**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc131443371)

[1 Постановка задачи 6](#_Toc131443372)

[1.1 Анализ предметной области 6](#_Toc131443373)

[1.2 Техническое задание на разработку ГОСТ 19.201-78 9](#_Toc131443374)

[1.2.1 Введение 9](#_Toc131443375)

[1.2.1.1 Наименование программы 9](#_Toc131443376)

[1.2.1.2 Краткая характеристика области применения 9](#_Toc131443377)

[1.2.2 Основания для разработки 9](#_Toc131443378)

[1.2.2.1 Основание для проведения разработки 9](#_Toc131443379)

[1.2.2.2 Наименование и условное обозначение темы разработки 9](#_Toc131443380)

[1.2.3 Назначение разработки 9](#_Toc131443381)

[1.2.3.1 Функциональное назначение 9](#_Toc131443382)

[1.2.3.2 Эксплуатационное назначение 9](#_Toc131443383)

[1.2.4 Требования к программе или программному изделию 9](#_Toc131443384)

[1.2.4.1 Требования к функциональным характеристикам 9](#_Toc131443385)

[1.2.4.2 Время восстановления после отказа 10](#_Toc131443386)

[1.2.4.3 Отказы из-за некорректных действий оператора 10](#_Toc131443387)

[1.2.4.4 Условия эксплуатации 10](#_Toc131443388)

[1.2.4.5 Требования к численности и квалификации персонала 10](#_Toc131443389)

[1.2.4.6 Требования к составу и параметрам технических средств 11](#_Toc131443390)

[1.2.4.7 Требования к информационной и программной совместимости 11](#_Toc131443391)

[1.2.4.8 Требования к маркировке и упаковке 11](#_Toc131443392)

[1.2.4.9 Требования к программной документации 12](#_Toc131443393)

[1.2.5 Стадии и этапы разработки 12](#_Toc131443394)

[1.2.5.1 Стадии разработки 12](#_Toc131443395)

[1.2.5.2 Этапы разработки 12](#_Toc131443396)

[1.2.5.3 Содержание работ по этапам 12](#_Toc131443397)

[1.2.6 Порядок контроля и приёмки 13](#_Toc131443398)

[1.2.6.1 Виды испытаний 13](#_Toc131443399)

[2 Проектирование 14](#_Toc131443400)

[2.1 Функциональная структура программы 14](#_Toc131443401)

[2.2 Схема модулей программы 16](#_Toc131443402)

[2.3 Паспорта основных модулей 17](#_Toc131443403)

[2.4 Интеграция модуля «Расчет скидки» 23](#_Toc131443404)

[2.5 Инспектирование компонент ПО на стандарты кодирования 24](#_Toc131443405)

[3 Тестирование программы 25](#_Toc131443406)

[3.1 Программа, методика и результаты испытаний ГОСТ 19.301-79 25](#_Toc131443407)

[3.1.1 Объект испытаний 25](#_Toc131443410)

[3.1.1.1 Наименование программы 25](#_Toc131443411)

[3.1.1.2 Назначение разработки 25](#_Toc131443412)

[3.1.2 Цель испытаний 25](#_Toc131443413)

[3.1.2.1 Основания для проведения испытаний 25](#_Toc131443414)

[3.1.2.2 Место и продолжительность испытаний 25](#_Toc131443415)

[3.1.2.3 Перечень документов, предъявляемых на испытания 25](#_Toc131443416)

[3.1.2.4 Объём испытаний 25](#_Toc131443417)

[3.1.3 Требования к программе 26](#_Toc131443418)

[3.1.4 Требования к программной документации 26](#_Toc131443419)

[3.1.5 Средства и порядок испытаний 26](#_Toc131443420)

[3.1.5.1 Технические средства, используемые во время испытаний 26](#_Toc131443421)

[3.1.5.2 Условия и порядок проведения испытаний 27](#_Toc131443422)

[3.1.6 Методика испытаний 27](#_Toc131443423)

[3.2 Протокол испытаний 28](#_Toc131443424)

[3.3 Тестирование при помощи Unit-Test 30](#_Toc131443430)

[ЗАКЛЮЧЕНИЕ 33](#_Toc131443431)

[ПРИЛОЖЕНИЕ А 34](#_Toc131443432)

[ПРИЛОЖЕНИЕ Б 90](#_Toc131443433)

# **ВВЕДЕНИЕ**

Данная практическая работа включает в себя разработку программы «Кафе-столовая», которая будет взаимодействовать с базой SQL Server для учета товаров.

