ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМ. Н. П. ОГАРЁВА»

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная цикловая комиссия (кафедра) общепрофессиональных и специальных (информационно-коммуникационных) дисциплин

**ОТЧЁТ**

по производственной практике обучающегося 4 курса

а Производственная практика (по профилю специальности) а  
тип практики в соответствии с ФГОС СПО

специальность 09.02.03 Программирование в компьютерных системах  
 код, наименование

ООО «ТТ-ПРО»

место прохождения практики: населенный пункт, профильная организация, структурное подразделение

срок прохождения практики с 25 ноября 2019 г. по 07 декабря 2019 г.

Автор отчёта Р. А. Куроедов

подпись, дата, инициалы, фамилия

Обозначение отчёта ОП– 02069964–П– 09.02.03– 11–19

Руководители практики:

от университета Е. Е. Старушенкова

подпись, дата, инициалы, фамилия

Руководители практики:

от профильной организации М. Ю. Чаиркин

подпись, дата, инициалы, фамилия

Отчёт защищён \_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата

Саранск

2019

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМ. Н. П. ОГАРЁВА»

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная цикловая комиссия (кафедра) общепрофессиональных и специальных (информационно-коммуникационных) дисциплин

УТВЕРЖДАЮ

Председатель ВПЦК

(кафедры)

общепрофессиональных и

специальных (информационно-коммуникационных) дисциплин,

преподаватель ПКК

\_\_\_\_\_\_М. Ф. Петянкин

(подпись) «\_\_\_» \_\_\_\_\_\_\_\_ 20\_\_ г.

**ЗАДАНИЕ**

на производственную практику (по профилю специальности) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

вид практики

обучающемуся Куроедову Роману Александровичу 4 курса, 409 группы, специальности 09.02.03 Программирование в компьютерных системах

Место прохождения практики ООО «ТТ-ПРО»

населенный пункт, профильная организация

Срок прохождения практики с 25 ноября 2019 г. по 07 декабря 2019 г.

начало (дата) – окончание (дата)

Срок представления отчёта и отзыва руководителя практики от профильной организации на защиту 07 декабря 2019 г.

дата

Общее задание

1 Цели и задачи практики:

Целями прохождения производственной практики (по профилю специальности) являются систематизация, закрепление и расширение теоретических знаний по специальности, приобретение обучающимися практического опыта, последовательное формирование практических навыков и умений, обеспечение связи практики с теоретическим обучением.

Задачи производственной практики:

* развитие первичных навыков по специальности   
  09.02.03 Программирование в компьютерных системах;
* получение навыков разработки программных модулей;
* соблюдение правил техники безопасности и санитарных норм;
* овладение правильными приемами работы.

2 Компетенции обучающегося, формируемые в результате прохождения практики:

1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
2. Организовать собственную деятельность, выбирать типовые методы   
   и способы выполнения профессиональных задач, оценивать их эффективность   
   и качество.
3. Принимать решения в стандартных и нестандартных ситуациях   
   и нести за них ответственность.
4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального   
   и личностного развития.
5. Использовать информационно-коммуникационные технологии   
   в профессиональной деятельности.
6. Работать в коллективе и в команде, эффективно общаться   
   с коллегами, руководством, потребителями.
7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения задания.
8. Самостоятельно определять задачи профессионального   
   и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
9. Ориентироваться в условиях частой смены технологий   
   в профессиональной деятельности.

В результате освоения компетенций базовой подготовки специальности   
09. 02. 03 Программирование в компьютерных системах техник-программист должен обладать профессиональными компетенциями, которые соответствуют основным видам профессиональной деятельности:

1. Разрабатывать объекты базы данных.
2. Реализовывать базу данных в конкретной СУБД.
3. Решать вопросы администрирования базы данных.
4. Реализовывать методы и технологии защиты информации в базах данных.
5. Индивидуальное задание на практику:

* работа по использованию различных моделей данных;
* корректная работа по нормализации отношений;
* работа по созданию объектов БД (таблиц);
* создание объектов БД (форм, отчетов);
* установка атрибутов и ключей;
* работа по построению схем БД (различного уровня сложности);
* работа с манипулированием данными (хранение, добавление, редактирование данных);
* работа по сортировке, поиску и фильтрации данных;
* работа по построению запросов к СУБД (различного уровня сложности);
* создание концептуальной, логической и физической модели данных;
* разработка и эксплуатация серверной части: создание, модификация и удаление таблиц;
* создание, перестройка и удаление индекса;
* разработка и эксплуатация клиентской части (интерфейса);
* создание хранимых процедур и триггеров в БД;
* работа по администрированию БД;
* решение вопросов обеспечения безопасности СУБД;
* технические методы и средства защиты БД;
* контроль доступа к данным, управление привилегиями   
  пользователей БД.

Руководитель практики

от университета 03.06.2019 Е. Е. Старушенкова

подпись, дата, инициалы, фамилия

Руководитель практики

от профильной организации 03.06.2019 М. Ю. Чаиркин

подпись, дата, инициалы, фамилия

Задание к исполнению

принял 03.06.2019 Р. А. Куроедов

подпись, дата, инициалы, фамилия

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМ. Н. П. ОГАРЁВА»

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная цикловая комиссия (кафедра) общепрофессиональных и специальных (информационно-коммуникационных) дисциплин

ДНЕВНИК

|  |
| --- |
| по производственной практике (по профилю специальности) |
| вид, тип практики |

обучающегося III курса Куроедова Романа Александровича

Ф.И.О.

специальность 09.02.03 Программирование в компьютерных системах

код, наименование

Начало практики 03 июня 2019 г.

Окончание практики 29 июня 2019 г.

Дневник представлен руководителю практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись практиканта, дата

Дневник проверен

руководителем практики

от университета Е. Е. Старушенкова

подпись, дата, инициалы, фамилия

Саранск

2019

ЗАПИСИ

о работах, выполненных в период практики

|  |  |  |
| --- | --- | --- |
| Дата | Краткое содержание | Замечания и подпись  руководителя |
| 03.06.19 | Установочная конференция, изучение правил техники безопасности |  |
| 04.06.19 | Анализ предметной области |  |
| 05.06.19 | Определение структуры данных |  |
| 06.06.19 | Создание модели данных |  |
| 07.06.19 | Определение структуры базы данных |  |
| 08.06.19 | Определение структуры базы данных |  |
| 10.06.19 | Создание ER-диаграммы |  |
| 11.06.19 | Выбор СУБД |  |
| 13.06.19 | Реализация защиты СУБД и БД |  |
| 14.06.19 | Изучение возможностей взаимодействия C#  и MySQL |  |
| 15.06.19 | Изучение возможностей взаимодействия C#  и MySQL |  |
| 17.06.19 | Установка необходимых расширений для реализации взаимодействия  C# и MySQL |  |
| 18.06.19 | Разработка и реализация пользовательского интерфейса для взаимодействия с БД |  |
| 19.06.19 | Разработка и реализация пользовательского интерфейса для взаимодействия с БД |  |
| 20.06.19 | Разработка и реализация пользовательского интерфейса для взаимодействия с БД |  |
| 21.06.19 | Разработка и реализация пользовательского интерфейса для взаимодействия с БД |  |
| 22.06.19 | Разработка и реализация пользовательского интерфейса для взаимодействия с БД |  |
| 24.06.19 | Разработка и реализация пользовательского интерфейса для взаимодействия с БД |  |
| 25.06.19 | Создание и встраивание SQL-запросов |  |
| 26.06.19 | Создание и встраивание SQL-запросов |  |
| 27.06.19 | Тестирование модуля взаимодействия с БД |  |
| 28.06.19 | Подготовка отчёта |  |
| 29.06.19 | Защита отчёта |  |

Руководитель практики Е. Е. Старушенкова

подпись, дата, инициалы, фамилия

Руководитель практики

от профильной организации М. Ю. Чаиркин

подпись, дата, инициалы, фамилия

М. П.

