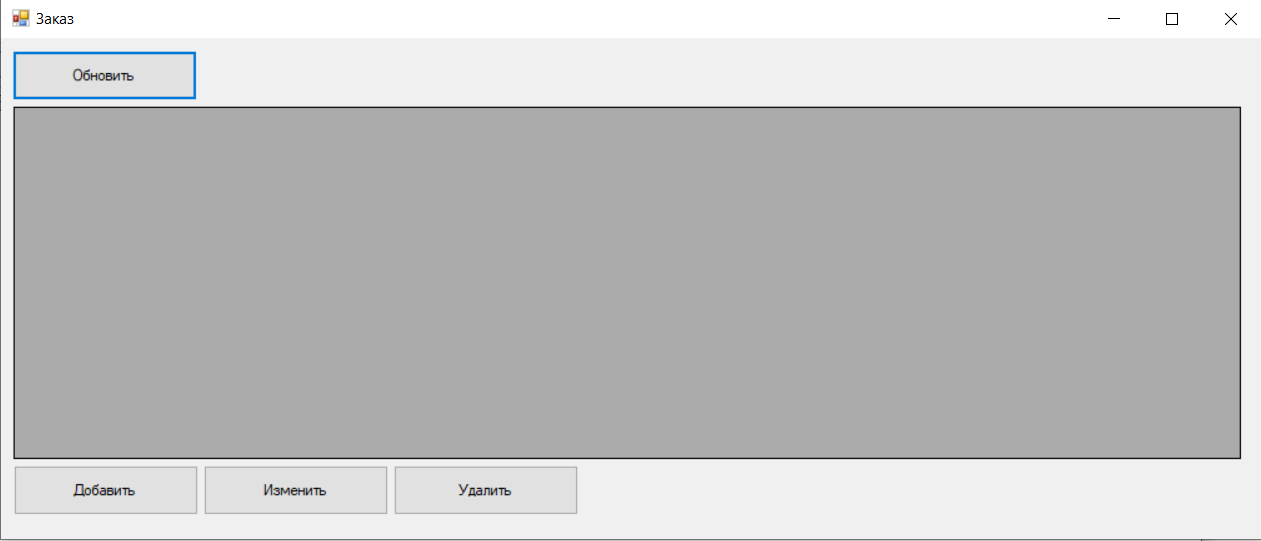


Рисунок 4.5.1 – форма таблицы “Заказ”

При обновлении таблицы происходит выборка данных из базы данных и предаётся в форму datagridView. После чего пользователь может приступить к работе.

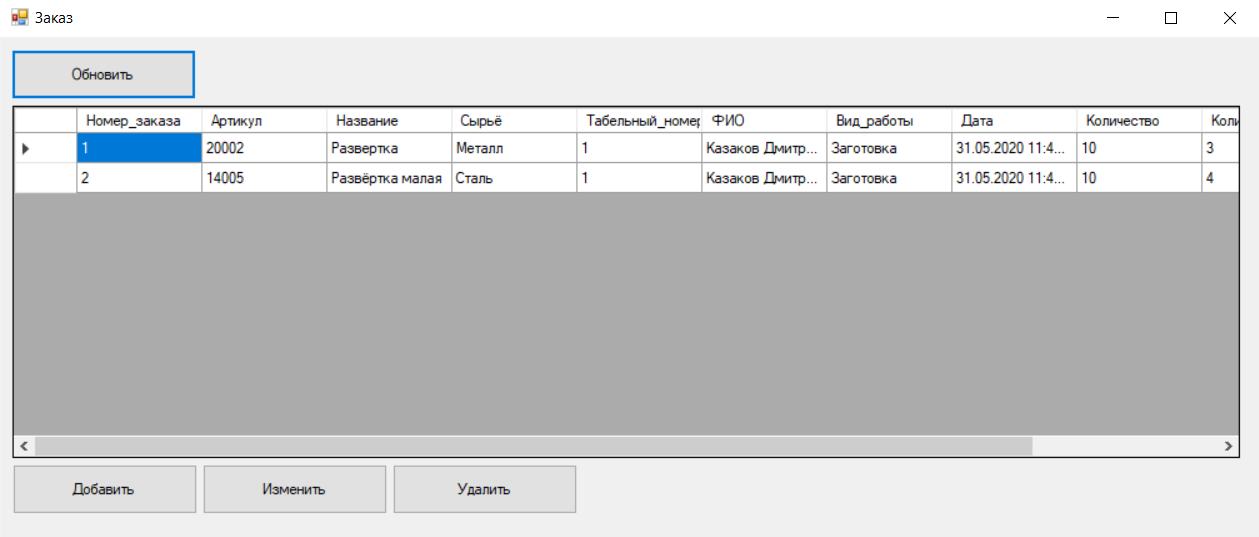


Рисунок 4.5.2 – заполненная форма таблицы “Заказ”

Главным отличием данной таблицы являются формы кнопок добавления и изменения. Форма кнопки добавления включает для себя 4 элемента для заполнения, остальные же элементы заполняются автоматический.