Для реализации поставленной задачи были выбраны следующие средства разработки: MS SQL Server 2019 и Microsoft Visual Studio 2022 на языке программирования C#.

# **Постановка задачи**

## **Анализ предметной области**

Целью практической работы является проектирование базы данных предметной области «Кафе-столовая» и разработка программы для работы. В ней должна содержаться информация о товаре, а также должна быть реализована возможность вывода чеков.

В БД должны храниться сущности и атрибуты данной темы:

* Сущность **Товары** содержит атрибуты: *Код товара* – является ключом, Наименование, Цена, Остатки, Код единицы измерения;
* Сущность **Единицы измерения** содержит атрибуты: *Код единицы измерения* – является ключом, Наименование;
* Сущность **Чеки** содержит атрибуты: *Код чека* – является ключом, Стоимость, Скидка, Стоимость со скидкой, Комплексный обед;
* Сущность **Покупки** содержит атрибуты: *Код покупки* – является ключом, Код чека, Код товара, Количество товара.

В данном практическом задании используются следующие ограничения данных в таблицах:

1. Таблица Товары

Поля Код товара, Наименование, Цена, Остатки, Код единицы измерения являются обязательными.

В поле **Наименование** используетсяограничение по количеству символов – 50.

1. Таблица Единицы измерения

Поля Код единицы измерения, Наименование являются обязательными.

В поле **Наименование** используетсяограничение по количеству символов – 50.

1. Таблица Чеки

Поля Код чека, Стоимость, Скидка, Стоимость со скидкой, Комплексный обед являются обязательными.

В поле **Наименование** используетсяограничение по количеству символов – 50.

1. Таблица Покупки

Поля Код покупки, Код чека, Код товара, Количество товара являются обязательными.

На основании вышеуказанного у нас получается следующая диаграмма, как показано на рисунке 1.



**Рисунок 1** – Расширенная ЕR- диаграмма

Ниже приведены таблицы, которые характеризуют каждую сущность БД по нашей теме:

Таблица 1 – Товары (Products)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Имя столбца** | **Содержательное описание** | **Тип данных** | **Размерность** | **Область допустимых значений** | **Возможность значения Null** | **Роль** | **Пример** |
| ProductID | Код товара | int | 4 |  | Нет | PK | 25 |
| Name | Наименование | nvarchar(50) | 100 |  | Нет |  | Борщ |
| Price | Цена | money | 16 |  | Нет |  | 100,00 |
| HoldCount | Остатки | int | 4 |  | Нет |  | 14 |
| MeasureID | Код единицы измерения | int | 4 |  | Нет | FK | 1 |

Таблица 2 – Единицы измерения (Measures)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Имя столбца** | **Содержательное описание** | **Тип данных** | **Размерность** | **Область допустимых значений** | **Возможность значения Null** | **Роль** | **Пример** |
| MeasureID | Код единицы измерения | int | 4 |  | Нет | PK | 1 |
| Name | Наименование | nvarchar(50) | 100 |  | Нет |  | шт |

Таблица 3 – Чеки (Checks)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Имя столбца** | **Содержательное описание** | **Тип данных** | **Размерность** | **Область допустимых значений** | **Возможность значения Null** | **Роль** | **Пример** |
| CheckID | Код чека | int | 4 |  | Нет | PK | 25 |
| Cost | Стоимость | money | 16 |  | Нет |  | 100,00 |
| Discount | Скидка | money | 16 |  | Нет |  | 100,00 |
| CostNDiscount | Стоимость со скидкой | money | 16 |  | Нет |  | 100,00 |
| IsComplexLunch | Комплексный обед | bit | 1 |  | Нет |  | true |