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| ВВЕДЕНИЕ | 10 |
| 1 Ознакомление со структурой предприятия | 12 |
| 2 Предметная область | 13 |
| 3 Программный инструментарий | 14 |
| 4 Реализация модуля взаимодействия с базой данных | 15 |
| 4.1 Интерфейс | 15 |
| 4.2 База данных | 15 |
| 4.3 Взаимодействие между пользователем и базой данных | 17 |
| 5.Организация защиты СУБД и БД | 19 |
| ЗАКЛЮЧЕНИЕ | 20 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 22 |
| ПРИЛОЖЕНИЕ А (обязательное) Схема данных | 23 |
| ПРИЛОЖЕНИЕ Б (обязательное)Схема базы данных | 24 |
| ПРИЛОЖЕНИЕ В (обязательное) Интерфейс | 25 |
| ПРИЛОЖЕНИЕ Г (обязательное)Листинг метода ShiftOpen() | 28 |
| ПРИЛОЖЕНИЕ Д (обязательное) Листинг метода ShiftClose() | 30 |
| ПРИЛОЖЕНИЕ Е (обязательное) Листинг метода AddToDBCheck() страницы Sellings | 31 |
| ПРИЛОЖЕНИЕ Ж (обязательное)Листинг метода AddToDBCheck() страницы Return | 34 |
| ПРИЛОЖЕНИЕ И (обязательное) Листинг метода Paid() страницы Sellings | 37 |
| ПРИЛОЖЕНИЕ И (обязательное) Листинг метода Paid() страницы Return | 39 |

**ВВЕДЕНИЕ**

Целью прохождения производственной практики (по профилю специальности) является: закрепление и углубление теоретической подготовки обучающегося по работе с базами данных; приобретение навыков самостоятельной и научно-исследовательской работы; приобретение опыта применения вычислительной техники для решения практических задач; приобретение практических навыков и компетенций в сфере профессиональной деятельности.

Задачами производственной практики (по профилю специальности) является систематизация, обобщение, закрепление и углубление знаний и умений, формирование общих и профессиональных компетенций, приобретение практического опыта в рамках указанных ниже профессиональных модулей, а также закрепление и совершенствование приобретенных в процессе обучения профессиональных умений обучающихся по изучаемой специальности, развитие общих и профессиональных компетенций, освоение современных производственных процессов, адаптация обучающихся к конкретным условиям деятельности организаций различных организационно-правовых норм.

Во время прохождения практики выполнили следующие виды работ:

Индивидуальное задание на практику:

* работа по использованию различных моделей данных;
* корректная работа по нормализации отношений;
* работа по созданию объектов БД (таблиц);
* создание объектов БД (форм, отчетов);
* установка атрибутов и ключей;
* работа по построению схем БД (различного уровня сложности);
* работа с манипулированием данными (хранение, добавление, редактирование данных);
* работа по сортировке, поиску и фильтрации данных;
* работа по построению запросов к СУБД (различного уровня сложности);
* создание концептуальной, логической и физической модели данных;
* разработка и эксплуатация серверной части: создание, модификация и удаление таблиц;
* создание, перестройка и удаление индекса;
* разработка и эксплуатация клиентской части (интерфейса);
* создание хранимых процедур и триггеров в БД;
* работа по администрированию БД;
* решение вопросов обеспечения безопасности СУБД;
* технические методы и средства защиты БД;
* контроль доступа к данным, управление привилегиями   
  пользователей БД.

**1 Ознакомление со структурой предприятия**

ООО «ТТ-ПРО» оказывает следующие услуги по поставке, монтажу   
и настройке:

* торгово-кассового оборудования, системы автоматизации для магазинов   
  с использованием штрих-кодирования и товароучётных программ;
* систем автоматизации для розничных магазинов, ресторанов и кафе;
* банковского оборудования (детекторы валют, счетчики банкнот);
* весового оборудования;
* систем видеонаблюдения;
* стеллажного оборудование;
* расходных материалов (чековая лента, термоэтикетки для печати   
  штрих-кодов);
* различного программного обеспечения;
* приемников и кнопок вызова персонала;
* техническому обслуживанию касс, весов.

**2 Предметная область**

Предметная область — касса. Она включает в себя следующие компоненты: учёт смен, учёт продаж, учёт возвратов и учёт товаров.

В соответствии с предметной областью система строится с учётом следующих особенностей:

* осуществление продажи как товаров, занесённых в базу данных,   
  так и товаров со свободной ценой, информация о которых не хранится в базе данных;
* осуществление возврата как товаров, занесённых в базу данных,   
  так и товаров со свободной ценой, информация о которых не хранится в базе данных;
* возможность внесения и изъятия денежных средств из кассы;
* возможность программного начала и завершения смены.

Базовые сущности этой предметной области:

* Смены. Атрибуты смен — id, дата начала и завершения смены, продажи, возвраты, изъятия и внесения.
* Продажи. Атрибуты продаж — артикул, наименование, количество, цена, id смены.
* Товары. Атрибуты товара — артикул, наименование товара, количество, цен, скидка.
* Возвраты. Атрибуты возврата — артикул, наименование, цена, количество, id смены.

Модель данных представлена в приложении А. ER-диаграмма базы данных представлена в приложении Б.

**3 Программный инструментарий**

Как среда разработки используется Microsoft Visual Studio 2019 Community в силу мощного функционала, бесплатного распространения и наличия навыков работы с данной IDE.

Microsoft Visual Studio 2019 Community — это бесплатная редакция интегрированной среды разработки от компании Microsoft. Данная редакция имеет меньший функционал по сравнению с платными редакциями, однако она включает все необходимое для создания полноценных приложений. Походит для индивидуальных разработчиков и обучения.

В качестве системы управления базами данных используется phpMyAdmin на платформе OpenServer в силу мощного функционала, бесплатного распространения и наличия навыков работы с данным программным комплексом.

PhpMyAdmin — бесплатный программный инструмент, написанный на PHP, предназначен для управления администрированием MySQL через web. phpMyAdmin поддерживает широкий спектр операций на MySQL и MariaDB. Часто используемые операции (управление базами данных, таблицами, колонки, отношения, индексы, потребители, разрешения и т.д.) можно выполнить через пользовательский интерфейс, в то время как у вас все еще есть возможность напрямую выполнять любой оператор SQL.

**4 Реализация модуля взаимодействия с базой данных**

**4.1 Интерфейс**

В качестве интерфейса взаимодействия пользователя с базой данных используется программная оболочка «TT-PRO» разработанная в рамках практики по профессиональному модулю ПМ.01 в срок с 06 мая 2019 года по   
01 июня 2019 года. Скриншоты интерфейса представлены в приложении В.

**4.2 База данных**

Для базы данных в СУБД был создан специальный пользователь «ttproadmin» с паролем «5687», не имеющий глобальных привилегий, а лишь привилегии на данную базу данных. База данных включает в себя следующие объекты:

* таблица items — таблица, хранящая данные о товарах, имеющихся в наличии;
* таблица returns — таблица, хранящая данные о возвратах товара;
* таблица sellings — таблица, хранящая данные о продажах товара;
* таблица shifts — таблица, хранящая данные о сменах.

Таблица 1 — Структура таблицы items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип | Сравнение | Null | Комментарий |
| \*articul | varchar(8) | utf8\_general\_ci | нет | артикул |
| name | varchar(100) | utf8\_general\_ci | нет | наименование |
| price | double |  | да | цена, руб. |
| number | int(11) |  | да | количество |
| discount | double |  | да | скидка, % |

Таблица 2 — Структура таблицы returns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип | Сравнение | Null | Комментарий |
| \*shift\_id | int(11) |  | нет | id смены |
| \*articul | varchar(8) | utf8\_general\_ci | нет | артикул |
| name | varchar(100) | utf8\_general\_ci | нет | наименование |
| price | double |  | нет | цена, руб. |
| number | int(11) |  | нет | количество |

Таблица 3 — Структура таблицы sellings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип | Сравнение | Null | Комментарий |
| \*shift\_id | int(11) |  | нет | id смены |
| \*articul | varchar(8) | utf8\_general\_ci | нет | артикул |
| name | varchar(100) | utf8\_general\_ci | нет | наименование |
| price | double |  | нет | цена, руб. |
| number | int(11) |  | нет | количество |

Таблица 4 — Структура таблицы shifts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип | Null | Дополнительно | Комментарий |
| \*id | int(11) | нет | AUTO\_INCREMENT | id смены |
| startdate | date | нет |  | дата начала |
| enddate | date | да |  | дата окончания |
| cash | double | да |  | количество денег в кассе |
| sellings | double | да |  | сумма продаж |
| returns | double | да |  | сумма возвратов |
| additions | double | да |  | сумма внесений |
| withdrawals | double | да |  | сумма изъятий |

**4.3 Взаимодействие между пользователем и базой данных**

При запуске программы самой первой страницей является shifts.

Для начала смены необходимо нажать кнопку «Начать смену».   
При нажатии вызывается метод ShiftOpen() класса ShiftStats. Данный метод создает новую запись в таблице shifts заполняя поля id и startdate. Листинг метода ShiftOpen() представлен в приложении Г.

Вся дальнейшая общая информация о смене хранится в полях класса ShiftStats и заносится в базу данных только при завершении смены.

При нажатии кнопки «Завершить смену» либо при закрытии основного окна вызывается метод ShiftClose() класса ShiftStats. Данный метод отправляет на сервер SQL-запрос обновляющий последнюю запись в таблице shifts, заполняя оставшиеся поля. Листинг метода ShiftClose() представлен   
в приложении Д.

При осуществлении продаж для поиска товаров по артикулу на странице sellings реализовано поле для ввода артикула товара. При нажатии   
на кнопку «Найти» Вызывается метод AddDBToCheck(), в котором реализован функционал поиска товара по артикулу и добавление его в коллекцию Items. Листинг метода AddDBToCheck() представлен в приложении E.