Автоматический заполняются такие элементы как label. Это решение обусловлено возможностью уменьшения шанса сделать ошибку при заполнении таблицы. По это при выборе элемента во 2 и 3 поле, некоторые данные заполняются автоматический. Данные берутся исходя из выбранных вариантов в поле. После отправляется запрос в соответствующие таблицы, возвращаемым значением служат данные соответствующие предоставленным первичным ключам.

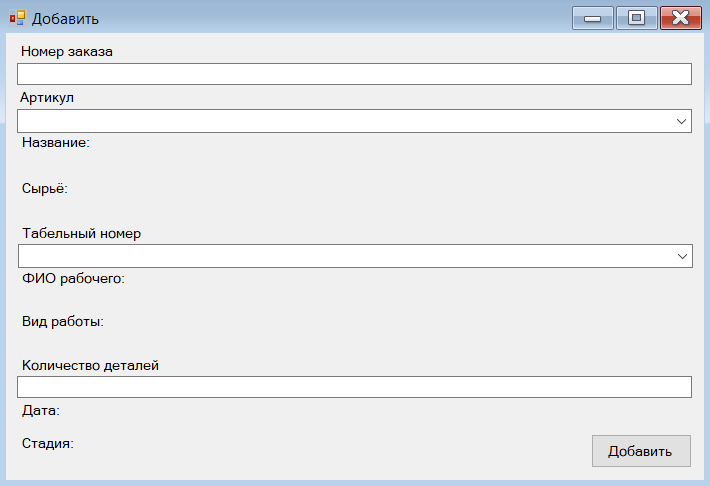


Рисунок 4.5.3 – форма добавления записи в таблицу Заказ

При выборе Артикула поля Название и Сырьё заполняются автоматический, и при выборе Табельного номера поля ФИО Рабочего и Вид работы заполняются автоматический.

Оба поля включают себя набор значений из других таблиц. Для добавления в элемент combobox (Артикул или Табельный номер) данных из другой таблицы, было выбрано событие формы “Добавления” Load которое выполняет действия при открытии формы. Главной задачей данного события будет заключатся в выборке данных из таблиц и отправку их в элементы combobox.

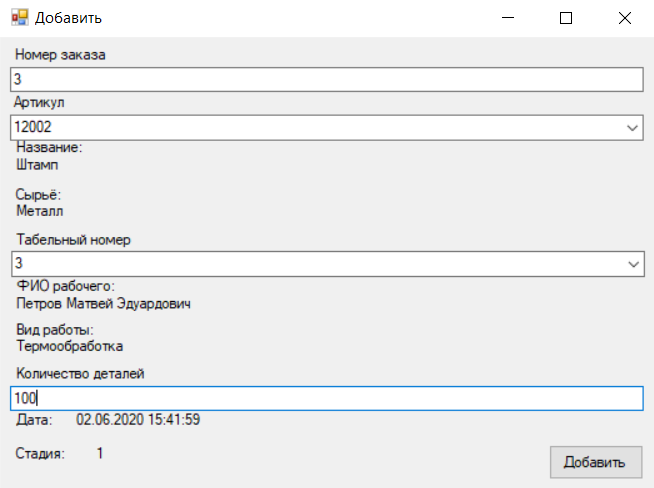


Рисунок 4.5.4 – пример заполнения формы добавления записи

После стандартного подключения к базе данных требуется написать запрос на все артикулы из таблицы склад и все табельные номера из таблицы рабочие. После получения данных, следует аккуратно конвертировать данные. Чтобы сделать это результаты запросов записываем в переменные типа Object, после записи требуется использовать цикл, в котором следует открыть файл для чтения m1.Read() и добавить с помощью функции Add по порядку все полученные артикулы, для второго запроса действие повторяется. В переменную m2 записывается результат, файл открывают для чтения и в цикле каждый полученный экземпляр добавляется в элемент combobox.