Таблица 4 – Покупки (Purchases)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Имя столбца** | **Содержательное описание** | **Тип данных** | **Размерность** | **Область допустимых значений** | **Возможность значения Null** | **Роль** | **Пример** |
| PurchaseID | Код покупки | int | 4 |  | Нет | PK | 12 |
| CheckID | Код чека | int | 4 |  | Нет | FK | 2 |
| ProductID | Код товара | int | 4 |  | Нет | FK | 3 |
| ProductCount | Количество товара | int | 4 |  | Нет |  | 2 |

## **Техническое задание на разработку ГОСТ 19.201-78**

# **Введение**

# **Наименование программы**

Наименование программы «Кафе-столовая».

# **Краткая характеристика области применения**

Программа предназначена для ведения информации о товаре и его продаже.

# **Основания для разработки**

# **Основание для проведения разработки**

Основанием для проведения разработки является задание на практику.

# **Наименование и условное обозначение темы разработки**

Наименование темы разработки – «Кафе-столовая».

# **Назначение разработки**

# **Функциональное назначение**

Ведение информации о товаре и его продаже.

# **Эксплуатационное назначение**

Программа предназначена для обработки данных.

# **Требования к программе или программному изделию**

# **Требования к функциональным характеристикам**

Программа должна обеспечивать возможность выполнения следующих функций:

1. Добавлять, редактировать, удалять информацию о товаре;
2. Добавлять, редактировать, удалять информацию о единицах измерения;
3. Добавлять, редактировать, удалять информацию о чеках;
4. Система скидок для общей стоимости;
5. Добавлять, редактировать, удалять информацию о покупках, а также реализовать для них проверку наличия на остатках товара и его списание.

# **Время восстановления после отказа**

Время восстановления после отказа, вызванного своем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 20 минут при условии соблюдения эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

# **Отказы из-за некорректных действий оператора**

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

# **Условия эксплуатации**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части их эксплуатации.

# **Требования к численности и квалификации персонала**

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц — системный администратор и пользователь программы - оператор.

Пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

# **Требования к составу и параметрам технических средств**

В состав технических средств должен входить IBM-совместимый персональный компьютер, включающий в себя:

Процессор: 2 ядра 1,8 ГГц и более;

ОЗУ: 2Гб, рекомендуется 4Гб и более;

Необходимо место на жёстком диске: 2ГБ и более.

# **Требования к информационной и программной совместимости**

Информационная система должна включать отображение входных данных и результата на экране.

Исходные коды программы должны быть реализованы на языке C#. В качестве интегрированной среды разработки программы должна быть использована среда Microsoft Visual Studio 2022 и Microsoft SQL Server 2019.

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 7, 8, 10 с платформой .NET Framework 4.7.2.

Требования к защите информации и программ не предъявляются.

Программа должна обеспечивать взаимодействие с пользователем (оператором) посредством графического пользовательского интерфейса, разработанного согласно рекомендациям лекции.

# **Требования к маркировке и упаковке**

Программное изделие должно иметь маркировку с обозначением фамилии студента, номера группы, наименования, номера версии. Маркировка должна быть нанесена на программное изделие в виде надписи.

Упаковка программного изделия должна осуществляться в конверт для дисков.

# **Требования к программной документации**

Состав программной документации должен включать в себя:

1. Техническое задание на разработку ГОСТ 19.201-78;
2. Программа, методика и результаты испытаний ГОСТ 19.301-79.

# **Стадии и этапы разработки**

# **Стадии разработки**

Разработка должна быть проведена в две стадии:

1. Техническое задание;
2. Технический (и рабочий) проекты.

# **Этапы разработки**

1. Разработка технического задания;
2. Разработка программы;
3. Разработка программной документации;
4. Испытания программы.