При нажатии на кнопку «Оплатить» производится вызов метода Paid(), отвечающего за внесение записей в таблицу sellings или их обновление, если записи с данным ключом существуют. При создании записи о проданном товаре в поле price заносится значение за штуку с учетом скидки. При обновлении записи увеличивается значение поля number. Помимо этого, в таблице items происходит уменьшение значения поля number у записи с соответствующим артикулом. Листинг метода Paid() представлен в приложении И.

Функционал, реализованный на странице Return идентичен ранее описанному и использует метод с таким же названием. В отличие от страницы Sellings данные заносятся в таблицу returns, а значение поля number   
в таблице items увеличивается. Листинг метода AddDBToCheck() страницы Return представлен в приложении Ж. Листинг метода Paid() страницы Return представлен в приложении К.

**5 Организация защиты СУБД и БД**

Для защиты СУБД в phpMyAdmin предусмотрено разделение прав доступа между пользователями. Пользователя может создать только пользователь, обладающий всеми глобальными правами. Такими пользователями являются root и mysql. Создать пользователя можно во вкладке «Учетные записи пользователей». Для защиты СУБД был создан специальный пользователь «ttproadmin» с паролем «5687», не имеющий глобальных привилегий, а лишь привилегии на данную базу данных.

Для защиты базы данных привилегиями на базу данных ttpro обладает лишь пользователь ttproadmin и корневые пользователи root и mysql; для остальных пользователей данная база данных является невидимой.

**ЗАКЛЮЧЕНИЕ**

В результате прохождения производственной практики (по профилю специальности) в рамках профессионального модуля ПМ.02 при выполнении работ был получен практический опыт:

* развитие первичных навыков по специальности 09.02.03 Программирование в компьютерных системах;
* получение навыков разработки программных модулей;
* соблюдением правил техники безопасности и санитарных норм;
* овладение правильными приемами работы.

Были усвоены следующие компетенции:

1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
2. Организовать собственную деятельность, выбирать типовые методы   
   и способы выполнения профессиональных задач, оценивать их эффективность   
   и качество.
3. Принимать решения в стандартных и нестандартных ситуациях   
   и нести за них ответственность.
4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального   
   и личностного развития.
5. Использовать информационно-коммуникационные технологии   
   в профессиональной деятельности.
6. Работать в коллективе и в команде, эффективно общаться   
   с коллегами, руководством, потребителями.
7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения задания.
8. Самостоятельно определять задачи профессионального   
   и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
9. Ориентироваться в условиях частой смены технологий   
   в профессиональной деятельности.

В результате освоения компетенций базовой подготовки специальности  
09.02.03 Программирование в компьютерных системах техник-программист должен обладать профессиональными компетенциями, которые соответствуют основным видам профессиональной деятельности:

1. Разрабатывать объекты базы данных.
2. Реализовывать базу данных в конкретной СУБД.
3. Решать вопросы администрирования базы данных.
4. Реализовывать методы и технологии защиты информации в базах данных.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. METANIT.COM [Электронный ресурс]. —   
   Режим доступа: https://metanit.com/sharp/wpf/. — Дата доступа: 17.06.2019.
2. Stack Overflow [Электронный ресурс]. —  
    Режим доступа: https://ru.stackoverflow.com. — Дата доступа: 17.06.2019.
3. MySQL Developer Zone [Электронный ресурс]. —   
   Режим доступа: https://dev.mysql.com. — Дата доступа: 17.06.2019.
4. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер, Е. Матвеев. — Санкт-Петербург : Издательство Питер СПб, 2019. — 896 с.
5. Дунаев, В. В. Базы данных. Язык SQL для студента / В. В. Дунаев. — Санкт-Петербург : БХВ-Петербург, 2017. — 376 с.
6. Маркин, А. В. Программирование на sql в 2 ч. Часть 1. / А. В. Маркин. — Москва : Издательство Юрайт, 2018. — 362 с.
7. Маркин, А. В. Программирование на sql в 2 ч. Часть 2. / А. В. Маркин. — Москва : Издательство Юрайт, 2018. — 362 с.
8. Кудрина, Е В. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C#. / Е В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2019. — 322 с.

**ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Схема данных**

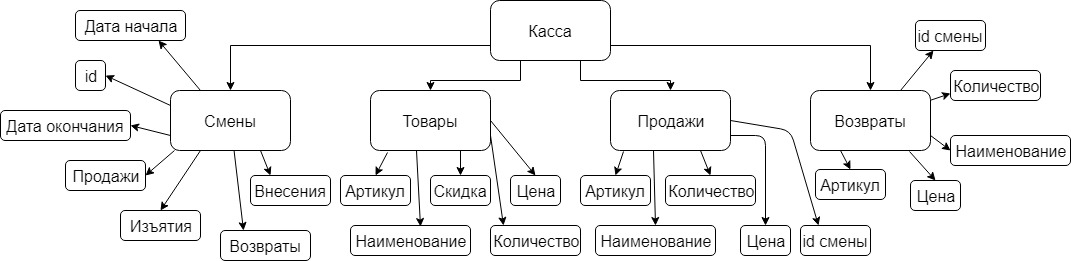


Рисунок А.1 — Модель данных

**ПРИЛОЖЕНИЕ Б**

**(обязательное)**

**Схема базы данных**



Рисунок Б.1 — ER-диаграмма

**ПРИЛОЖЕНИЕ В**

**(обязательное)**

**Интерфейс**

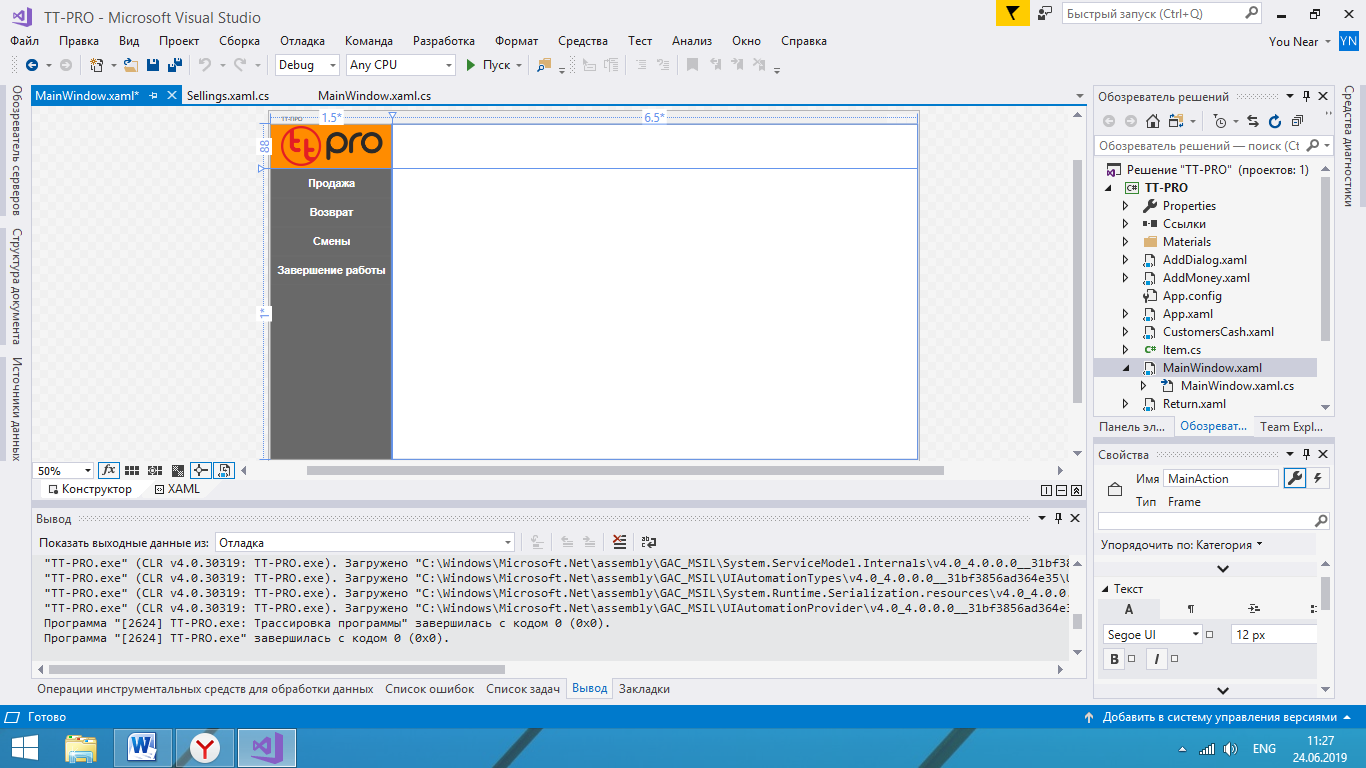