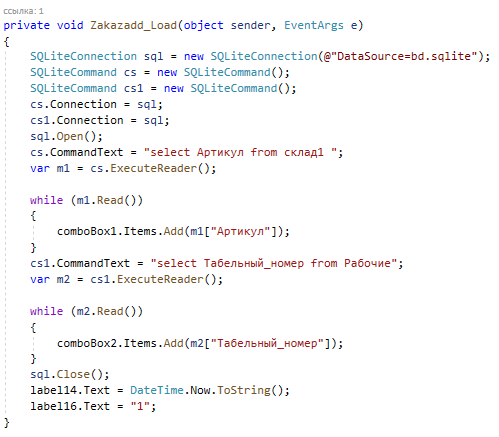


Рисунок 4.5.6 – код выполняемый при загрузке формы добавления

После того как данные будут загружены в требуемые элементы от пользователя требуется заполнить поля.

При изменении полей селекторов (combobox) несколько полей будут изменять значение автоматический. Чтобы реализовать данную функцию потребовалось добавить событие селектора comboBox1\_SelectedIndexChanged смыслом которого являет, что при изменении выполняется ряд функций, описанных внутри. Задача функций заключается в получении записей из таблиц с помощью запросов и передача их в соответствующие элемент в форме. В первую очередь следует проверка на то не оказался ли пустым при изменении значения селектор, если он оказался без значения, то ничего не происходит, если же значение есть, создаётся запрос select с помощью которого мы получаем данные из таблиц и передаём их в два элемента label4 и label6, где первый принимает значение названия детали, а второй получает значения сырья детали. Для второго селектора действия идентичны.

Заполнение элементов при изменении combobox1:

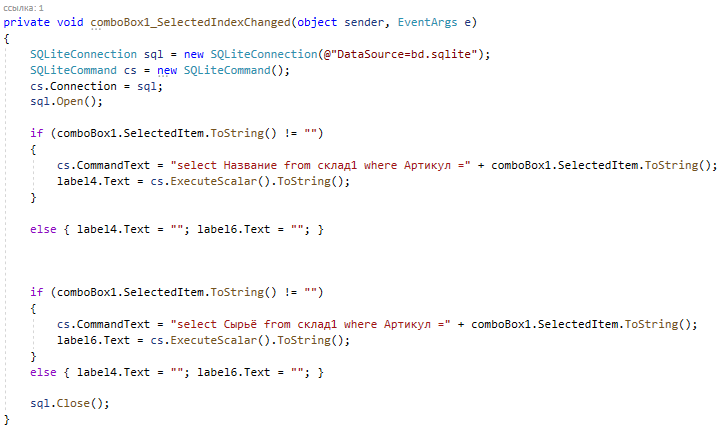


Рисунок 4.5.7 – заполнение элементов label (Название и Сырьё)

Заполнение элементов при изменении combobox2:



Рисунок 4.5.6 - заполнение элементов label (ФИО и Вид работы)

После того как текстовые поля будут заполнены, а в селекторах выбраны значения, будет нажата кнопка добавить. После её нажатия программа начнёт подключение к файлу базы данных и создаст переменную для выполнения запросов.

Первым запросом будет получение данных о количестве стадий. Так как в форме не было создано элемента для автоматического заполнения при изменении селектора за не надобностью, эти данные отправятся в объявленную переменную.

Затем следует проверка на существование первичного ключа, смысл которой заключается в предотвращении добавлении уже существующего значения. Происходит это с помощью функции ветвления if. В зависимости от значения программа выдаст окно о существующем Номере заказа или же продолжит свою работу.

Последним действием после успешно пройденной проверки является добавление новой записи в базу данных. Исполняется это с помощью запроса insert. В нём были описаны таблицы, в которые будет добавлена информация, затем по порядку добавлены названия столбцов и в таком же порядке будут добавлены имена элементов. Из текстовых полей textbox будут браться текстовые значения, введённые пользователем, так же будут браться значения и из полей label, а значения селекторов (combobox) берутся с помощью функции выбранного значения selectedItem. Затем всё это выполняется с помощью функции cs.ExecuteNonQuery(), смысл которой заключается в простом выполнении запроса без возвращаемого значения. Далее следует закрытие файла базы данных.