# **Содержание работ по этапам**

Таблица 5 – Содержание работ по этапам

|  |  |
| --- | --- |
| **№ п/п** | **Наименование работ** |
| 1. | Анализ предметной области данной задачи |
| 2. | Разработка технического задания на разработку ГОСТ 19.201-78 |
| 3. | Разработка алгоритмов основных действий |
| 4. | Разработка модулей программы согласно спецификации и интерфейса программы |
| 5. | Интеграция модуля (расчет скидки) |
| 6. | Разработка документации для программы испытаний |
| 7. | Тестирование программы |
| 8. | Отладка программы |
| 9. | Трассировка одного из модулей |
| 10. | Инспектирование компонент ПО на предмет соответствия стандартам кодирования |

# **Порядок контроля и приёмки**

# **Виды испытаний**

Приёмосдаточные испытания должны проводиться на объекте заказчика в сроки выполнения для практического задания.

Приёмосдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованной заказчиком «Программы и методики испытаний».

# **Проектирование**

## **Функциональная структура программы**

Описываются функции, которые будет выполнять программа. Согласно технологии структурного программирования, любая составная функция может, и должна быть разложена на элементарные для данного уровня функции.

Ниже приведена таблица, в которую включаются все возможные функции, которые планируются использовать в программе в соответствии с таблицей 6.

Таблица 6 – Таблица спецификации функций

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Действие** | **Объект** | **Вид** |
| **Работа с главной страницей** | | | |
| Открытие страницы «Товары» | Открывает страницу со списком товаров | Страница | Основная |
| Открытие страницы «Единицы измерения» | Открывает страницу со списком единиц измерения | Страница | Основная |
| Открытие страницы «Чеки» | Открывает страницу со списком чеков и списком соответствующих чеку покупок | Страница | Основная |
| Отображение сообщения «О программе» | Показывает сообщение, где написаны подробности о программе | Страница | Дополнительная |
| Выход | Закрывает приложение | Приложение | Дополнительная |
| **Работа со страницей «Товары»** | | | |
| Добавление новой записи в таблицу товаров | Добавляет запись | Таблица | Основная |
| Изменение выбранной записи в таблице товаров | Редактирует выбранную запись | Таблица | Основная |
| Удаление выбранной записи в таблице товаров | Удаляет выбранную запись | Таблица | Основная |
| Поиск необходимой записи в таблице товаров | Ищет нужную запись по критерию поиска | Таблица | Основная |
| Обновление данных в таблице товаров | Обновляет данные в таблице, получая их от сервера | Таблица | Дополнительная |
| Вернуться на главную | Возвращает пользователя на главную страницу | Страница | Основная |
| **Работа со страницей «Единицы измерения»** | | | |
| Добавление новой записи в таблицу единиц измерения | Добавляет запись | Таблица | Основная |
| Изменение выбранной записи в таблице единиц измерения | Редактирует выбранную запись | Таблица | Основная |
| Удаление выбранной записи в таблице единиц измерения | Удаляет выбранную запись | Таблица | Основная |
| Поиск необходимой записи в таблице единиц измерения | Ищет нужную запись по критерию поиска | Таблица | Основная |
| Обновление данных в таблице единиц измерения | Обновляет данные в таблице, получая их от сервера | Таблица | Дополнительная |
| Вернуться на главную | Возвращает пользователя на главную страницу | Страница | Основная |
| **Работа со страницей «Чеки»** | | | |
| Добавление новой записи в таблицу чеков и покупок | Добавляет запись | Таблицы | Основная |
| Изменение выбранной записи в таблице чеков и покупок | Редактирует выбранную запись | Таблицы | Основная |
| Удаление выбранной записи в таблице чеков и покупок | Удаляет выбранную запись | Таблицы | Основная |
| Поиск необходимой записи в таблице чеков | Ищет нужную запись по критерию поиска | Таблица | Основная |
| Обновление данных в таблице чеков и покупок | Обновляет данные в таблице, получая их от сервера | Таблицы | Дополнительная |
| Вернуться на главную | Возвращает пользователя на главную страницу | Страница | Основная |

## **Схема модулей программы**

Структура программы в соответствии с рисунком 2.