Рисунок В.1 — Окно MainWindow

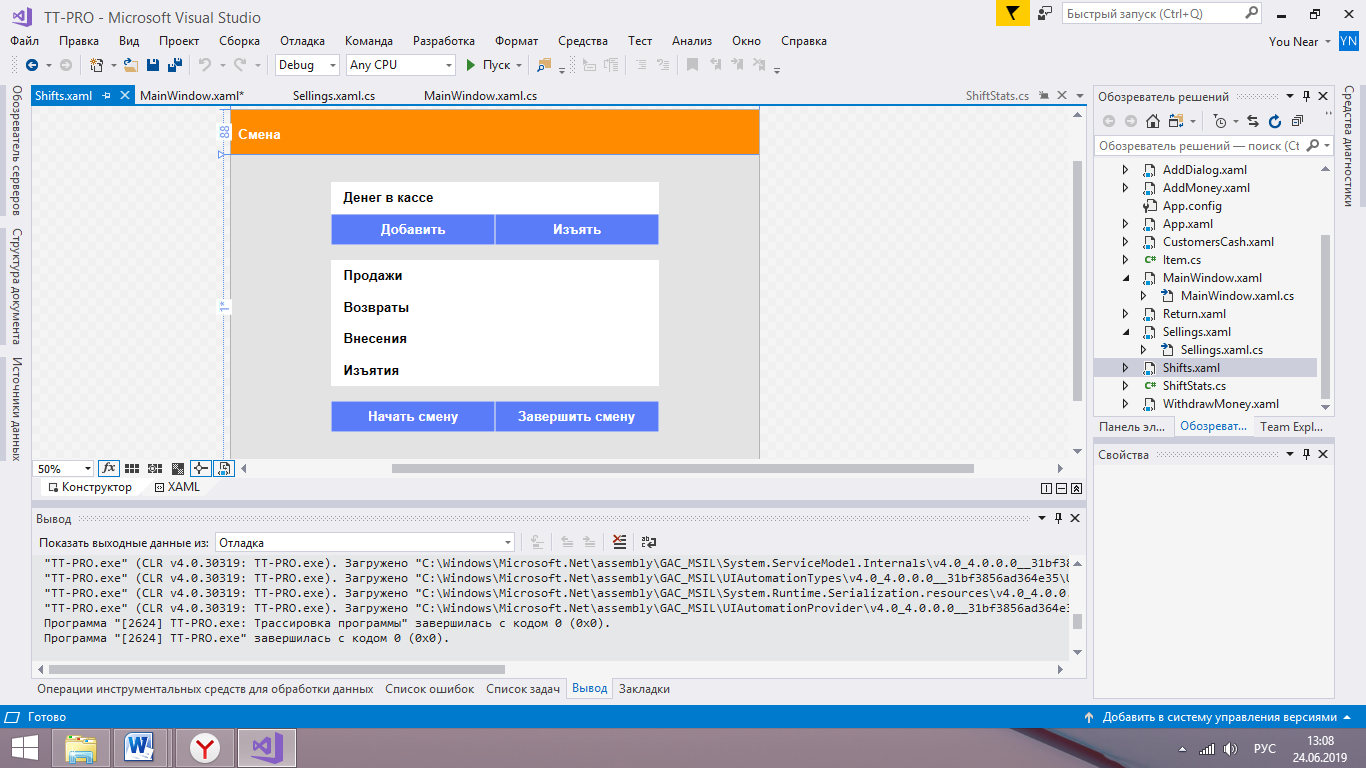


Рисунок В.2 — Страница Shifts

Продолжение ПРИЛОЖЕНИЯ В

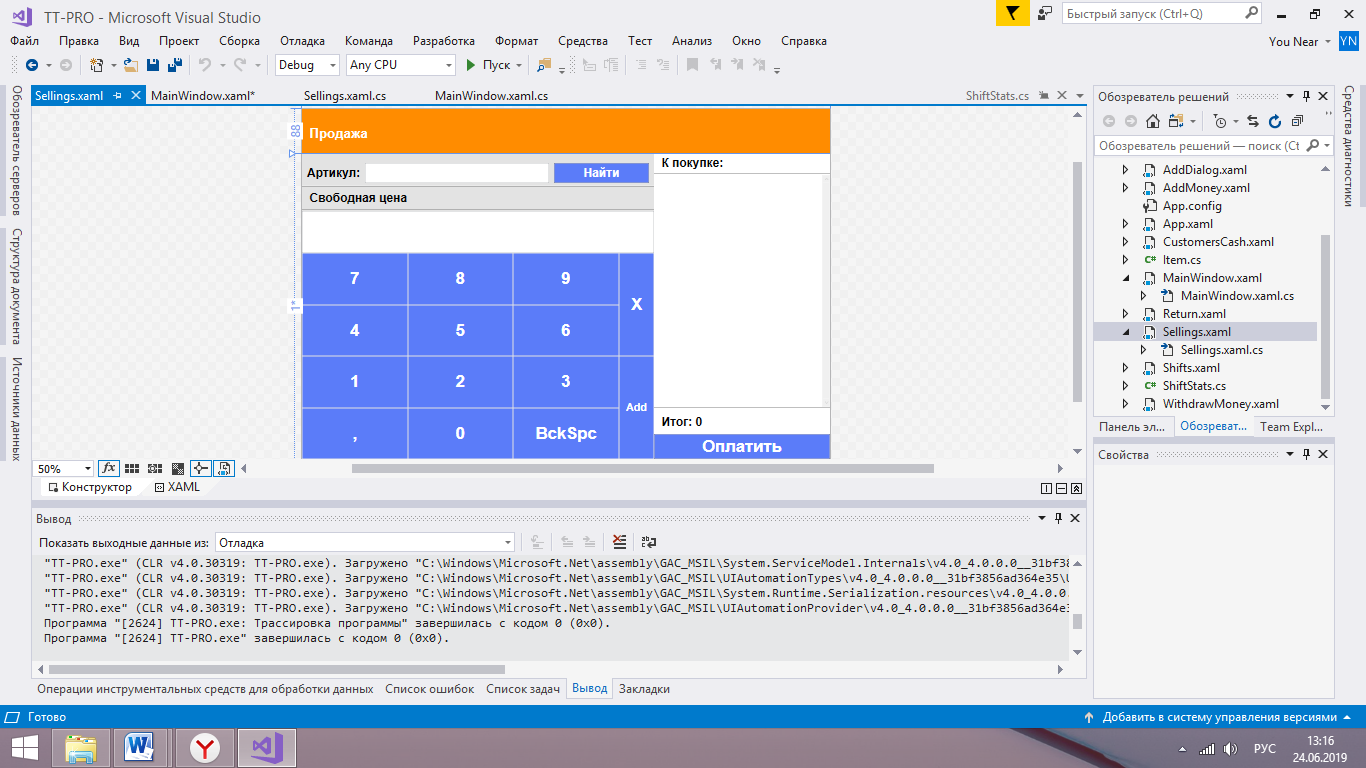


Рисунок В.3 — Страница Sellings

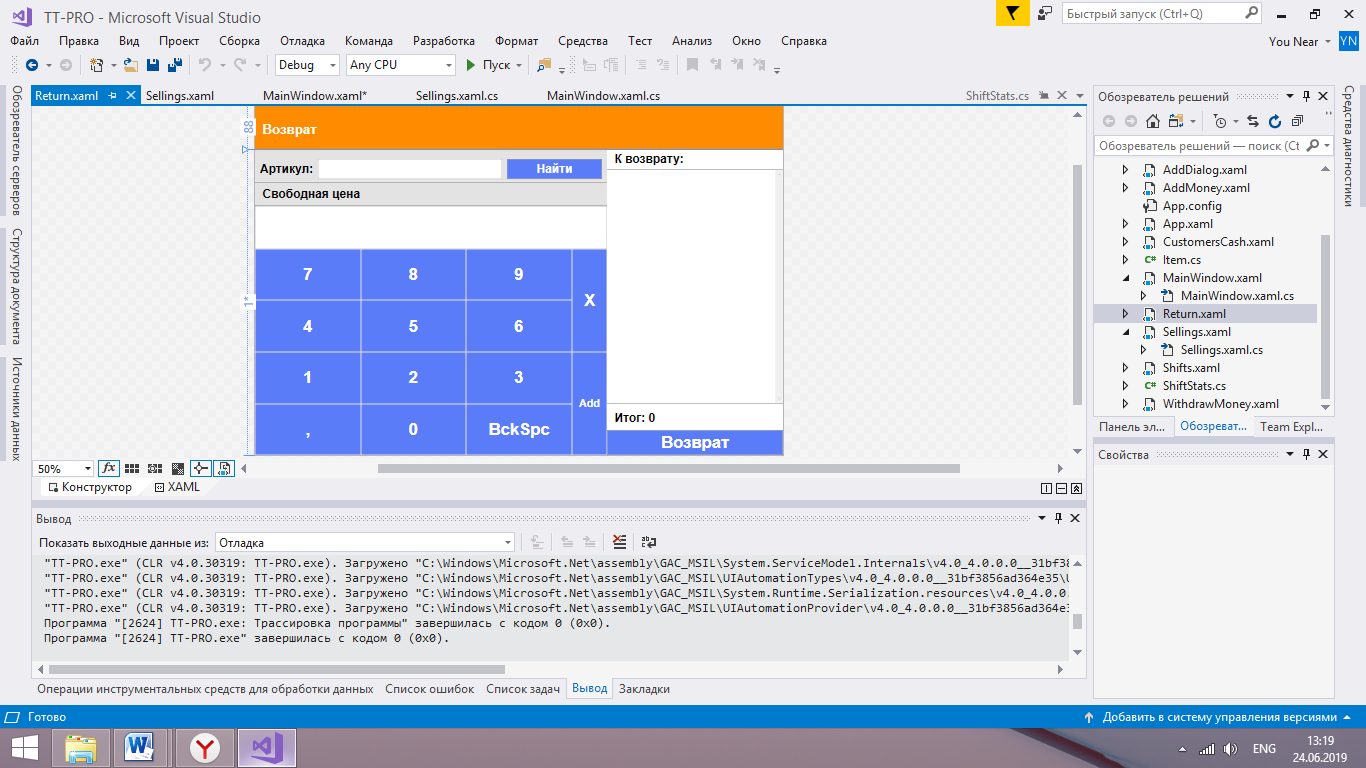


Рисунок В.4 — Страница Return

Окончание ПРИЛОЖЕНИЯ В

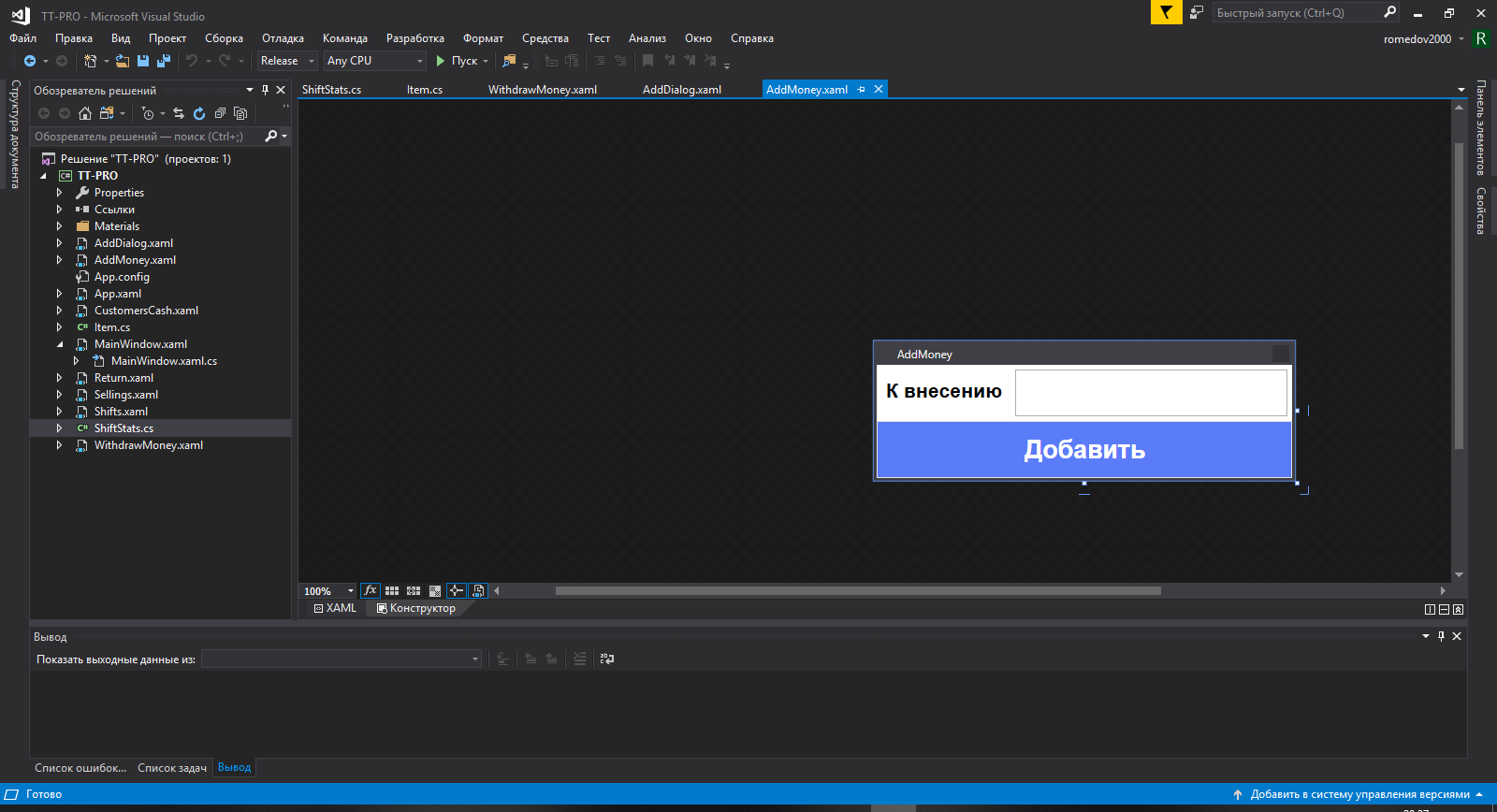