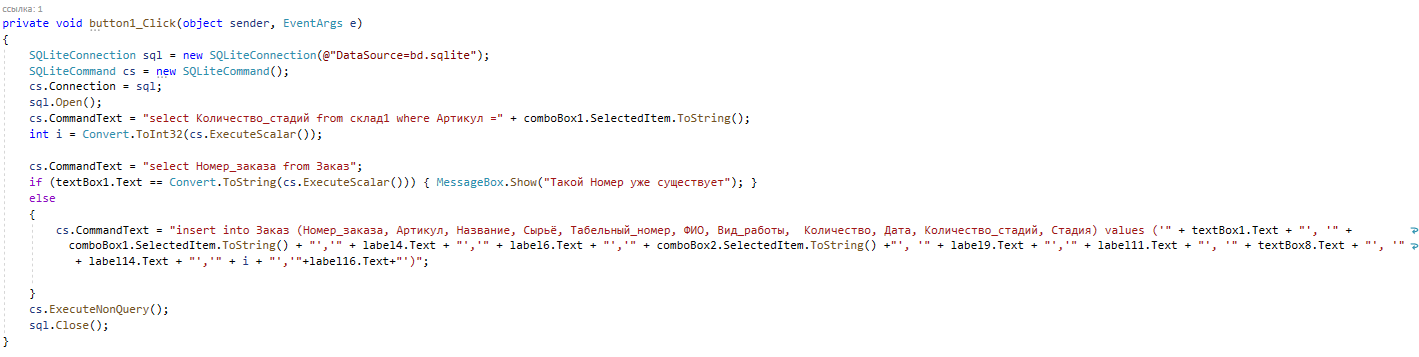


Рисунок 4.5.7 – код добавления в таблицу Заказ

Следующей кнопкой будет изменение записей в таблице Заказ. Её форма так же изменена как форма кнопки “Добавить”. Чтобы пользователь смог изменить данные их преждевременно нужно загрузить во все текстовые формы. При нажатии на кнопку “Изменить” в форме заказа, создаётся экземпляр класса формы изменения и во все его элементы добавляется выбранное пользователем значение. Чтобы сделать это используется многократно запрос select или же выборка. Для особых элементов, а именно combobox создаётся особенное добавление значение. При котором сначала всё записывается в переменную типа var (Object), после передаётся в сам селектор. Выполнив все запросы, программа открывает саму форму с уже заполненными полями, где пользователь может поменять некоторый из них на усмотрение.

Первая часть кода, в которой описано открытие базы данных, получения данных о Артикулах и Табельных номерах и заполнение ими селекторы.

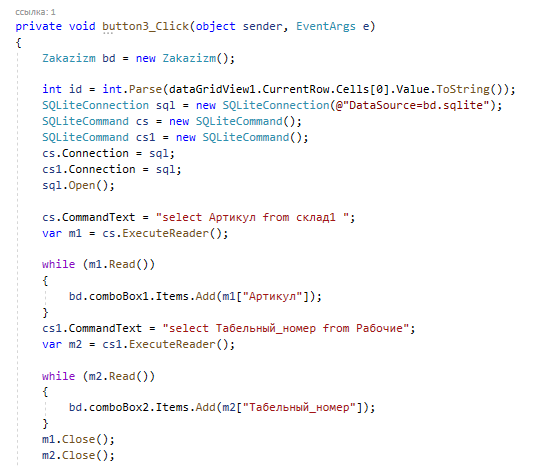


Рисунок 4.5.8 – код добавления элементов в форму для изменения (часть 1)

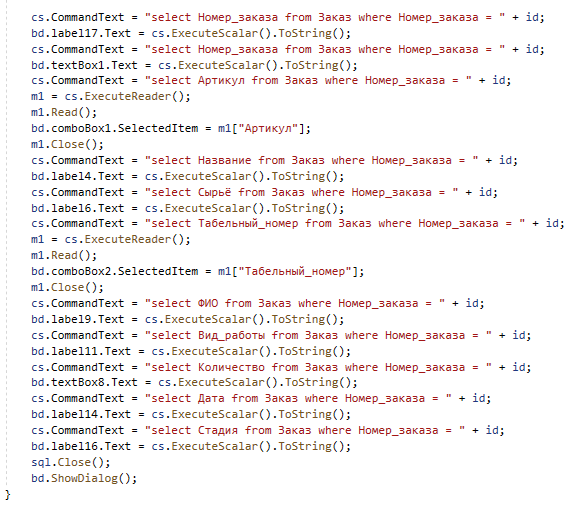


Рисунок 4.5.9 - код добавления элементов в форму для изменения (часть 2)

После ряда выполненных задач открывается уже заполненная форма.

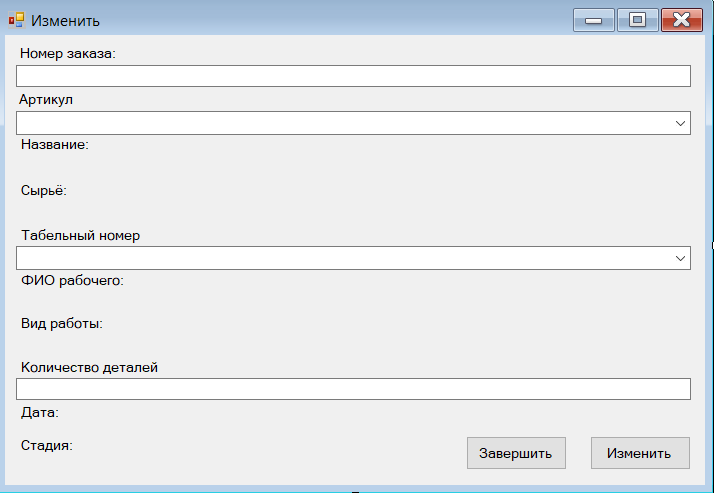


Рисунок 4.5.10– форма изменения для таблицы “Заказ”