**Рисунок 2** – Структура программы

## **Паспорта основных модулей**

У всех страниц общий алгоритм выполнения функций добавления, редактирования и удаления, поиска или обновления списка. Единственное отличие – при добавлении покупки в чеке, происходит проверка остатков товара и его списание нужного количества. Ниже приведен перечень основных модулей, фигурирующий в работе с данными. Структура данных для добавления, удаления и изменения повторяется. Они будут отмечены в одной таблице.

Таблица 7 – Структура данных функции «Добавить/изменить/удалить товар»

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Описание** |
| Product (общее имя) | Products | Экземпляр объекта типа Products. |
| ProductID | int | Свойство первичного ключа сущности в коде (для товара). |
| Name | string | Свойство имени в коде для товара. |
| Price | decimal | Свойство цены товара. |
| HoldCount | int | Свойство остатков товара. |
| MeasureID | int | Свойство кода единицы измерения. |

Таблица 8 – Структура данных функции «Добавить/изменить/удалить единицу измерения»

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Описание** |
| Measure (общее имя) | Measures | Экземпляр объекта типа Measures. |
| MeasureID | int | Свойство первичного ключа сущности в коде  (для единицы измерения). |
| Name | string | Свойство имени в коде для единицы измерения. |

Таблица 9 – Структура данных функции «Добавить/изменить/удалить чек и список покупок по нему»

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Описание** |
| Check (общее имя) | Checks | Экземпляр объекта типа Checks. |
| CheckID | int | Свойство первичного ключа сущности в коде (для чеков).  Также является свойством внешнего ключа сущности в коде (для покупок). |
| Cost | decimal | Свойство стоимости всех товаров в чеке. |
| Discount | decimal | Свойство скидки относительно стоимости товаров в чеке. |
| CostNDiscount | decimal | Свойство стоимости со скидкой. |
| IsComplexLunch | bool | Свойство, отвечающее за идентификацию комплексного обеда, который не должен быть включен в систему скидок. |
| Purchase (общее имя) | Purchases | Экземпляр объекта типа Purchases. |
| PurchaseID | int | Свойство первичного ключа сущности в коде (для покупок). |
| ProductID | int | Свойство кода товара, являющееся внешним ключом для идентификации, какому товару соответствует покупка. |
| ProductCount | int | Свойство количества купленного товара. |

Алгоритм работы со страницей «Добавление записи» изображен на рисунке 3.



**Рисунок 3** – Алгоритм функции «Добавление записи»

Алгоритм работы с функцией «Изменение выбранной записи» показан на рисунке 4.



**Рисунок 4** – Алгоритм функции «Изменение выбранной записи»

Алгоритм работы «Удаление выбранной записи» приведен на рисунке 5.



**Рисунок 5** – Алгоритм функции «Удаление выбранной записи»

Реализация поиска выполнена в автоматизированном порядке (автоматически ищет относительно введенных данных без нажатия кнопок). Поиск по коду и имени ведется в таблицах «Товары» и «Единицы измерения». По коду и итоговой сумме ведется в «Чеки». Алгоритм работы «Поиска» показан на рисунке 6.



**Рисунок 6** – Алгоритм функции «Поиск»

## **Интеграция модуля «Расчет скидки»**

Был интегрирован модуль для ускорения работы сотрудника кафе-столовой, который рассчитывает скидку, если покупаемое не является комплексным обедом. Скидка рассчитывается так, что от 300 до 500 руб. 3%, от 501 до 1000 руб. 5%, от 1001 до 5000 руб. 7%, от 5001 и выше – 10%.

Данный модуль позволил ускорить работу с программой. Сотруднику не нужно вручную вести расчет скидки. Продуктивность работы возросла.

## **Инспектирование компонент ПО на стандарты кодирования**

Процесс инспектирования на стандарты кодирования прошел успешно. Программа написана в соответствии с ними. Код программы можно увидеть в п. «ПРИЛОЖЕНИЕ А» для сравнения. Сами стандарты описаны на сайте Майкрософт:

https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/coding-style/coding-conventions

# **Тестирование программы**

# **Программа, методика и результаты испытаний ГОСТ 19.301-79**



# **Объект испытаний**

# **Наименование программы**

Наименование программы «Кафе-столовая».