Рисунок В.5 — Окно AddMoney

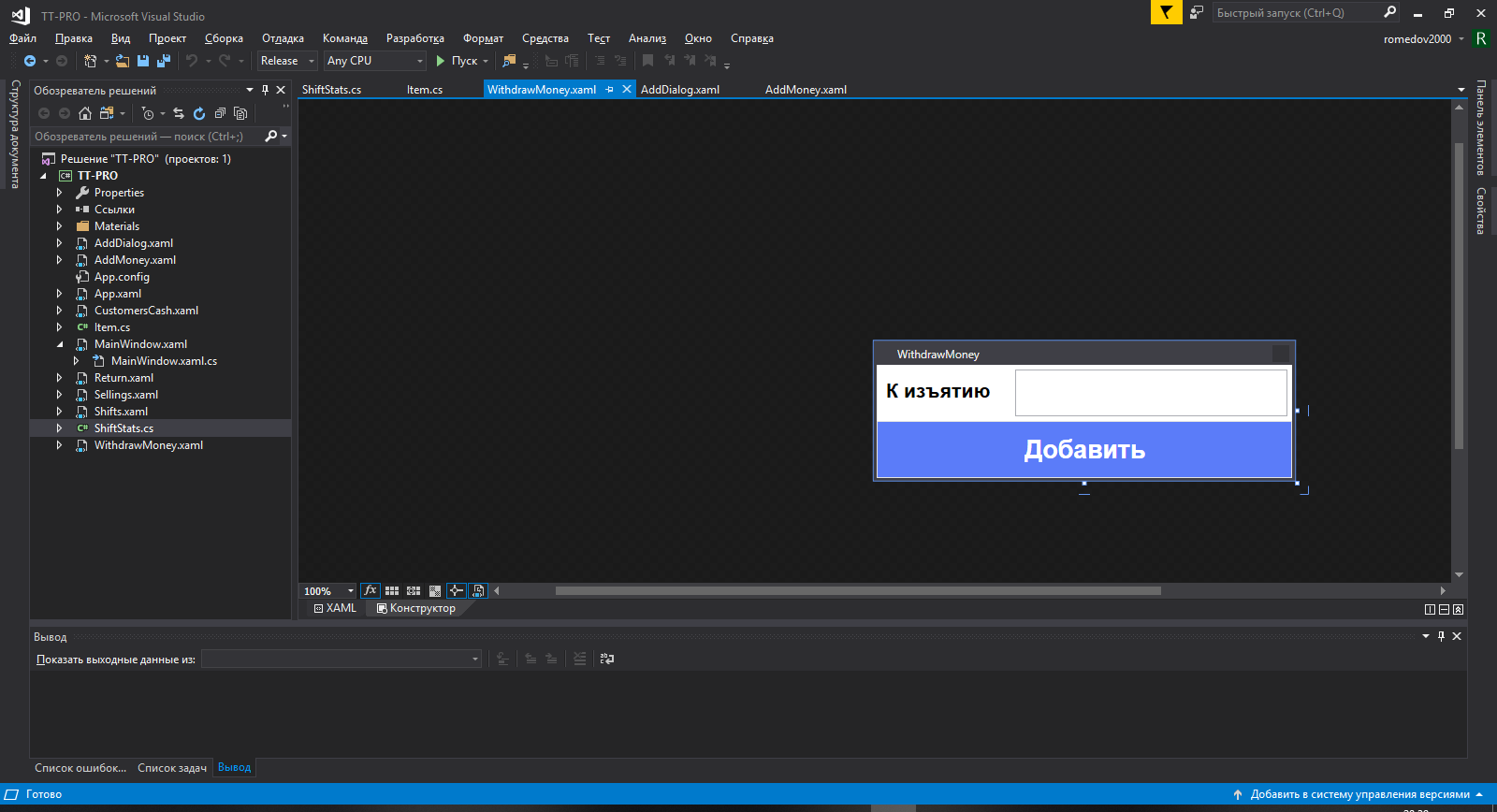


Рисунок В.6 — Окно WithdrawMoney

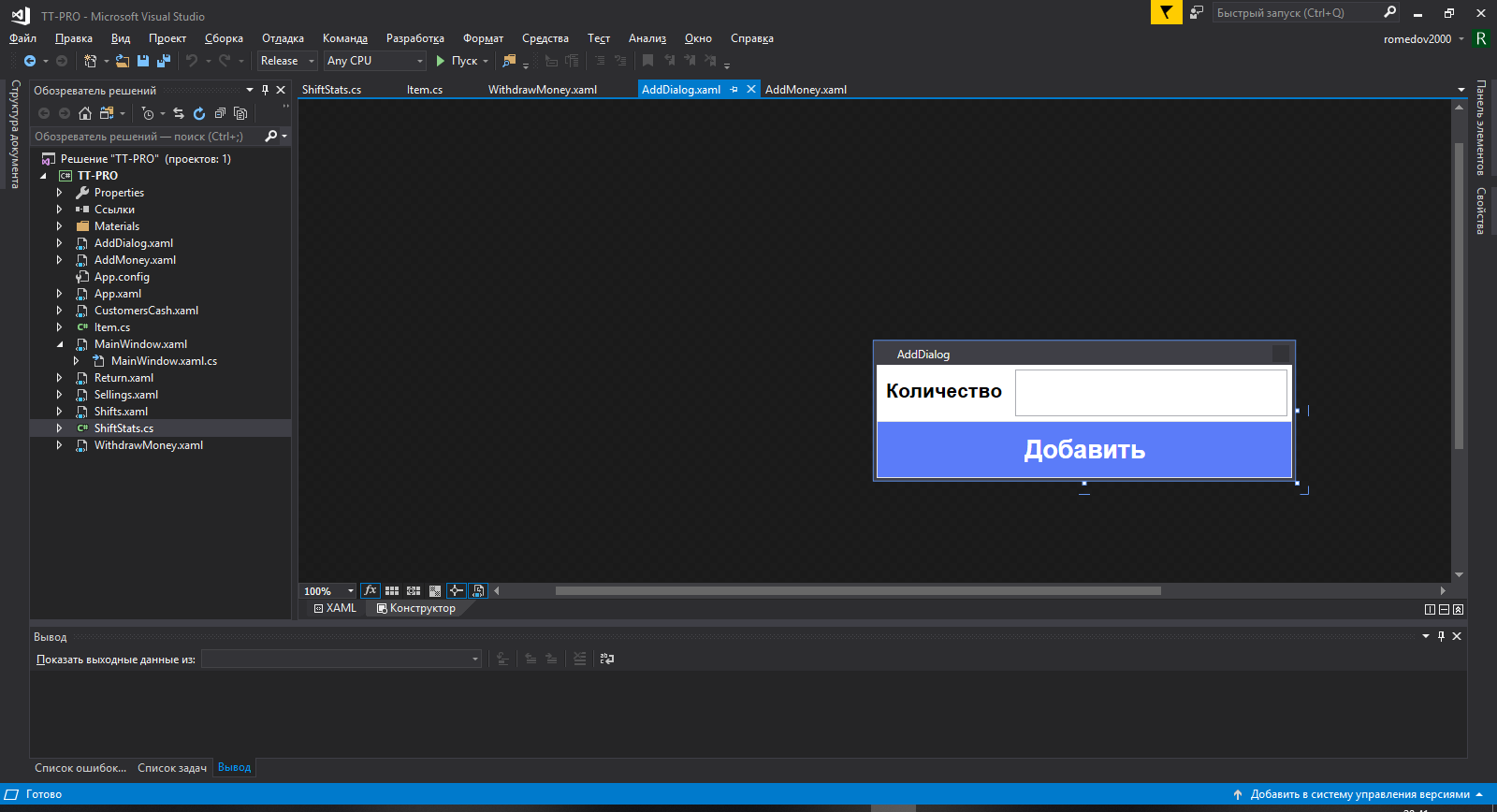


Рисунок В.7 — Окно AddDialog

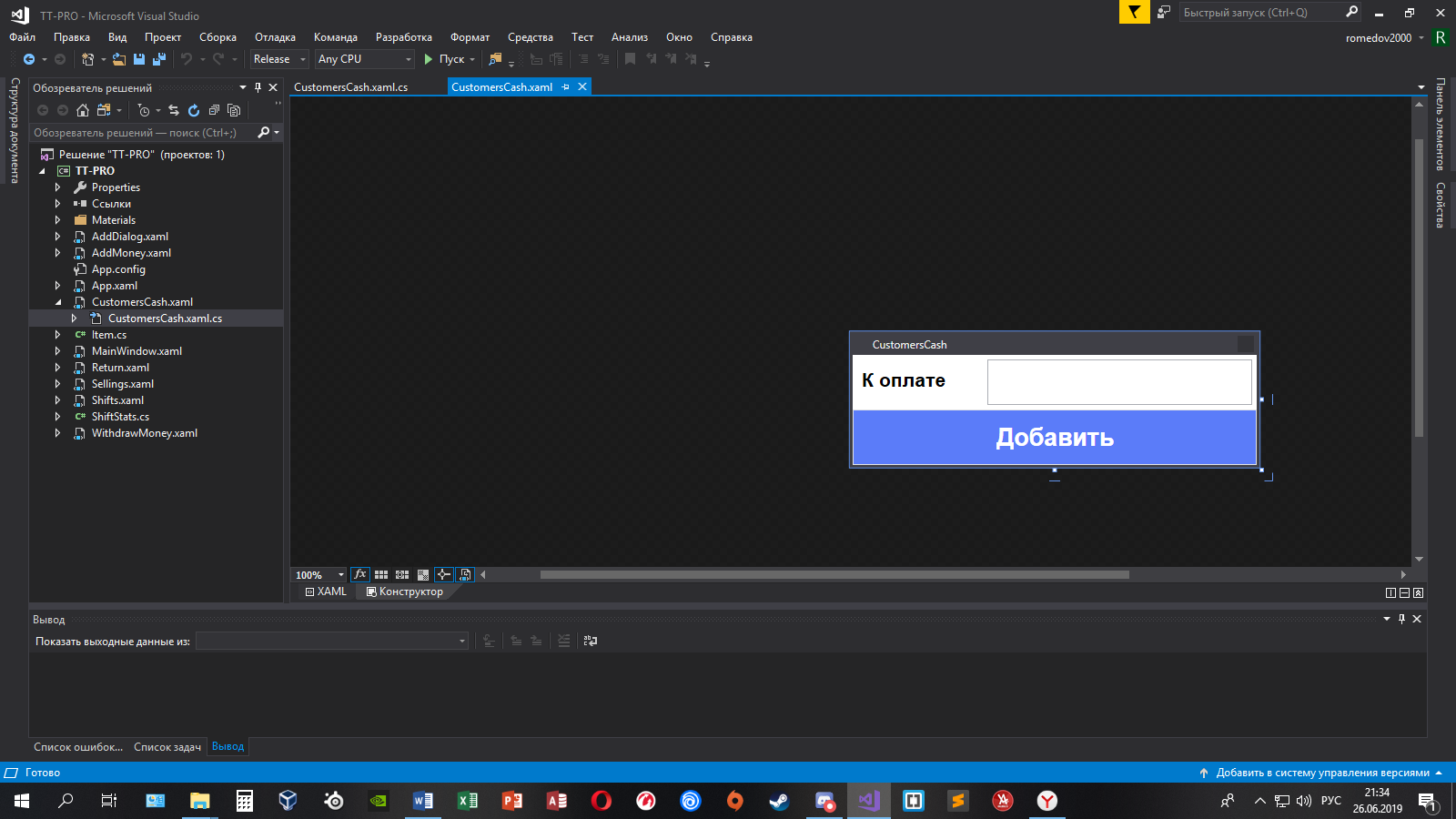


Рисунок В.8 — Окно AddDialog

**ПРИЛОЖЕНИЕ Г**

**(обязательное)**

**Листинг метода ShiftOpen()**

public bool ShiftOpen()