После открытия формы пользователь может изменить значения по своему усмотрению. При изменения данных в селекторах, происходит тоже самое что и в форме кнопки “Добавить”, то есть каждый относящийся к нему label изменяется при совершении выбора в селекторе. После этого при нажатии на кнопку изменить произойдёт обновление данных в таблице. Происходит это с помощью запросов Upgrade каждый из которых обновляет определённое поле в таблице.

Так же в таблице присутствует дополнительный элемент хранящий в себе первичный ключ до изменения, что позволяет продолжать работу даже при изменении первичного ключа.



Рисунок 4.5.11 – код обновления записи в таблице Заказ

В данной форме присутствует кнопка “Завершить”. Это кнопка была создана специально для обновления стадии заказа. Дело в том, что каждый заказ имеет стадии и во время выполнения заказа они меняются. Чтобы реализовать данную функцию была добавлена эта кнопка при нажатии которой очищаются элементы combobox, а текстовое поле label отвечающее за текущую стадию увеличивается пока не достигнет своего предела. Здесь как раз требуется тот запрос, исполняемый в самом начале при добавлении новый записи. Его смысл в том, чтобы достать максимальное количество стадий. После получения этих данных программа будет продолжать обновлять стадию и записи о заказе пока не достигнет максимального значения. После достижения этой цифры заказ автоматический удаляется. Каждый раз при нажатии кнопки завершения стадии, экземпляр отправляется в созданную заранее таблицу История. По средству добавления нового запроса. Это используется для введения учёта выполненной работы. Поэтому пользователь может всегда посмотреть какой именно работник сколько выполнил работы. Так же для удобства при добавлении записи было добавлено время, которое бралось с компьютера и записывалось в заранее определённое поле, что позволит пользователю знать в какое время какой заказ был взят.



Рисунок 4.5.12 – код кнопки завершения

Завершающей кнопкой после изменения является кнопка удаления. Из её названия понятно, что данная кнопка служит для удаления записей в базе данных. Выбранная пользователем запись удаляется из базы данных. При этом элемент datagridView не обновляется и от пользователя требуется нажать кнопку “Обновить” чтобы получить более позднюю версию таблицы из базы данных.

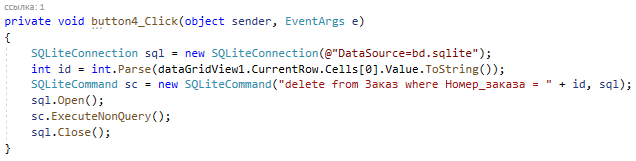


Рисунок 4.5.13 – код кнопки удаления в таблице Заказ

## **Создание формы таблицы Истории**

Для создания формы таблицы использовалось стандартная форма с внесёнными изменениями. Суть данной формы заключается в предоставление таблицы Истории чтобы пользователь мог ознакомиться с записями о выполненной работе. Эта форма позволяет выполнять основную функцию данной программы, а именно контроль за рабочими. Как уже стало понятно выше это будет происходить с помощью таблицы истории. После того как каждый рабочий принесёт документацию о выполненной работе директор компании сможет сгенерировать отчёт о проделанной работе и увидеть является ли принесённый документ не правдивым или же всё сделано правильно. Данная таблица несёт большую важность для выполнения основной функции. Так же чтобы можно было увидеть всю картину целиком была добавлена функция создания отчёта.

Для начала в форму не были добавлены кнопки добавить и изменить. Причиной этому стало мошенничество со стороны работников с базой данных. Во избежание конфликтов было решено не добавлять данные функции. Присутствует кнопка Обновление, она выполняет роль, как и в других формах, а именно отображает последнюю версию таблицы История базы данных.