# **Назначение разработки**

Программа предназначена для обработки данных.

# **Цель испытаний**

# **Основания для проведения испытаний**

Основанием для проведения испытаний является задание на практику.

# **Место и продолжительность испытаний**

Место испытаний: дом, колледж. Продолжительность 2 часа.

# **Перечень документов, предъявляемых на испытания**

Состав программной документации должен включать в себя:

1. Техническое задание;
2. Программа, методика и результаты испытаний.

# **Объём испытаний**

Испытания проводятся в два этапа:

1. Ознакомительный;
2. Испытания.
   * + - 1. **Перечень проверок, проводимых на 1 этапе испытаний**

Перечень проверок, проводимых на 1-м этапе испытаний, должен включать в себя:

1. Проверку комплектности программной документации;
2. Проверку комплектности состава технических и программных средств.
   * + - 1. **Перечень проверок, проводимых на 2 этапе испытаний**

Перечень проверок, проводимых на 2-м этапе испытаний, должен включать в себя:

1. Проверку соответствия технических характеристик программы;
2. Проверку степени выполнения требований функционального назначения программы.
   * + - 1. **Количественные и качественные характеристики, подлежащие оценке**

# **Требования к программе**

# **Требования к программной документации**

Состав программной документации должен включать в себя:

1. Техническое задание;
2. Программа, методика и результаты испытаний.

# **Средства и порядок испытаний**

# **Технические средства, используемые во время испытаний**

В состав технических средств должен входить IBM-совместимый персональный компьютер, включающий в себя:

Процессор: 2 ядра 1,8 ГГц и более;

ОЗУ: 2Гб, рекомендуется 4Гб и более;

Необходимо место на жёстком диске: 2ГБ и более.

* + - * 1. **Порядок проведения испытаний**

Испытания должны проводиться поэтапно согласно п. «Перечень этапов испытаний» настоящего документа.

# **Условия и порядок проведения испытаний**

* + - * 1. **Условия начала и завершения отдельных этапов испытаний**

Необходимым и достаточным условием завершения 1 этапа испытаний и начала 2 этапа испытаний является успешное завершение проверок, проводимых на 1 этапе (см. п. «Перечень проверок, проводимых на 1 этапе испытаний»).

Условием завершения 2 этапа испытаний является успешное завершение проверок, проводимых на 2 этапе испытаний.

# **Методика испытаний**

**Методы проведения проверки комплектности программной документации.**

Проверка комплектности программной документации на программное изделие производится визуально преподавателем. В ходе проверки сопоставляется состав и комплектность программной документации, представленной исполнителем, с перечнем программной документации.

Проверка считается завершенной в случае соответствия состава и комплектности программной документации, представленной исполнителем, перечню программной документации, приведенному в указанном вышек пункте.

**Методы проведения проверки степени выполнения требований функционального назначения программы.**

Тестирование серого ящика или поведенческое тестирование – стратегия (метод) тестирования функционального проведения программы с точки зрения внешнего мира, при котором используется знание о внутреннем устройстве тестируемого объекта. Под методом понимаются систематические методы отбора и создания тестов для тестового набора. Стратегия поведенческого теста исходит из технических требований и их спецификаций.

# **Протокол испытаний**

Результаты предварительного тестирования представлены в таблице 10.