{

string connStr = $"server={host};user={user};database={db};password={passwd};";

MySqlConnection conn = new MySqlConnection(connStr);

try

{

conn.Open();

string sqlOut = "SELECT MAX(id) FROM shifts";

MySqlCommand cmdOut = new MySqlCommand(sqlOut, conn);

var res = cmdOut.ExecuteScalar();

if (res != null)

{

id = res.ToString() != "" ? int.Parse(res.ToString()) : 0;

string sqlPreviousCash = $"SELECT cash FROM shifts WHERE id={id}";

MySqlCommand cmdPC = new MySqlCommand(sqlPreviousCash, conn);

var i = cmdPC.ExecuteScalar();

if (i!=null)

{

CashBox = double.Parse(i.ToString().Replace('.',','));

string sqlIn = "INSERT INTO shifts (startdate) VALUES (STR\_TO\_DATE(now(), '%Y-%m-%d'))";

MySqlCommand cmdIn = new MySqlCommand(sqlIn, conn);

if (cmdIn.ExecuteNonQuery() < 0)

{

System.Windows.MessageBox.Show($"Сбой при выполнении запроса к базе данных!");

conn.Close();

return false;

}

}

else

{

cashBox = 0;

string sqlIn = "INSERT INTO shifts (startdate) VALUES (STR\_TO\_DATE(now(), '%Y-%m-%d'))";

MySqlCommand cmdIn = new MySqlCommand(sqlIn, conn);

if (cmdIn.ExecuteNonQuery() < 0)

{

Окончание ПРИЛОЖЕНИЯ Г

System.Windows.MessageBox.Show($"Сбой при выполнении запроса к базе данных!");

conn.Close();

return false;

}

}

return true;

}

else

{

conn.Close();

return false;

}

}

catch (MySqlException e)

{

System.Windows.MessageBox.Show($"Сбой подключения к базе данных: {e.Message}.");

return false;

}

}

**ПРИЛОЖЕНИЕ Д**

**(обязательное)**

**Листинг метода ShiftClose()**

public bool ShiftClose()

{

string connStr = $"server={host};user={user};database={db};password={passwd};";

MySqlConnection conn = new MySqlConnection(connStr);

try

{

conn.Open();

id++;

string sql = $"UPDATE shifts SET enddate=STR\_TO\_DATE(now(), '%Y-%m-%d'), cash={cashBox.ToString().Replace(',','.')}, sellings={cashGotten.ToString().Replace(',', '.')}, returns={cashReturned.ToString().Replace(',', '.')}, additions={cashAdded.ToString().Replace(',', '.')}, withdrawals={cashWithdrawn.ToString().Replace(',', '.')} WHERE id={id}";

MySqlCommand cmd = new MySqlCommand(sql, conn);

if (cmd.ExecuteNonQuery() < 0)

{

System.Windows.MessageBox.Show($"Сбой при выполнении запроса к базе данных!");

return false;

}

conn.Close();

return true;

}

catch (MySqlException e)

{

System.Windows.MessageBox.Show($"Сбой подключения к базе данных: {e.Message}.");

return false;

}

}

**ПРИЛОЖЕНИЕ Е**

**(обязательное)**

**Листинг метода AddToDBCheck() страницы Sellings**

private void AddDBToCheck(object sender, EventArgs e)

{

AddDialog ad = new AddDialog();

int counter = 0;

if (ArticleInput.Text == "" || ArticleInput.Text == "0")

return;

MySqlConnection conn = new MySqlConnection(connStr);

try

{

conn.Open();

string query = $"SELECT \* FROM items WHERE articul={ArticleInput.Text}";

MySqlCommand cmd = new MySqlCommand(query, conn);

MySqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

ad.ShowDialog();

if (ad.Success)

{

readNum = int.Parse(reader[3].ToString());

if (readNum < AddDialog.num)

{

MessageBox.Show($"На складе осталось только {readNum} штук!");

return;

}

double discount = int.Parse(reader[4].ToString());

discount = 1 - discount / 100;

double finalPrice = Math.Round(Double.Parse(reader[2].ToString()) \* discount, 2);

Items.Add(new Item(reader[0].ToString(), reader[1].ToString(), finalPrice, AddDialog.num));

Grids.Add(new Grid());

ItemViewer.Children.Add(Grids.Last());

var rowDefinition = new RowDefinition();

var converter = new GridLengthConverter();

var columnDefinition0 = new ColumnDefinition()

{

Width = new GridLength(1, GridUnitType.Star),

};

var columnDefinition1 = new ColumnDefinition()

{

Width = new GridLength(110),

};

Продолжение ПРИЛОЖЕНИЯ Е

var columnDefinition2 = new ColumnDefinition()

{

Width = new GridLength(40),

};

Grids.Last().RowDefinitions.Add(rowDefinition);

Grids.Last().ColumnDefinitions.Add(columnDefinition0);

Grids.Last().ColumnDefinitions.Add(columnDefinition1);

Grids.Last().ColumnDefinitions.Add(columnDefinition2);

var color = new BrushConverter();

var ItemNameTextBlock = new TextBlock()

{

Text = Items.Last().Name,

FontFamily = new FontFamily("Arial"),

Background = new SolidColorBrush(Colors.White),

Foreground = new SolidColorBrush(Colors.Black),

FontSize = 18,

TextWrapping = TextWrapping.Wrap,

Margin = new Thickness(10, 5, 0, 5),

};

ItemNameTextBlock.SetValue(TextBlock.FontWeightProperty, FontWeights.DemiBold);

var ItemPriceTextBlock = new TextBlock()

{

Text = (Items.Last().Price \* AddDialog.num).ToString(),

FontFamily = new FontFamily("Arial"),

Background = new SolidColorBrush(Colors.White),

Foreground = new SolidColorBrush(Colors.Black),

FontSize = 18,

Margin = new Thickness(10, 5, 0, 5),

TextWrapping = TextWrapping.Wrap,

};

ItemPriceTextBlock.SetValue(TextBlock.FontWeightProperty, FontWeights.DemiBold);

var ItemRemoveButton = new Button()

{

Content = "X",

FontFamily = new FontFamily("Arial"),

Окончание ПРИЛОЖЕНИЯ Е

Background (Brush)color.ConvertFromString ("#5b7cf9"),

Foreground = new SolidColorBrush(Colors.White),

BorderBrush = new SolidColorBrush(Colors.Transparent),

FontSize = 18,

HorizontalAlignment = HorizontalAlignment.Stretch,

VerticalAlignment = VerticalAlignment.Stretch,

HorizontalContentAlignment = HorizontalAlignment.Center,

VerticalContentAlignment = VerticalAlignment.Center,

Margin = new Thickness(10, 5, 0, 5),

};

ItemRemoveButton.Click += RemoveItemFromItemViewer;

Grid.SetRow(ItemNameTextBlock, 0);

Grid.SetColumn(ItemNameTextBlock, 0);

Grids.Last().Children.Add(ItemNameTextBlock);

Grid.SetRow(ItemPriceTextBlock, 0);

Grid.SetColumn(ItemPriceTextBlock, 1);

Grids.Last().Children.Add(ItemPriceTextBlock);

Grid.SetRow(ItemRemoveButton, 0);

Grid.SetColumn(ItemRemoveButton, 2);

Grids.Last().Children.Add(ItemRemoveButton);

\_conclusion += Math.Round(Items.Last().Price\*Items.Last().Number, 2);

Conclusion.Content = $"Итог: {\_conclusion}";

AddDialog.num = 1;

}

counter++;

}

if (counter == 0)

{

MessageBox.Show($"Товара с артикулом {ArticleInput.Text} не существует!");

return;

}

counter = 0;

conn.Close();

ArticleInput.Text = "";

}

catch (MySqlException ex)

{

MessageBox.Show($"Сбой подключения к базе данных: {ex.Message}.");}}

**ПРИЛОЖЕНИЕ Ж**

**(обязательное)**

**Листинг метода AddToDBCheck() страницы Return**

private void AddDBToCheck(object sender, EventArgs e)

{

AddDialog ad = new AddDialog();

int counter = 0;

if (ArticleInput.Text == "" || ArticleInput.Text == "0")

return;

MySqlConnection conn = new MySqlConnection(connStr);

try

{

conn.Open();

string query = $"SELECT \* FROM items WHERE articul={ArticleInput.Text}";

MySqlCommand cmd = new MySqlCommand(query, conn);

MySqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

ad.ShowDialog();

if (ad.Success)

{

double discount = int.Parse(reader[4].ToString());

discount = 1 - discount / 100;

double finalPrice = Math.Round(Double.Parse(reader[2].ToString()) \* discount, 2);

Items.Add(new Item(reader[0].ToString(), reader[1].ToString(), finalPrice, AddDialog.num));

Grids.Add(new Grid());