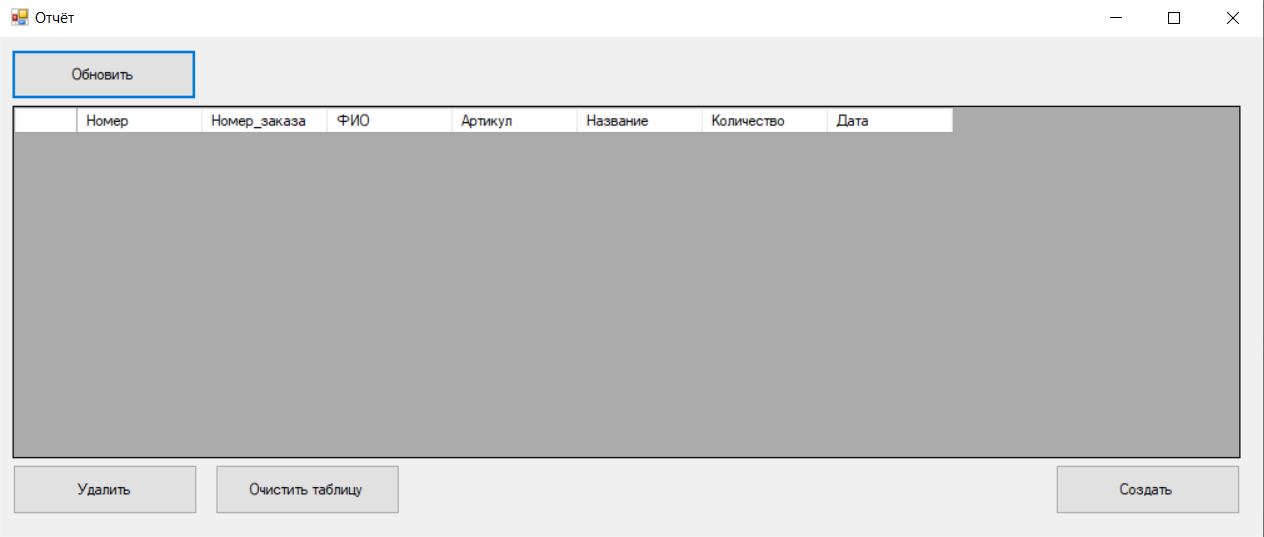


Рисунок 4.6.1 – отображение таблицы История

Так же в форме присутствует кнопка “Очистить таблицу”. Она очищает таблицу, то есть удаляет все экземпляры таблицы тем самым опустошая её. Эта функция создана для своевременной очистки таблицы. То есть каждый раз при генерировании отчёта пользователь может очистить таблицу и создать новый период истории.

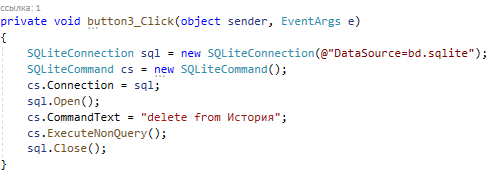


Рисунок 4.6.2 – очистка таблицы История

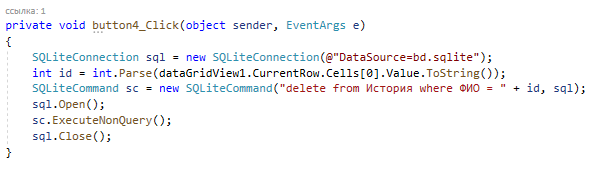


Рисунок 4.6.3 – удаление записи таблицы История

Создание отчёта происходит с помощью сторонней библиотеки itextsharp. Данная библиотека предоставляет набор классов и функций для работы с файлами. С помощью неё можно создавать, удалять, читать и записывать файлы любого формата.

В первую очередь нужно скачать и подключить библиотеку. Скачать её можно со стороннего ресурса. Для подключения требуется поместить файлы в корень программы после чего в ссылках указать путь к программе. После этого данной библиотекой можно пользоваться. К сожалению данный ресурс не имеет собственного набора шрифтов. Но в основном на ресурсах, которые предоставляют к скачиванию библиотеку, можно так же и скачать подходящий шрифт. При добавлении текста в созданный файл обязательно требуется файл содержащий шрифт.

После скачивания и подключения файлов библиотек к программе можно начинать работу. В первую очередь требуется создать сам файл. Для начала с помощью типа var был создан экземпляр класса документ “var doc”. Инструментарий позволяет читать и записывать файлы, но не файлы формата PDF. Затем создаётся экземпляр, содержащий в себе файл. Для этого нужно присвоить файлу класс библиотеки itextsharp “PdfWriter”. Данный класс позволяет считывать и создавать документы, в его условии нужно написать название документа в кавычках и описать условие, в этом случае это будет FileMode.Create, это означает что файл будет создан, но, если он уже существует в система он будет перезаписан на новый.

После этого требуется создать переменную класса, отвечающую за шрифт и цвет шрифта, а также его размеры. Она потребуется для добавления текста в файл.