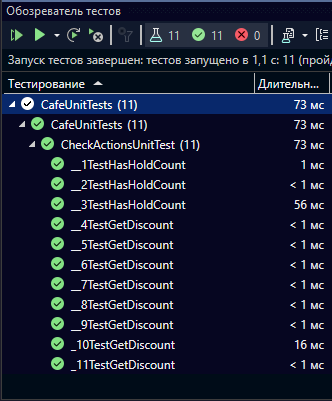
Таблица 10 – Тестирование «Серого ящика»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Разработано:** Лопаткин Сергей | | | | **Начальные установки:** нет | | |
| **Цели:** Проверки степени выполнения требований функционального назначений программы. | | | | | | |
| **Номер теста** | **Шаги теста** | **Ожидаемый результат** | **Полученный результат** | | **Результат тестирования (Успешно/не успешно)** | **Тестер/Дата** |
| **Тестирование функций главной страницы** | | | | | | |
| 1 | Нажатие кнопки «Товары» | Открытие страницы «Товары» | Открытие страницы «Товары» | | Успешно | Лопаткин С.М. 03.04.2023 |
| 2 | Нажатие кнопки «Единицы измерения» | Открытие страницы «Единицы измерения» | Открытие страницы «Единицы измерения» | | Успешно | Лопаткин С.М. 03.04.2023 |
| 3 | Нажатие кнопки «Чеки» | Открытие страницы «Чеки» | Открытие страницы «Чеки» | | Успешно | Лопаткин С.М. 03.04.2023 |
| 4 | Нажатие кнопки «О программе» | Сообщение с подробной информацией о программе | Сообщение с подробной информацией о программе | | Успешно | Лопаткин С.М. 03.04.2023 |
| 5 | Нажатие кнопки «Закрытие программы» | Закрытие программы | Закрытие программы | | Успешно | Лопаткин С.М. 03.04.2023 |
| **Тестирование функций страницы «Товара»** | | | | | | |
| 6 | Нажатие кнопки «Добавить» | Добавление записи | Добавление записи | | Успешно | Лопаткин С.М. 03.04.2023 |
| 7 | Нажатие кнопки «Изменить» | Изменение информации выбранной записи | Изменение информации выбранной строки | | Успешно | Лопаткин С.М. 03.04.2023 |
| 8 | Нажатие кнопки «Удалить» | Удаление выбранной записи | Удаление выбранной строки | | Успешно | Лопаткин С.М. 03.04.2023 |
| 9 | Нажатие кнопки «Обновить» | Получение данных из базы данных | Получение данных из базы данных | | Успешно | Лопаткин С.М. 03.04.2023 |
| 10 | Ввод в поле поиска кода или имени | Выделение в списке найденной записи, соответствующей запросу | Выделение в списке найденной записи, соответствующей запросу | | Успешно | Лопаткин С.М. 03.04.2023 |
| 11 | Нажатие кнопки «Назад» | Возврат на главную страницу | Возврат на главную страницу | | Успешно | Лопаткин С.М. 03.04.2023 |
| **Тестирование функций страницы «Единицы измерения»** | | | | | | |
| 12 | Нажатие кнопки «Добавить» | Добавление записи | Добавление записи | | Успешно | Лопаткин С.М. 03.04.2023 |
| 13 | Нажатие кнопки «Изменить» | Изменение информации выбранной записи | Изменение информации выбранной строки | | Успешно | Лопаткин С.М. 03.04.2023 |
| 14 | Нажатие кнопки «Удалить» | Удаление выбранной записи | Удаление выбранной строки | | Успешно | Лопаткин С.М. 03.04.2023 |
| 15 | Нажатие кнопки «Обновить» | Получение данных из базы данных | Получение данных из базы данных | | Успешно | Лопаткин С.М. 03.04.2023 |
| 16 | Ввод в поле поиска кода или имени | Выделение в списке найденной записи, соответствующей запросу | Выделение в списке найденной записи, соответствующей запросу | | Успешно | Лопаткин С.М. 03.04.2023 |
| 17 | Нажатие кнопки «Назад» | Возврат на главную страницу | Возврат на главную страницу | | Успешно | Лопаткин С.М. 03.04.2023 |
| **Тестирование функций страницы «Чеки»** | | | | | | |
| 26 | Нажатие кнопки «Добавить» | Добавление записи | Добавление записи | | Успешно | Лопаткин С.М. 03.04.2023 |
| 27 | Нажатие кнопки «Изменить» | Изменение информации выбранной записи | Изменение информации выбранной строки | | Успешно | Лопаткин С.М. 03.04.2023 |
| 28 | Нажатие кнопки «Удалить» | Удаление выбранной записи | Удаление выбранной строки | | Успешно | Лопаткин С.М. 03.04.2023 |
| 29 | Нажатие кнопки «Обновить» | Получение данных из базы данных | Получение данных из базы данных | | Успешно | Лопаткин С.М. 03.04.2023 |
| 16 | Ввод в поле поиска кода или имени | Выделение в списке найденной записи, соответствующей запросу | Выделение в списке найденной записи, соответствующей запросу | | Успешно | Лопаткин С.М. 03.04.2023 |
| 31 | Нажатие кнопки «Назад» | Возврат на главную страницу | Возврат на главную страницу | | Успешно | Лопаткин С.М. 03.04.2023 |
| 32 | Расчет скидки относительно общей стоимости добавленных продуктов (любых) и условия комплексного обеда | Верный расчет скидки | Верный расчет скидки | | Успешно | Лопаткин С.М. 03.04.2023 |
|  |  |  |  | |  |  |