ItemViewer.Children.Add(Grids.Last());

var rowDefinition = new RowDefinition();

var converter = new GridLengthConverter();

var columnDefinition0 = new ColumnDefinition()

{

Width = new GridLength(1, GridUnitType.Star),

};

var columnDefinition1 = new ColumnDefinition()

{

Width = new GridLength(110),

};

var columnDefinition2 = new ColumnDefinition()

{

Width = new GridLength(40),

};

Grids.Last().RowDefinitions.Add(rowDefinition);

Grids.Last().ColumnDefinitions.Add(columnDefinition0);

Продолжение ПРИЛОЖЕНИЯ Ж

Grids.Last().ColumnDefinitions.Add(columnDefinition1); Grids.Last().ColumnDefinitions.Add(columnDefinition2);

var color = new BrushConverter();

var ItemNameTextBlock = new TextBlock()

{

Text = Items.Last().Name,

FontFamily = new FontFamily("Arial"),

Background = new SolidColorBrush(Colors.White),

Foreground = new SolidColorBrush(Colors.Black),

FontSize = 18,

TextWrapping = TextWrapping.Wrap,

Margin = new Thickness(10, 5, 0, 5),

};

ItemNameTextBlock.SetValue(TextBlock.FontWeightProperty, FontWeights.DemiBold);

var ItemPriceTextBlock = new TextBlock()

{

Text = (Items.Last().Price \* AddDialog.num).ToString(),

FontFamily = new FontFamily("Arial"),

Background = new SolidColorBrush(Colors.White),

Foreground = new SolidColorBrush(Colors.Black),

FontSize = 18,

Margin = new Thickness(10, 5, 0, 5),

TextWrapping = TextWrapping.Wrap,

};

ItemPriceTextBlock.SetValue(TextBlock.FontWeightProperty, FontWeights.DemiBold);

var ItemRemoveButton = new Button()

{

Content = "X",

FontFamily = new FontFamily("Arial"),

Background = (Brush)color.ConvertFromString("#5b7cf9"),

Foreground = new SolidColorBrush(Colors.White),

BorderBrush = new SolidColorBrush(Colors.Transparent),

FontSize = 18,

HorizontalAlignment = HorizontalAlignment.Stretch,

Окончание ПРИЛОЖЕНИЯ Ж

VerticalAlignment = VerticalAlignment.Stretch,

HorizontalContentAlignment = HorizontalAlignment.Center,

VerticalContentAlignment = VerticalAlignment.Center,

Margin = new Thickness(10, 5, 0, 5),

};

ItemRemoveButton.Click += RemoveItemFromItemViewer;

Grid.SetRow(ItemNameTextBlock, 0);

Grid.SetColumn(ItemNameTextBlock, 0);

Grids.Last().Children.Add(ItemNameTextBlock);

Grid.SetRow(ItemPriceTextBlock, 0);

Grid.SetColumn(ItemPriceTextBlock, 1);

Grids.Last().Children.Add(ItemPriceTextBlock);

Grid.SetRow(ItemRemoveButton, 0);

Grid.SetColumn(ItemRemoveButton, 2);

Grids.Last().Children.Add(ItemRemoveButton);

conclusion += Math.Round(Items.Last().Price \* Items.Last().Number, 2);

Conclusion.Content = $"Итог: {conclusion}";

AddDialog.num = 1;

}

counter++;

}

if (counter == 0)

{

MessageBox.Show($"Товара с артикулом {ArticleInput.Text} не существует!");

return;

}

counter = 0;

conn.Close();

ArticleInput.Text = "";

}

catch (MySqlException ex)

{

System.Windows.MessageBox.Show($"Сбой подключения к базе данных: {ex.Message}.");

}

}

**ПРИЛОЖЕНИЕ И**

**(обязательное)**

**Листинг метода Paid() страницы Sellings**

private void Paid(object sender, EventArgs e)

{

if (Items.Count() == 0)

{

MessageBox.Show("Операция прервана!\nТоваров к оплате предоствалено: 0.");

return;

}

CustomersCash CCash = new CustomersCash();

CCash.Owner = mw;

CCash.ShowDialog();

if (CCash.Success)

{

MySqlConnection conn = new MySqlConnection(connStr);

conn.Open();

string sql;

foreach (Item item in Items)

{

sql = "INSERT INTO sellings (shift\_id,articul,name,price,number) VALUES (@id,@articul,@name,@price,@number) ON DUPLICATE KEY UPDATE number=number+@number";

MySqlCommand cmd = new MySqlCommand(sql, conn);

cmd.Parameters.Add("@id", MySqlDbType.Int32);

cmd.Parameters["@id"].Value = mw.MyShift.Id;

cmd.Parameters.Add("@articul", MySqlDbType.VarChar);

cmd.Parameters["@articul"].Value = item.Articul;

cmd.Parameters.Add("@name", MySqlDbType.VarChar);

cmd.Parameters["@name"].Value = item.Name;

cmd.Parameters.Add("@price", MySqlDbType.Double);

cmd.Parameters["@price"].Value = item.Price;

cmd.Parameters.Add("@number", MySqlDbType.Int32);

cmd.Parameters["@number"].Value = item.Number;

if (cmd.ExecuteNonQuery() < 0)

{

MessageBox.Show($"Сбой при выполнении запроса к базе данных!");

}

sql = "UPDATE items SET number=number-@number WHERE articul=@articul";

MySqlCommand cmd2 = new MySqlCommand(sql, conn);

cmd2.Parameters.Add("@articul", MySqlDbType.VarChar);

cmd2.Parameters["@articul"].Value = item.Articul;

cmd2.Parameters.Add("@number", MySqlDbType.Int32);

cmd2.Parameters["@number"].Value = item.Number;

Окончание ПРИЛОЖЕНИЯ И

if (cmd2.ExecuteNonQuery() < 0)

{

MessageBox.Show($"Сбой при выполнении запроса к базе данных!");

}

}

conn.Close();

\_conclusion = 0;

Items.Clear();

ItemViewer.Children.Clear();

Grids.Clear();

Conclusion.Content = "Итог: 0";

if (mw.MyShift.CashBox <= 0)

{

mw.MainAction.Source = new Uri("Shifts.xaml", UriKind.Relative);

MessageBox.Show("В кассе нет денег! Для продолжения работы внесите сумму!");

}

}

}

**ПРИЛОЖЕНИЕ К**

**(обязательное)**

**Листинг метода Paid() страницы Return**

private void Paid(object sender, EventArgs e)

{

if (Items.Count() == 0)

{

MessageBox.Show("Операция прервана!\nТоваров к оплате предоствалено: 0.");

return;

}

MySqlConnection conn = new MySqlConnection(connStr);

conn.Open();

string sql;

foreach (Item item in Items)

{

sql = "INSERT INTO returns (shift\_id,articul,name,price,number) VALUES (@id,@articul,@name,@price,@number) ON DUPLICATE KEY UPDATE number=number+@number";

MySqlCommand cmd = new MySqlCommand(sql, conn);

cmd.Parameters.Add("@id", MySqlDbType.Int32);

cmd.Parameters["@id"].Value = mw.MyShift.Id;

cmd.Parameters.Add("@articul", MySqlDbType.VarChar);

cmd.Parameters["@articul"].Value = item.Articul;

cmd.Parameters.Add("@name", MySqlDbType.VarChar);

cmd.Parameters["@name"].Value = item.Name;

cmd.Parameters.Add("@price", MySqlDbType.Double);

cmd.Parameters["@price"].Value = item.Price;

cmd.Parameters.Add("@number", MySqlDbType.Int32);

cmd.Parameters["@number"].Value = item.Number;

if (cmd.ExecuteNonQuery() < 0)

{

MessageBox.Show($"Сбой при выполнении запроса к базе данных!");

}

sql = "UPDATE items SET number=number+@number WHERE articul=@articul";

MySqlCommand cmd2 = new MySqlCommand(sql, conn);

cmd2.Parameters.Add("@articul", MySqlDbType.VarChar);

cmd2.Parameters["@articul"].Value = item.Articul;

cmd2.Parameters.Add("@number", MySqlDbType.Int32);

cmd2.Parameters["@number"].Value = item.Number;

if (cmd2.ExecuteNonQuery() < 0)

{

MessageBox.Show($"Сбой при выполнении запроса к базе данных!");

}

Окончание ПРИЛОЖЕНИЯ К

}

conn.Close();

MessageBox.Show($"Товара возвращено на {conclusion}.");

mw.MyShift.CashReturned += conclusion;

mw.MyShift.CashBox -= conclusion;

conclusion = 0;

Items.Clear();

ItemViewer.Children.Clear();

Grids.Clear();

Conclusion.Content = "Итог: 0";

if (mw.MyShift.CashBox <= 0)

{

mw.MainAction.Source = new Uri("Shifts.xaml", UriKind.Relative);

MessageBox.Show("В кассе нет денег! Для продолжения работы внесите сумму!");

}