Когда все переменные объявлены, можно начинать работу с файлом. Для этого его нужно открыть, это нужно сделать с помощью функции doc.Open(). После открытия файла можно добавлять туда данные.

Чтобы перенести таблицу из базы данных нужно создать переменную класса PdfTable. Этот класс позволяет создавать таблицы и заполнять их информацией.

Чтобы писать текст нужно воспользоваться классом Pdfcell и описать текст в свойствах, так же нужно вставить переменную содержащую шрифт.

PdfPCell cell = new PdfPCell(new Phrase("Отчёт: проделанная работа", font));

Затем добавляем в переменную cell таблицу datagridView и задаём значения ширины границ.

Следующим шагом будет добавление таблицы через цикл который будет проходить через всю таблицу.

table.AddCell(cell);

for (int j = 0; j < dataGridView1.Rows.Count; j++)

{

for (int k = 0; k < dataGridView1.Columns.Count; k++)

{

if (dataGridView1.Rows[j].Cells[k].Value != null)

table.AddCell(new Phrase(dataGridView1.Rows[j].Cells[k].Value.ToString(), font));

После следует добавить таблицу из переменной в файл, закрыть его и на этом работа с отчётом будет закончена.

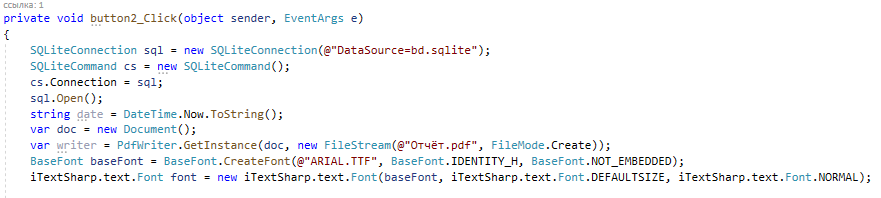


Рисунок 4.6.4 – код создания отчёта (часть 1)

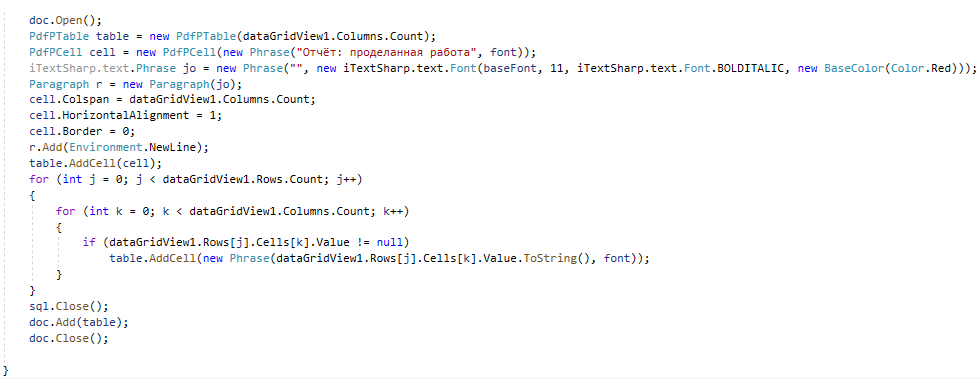


Рисунок 4.6.5 – код создания отчёта (часть 2)

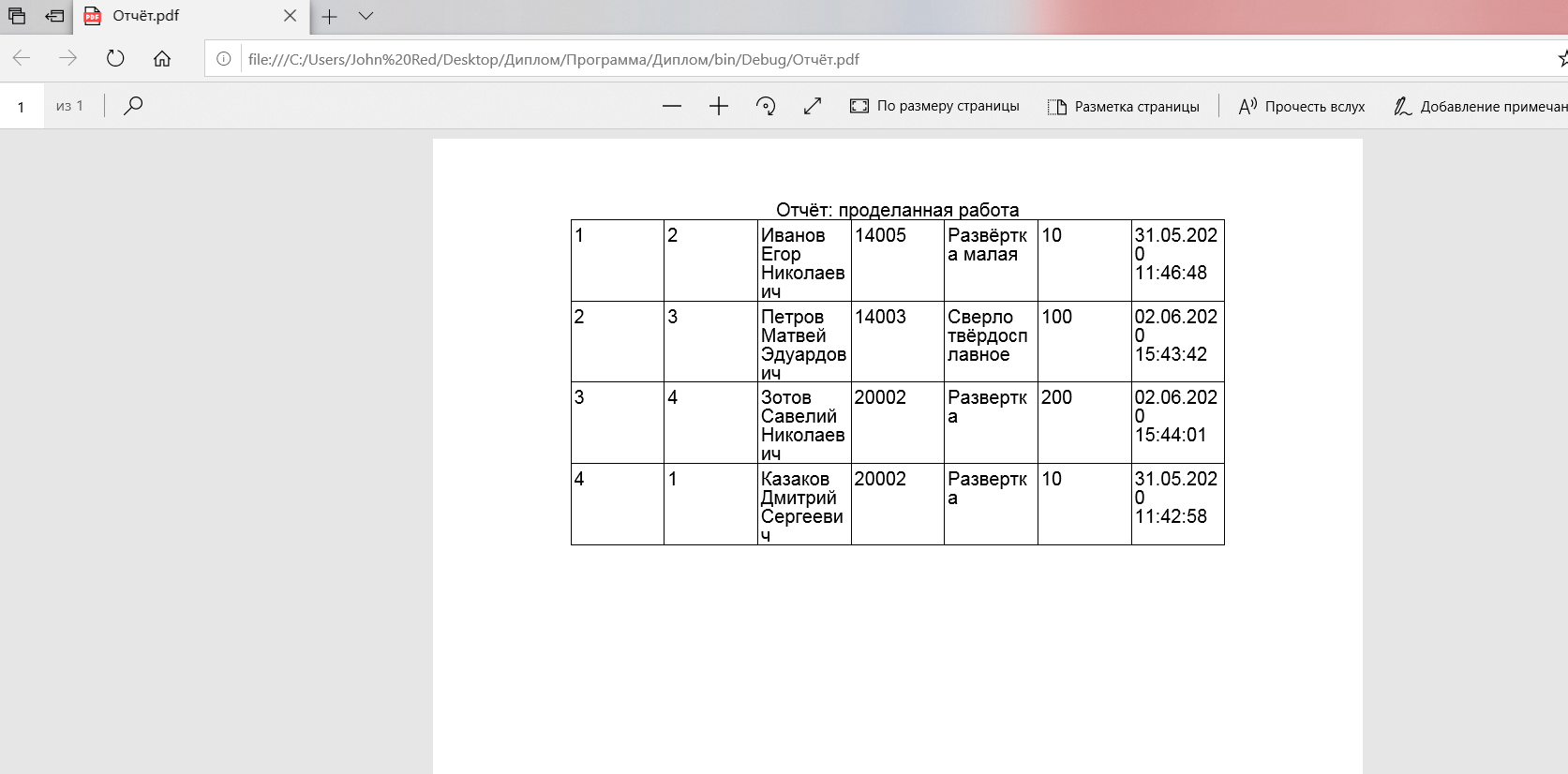


Рисунок 4.6.6 – пример создания отчёта

# **ЗАКЛЮЧЕНИЕ**

Во время выполнения дипломной работы мне пришлось столкнуться с многими трудностями, которые в процесс меняли задуманный проект. При этом основа осталось та же, поэтому во время создания диплома была спроектирована и написана программа, автоматизирующая бизнес-процессы предприятия ООО ”Инструмент-Сервис”. Во время изучения предметной области я смог научиться новым способам проектирования, так же при проектировании было получен практический навык работы с языки, а также с языком программы C#. Получены знания по созданию баз данных и управлению ими. Была изучена экономическая и правовая часть создания проектов. В частности, получены навыки общения с заказчиком, написания технического задания.

После выполнения работы для заказчика, был замечен положительный рост экономического состояния предприятия. Была налажена система поставки и производства. Жалобы со стороны рабочих сократились, что привело к повышенной эффективности работы.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Криспин Л., Грегори Д. Гибкое тестирование. Практическое руководство для тестировщиков ПО и гибких команд: Издательство Вильямс, 2010г.

<http://padabum.com/d.php?id=16896>

1. Албахари Д. С# 7.0. Карманный справочник: Издательство Вильямс, 2018г.

<https://www.ozon.ru/context/detail/id/142089789/>

1. Мартин Р. Принципы, паттерны и методики гибкой разработки на языке C#: Издательство Символ-Плюс, 2011г.

<https://litportal.ru/avtory/robert-martin-11751455/kniga-principy-patterny-i-metodiki-gibkoy-razrabotki-na-yazyke-c-741934.html>

1. Прайс М. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов: Издательство Питер, 2018г.

<http://diggerdnepr.ddns.net/wp-content/uploads/2019/03/c_7_i_net_core_razrabotka_dlya_professionalov_.pdf>

1. Филипп Д. Язык программирования C# 7 и платформы .NET и .NET Core: Диалектика-Вильямс, 2019г.

<http://sd.blackball.lv/books/17526?mode=read>

1. Сергей Т. Паттерны проектирования на платформе .NET: Издательство Питер, 2016г.

<https://goo.su/1Dhm>