3. 3. **Тестирование при помощи Unit-Test**

Для того, чтобы убедиться в правильности отображаемых данных, разработаем тестовые модули для расчета скидки и проверки наличия на остатках товара для дальнейшего его списания. Проверяются граничные значения условий, чтобы подтвердить их работоспособность в точности.

Результаты тестирования изображены на рисунке 7. Все успешно. Работает корректно.



**Рисунок 7** – Юнит-тестирование

Листинг кода юнит-тестов можно посмотреть в п. «ПРИЛОЖЕНИЕ Б».

**Протоколы тестирования методов класса CheckActions**

Таблица 11 - Тестирование при помощи Unit-Test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тестер и разработчик:** Лопаткин Сергей | | | | | | |
| **Unit Test №** | **Назначение** | **Начальные значения** | **Ожидаемый результат** | **Фактический результат** | **Успешность** |
| 1 | Проверка работы спуска до 0 остатков | HoldCount = 2, count = 2 | true | true | Не успешно |
| 2 | Проверка работы с остатками больше 0 | HoldCount = 3, count = 2 | true | true | Не успешно |
| 3 | Проверка работы с остатками меньше 0 | HoldCount = 1, count = 2 | false | false | Не успешно |
| 4 | Проверка работы скидки до 300 р. | cost = 299 | 0 | 0 | Не успешно |
| 5 | Проверка работы скидки от 300 р. | cost = 300 | 9 | 9 | Не успешно |
| 6 | Проверка работы скидки до 500 р. | cost = 500 | 15 | 15 | Не успешно |
| 7 | Проверка работы скидки от 501 р. | cost = 501 | 25.05 | 25.05 | Не успешно |
| 8 | Проверка работы скидки до 1000 р. | cost = 1000 | 50 | 50 | Не успешно |
| 9 | Проверка работы скидки от 1001 р. | cost = 1001 | 70.07 | 70.07 | Не успешно |
| 10 | Проверка работы скидки до 5000 р. | cost = 5000 | 350 | 350 | Не успешно |
| 11 | Проверка работы скидки от 5001 р. | cost = 5001 | 500.1 | 500.1 | Не успешно |

# **ЗАКЛЮЧЕНИЕ**

В данном практической работе была реализована база данных и программа для ведения учета товаров в кафе-столовой. Можно взаимодействовать с данными в соответствии с заданием. Имеется данный перечень возможностей:

1. Добавлять, редактировать, удалять информацию о товаре;
2. Добавлять, редактировать, удалять информацию о единицах измерения;
3. Добавлять, редактировать, удалять информацию о чеках;
4. Система скидок для общей стоимости;
5. Добавлять, редактировать, удалять информацию о покупках, а также реализовать для них проверку наличия на остатках товара и его списание.

# **ПРИЛОЖЕНИЕ А**

**Листинг основы**

**Листинг главного окна**

<Window x:Class="Cafe.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Cafe"

mc:Ignorable="d"

Title="Кафе-столовая"

MinHeight="450"

MinWidth="800"

WindowStartupLocation="CenterScreen"

ResizeMode="CanResizeWithGrip"

Background="Black" Closing="Window\_Closing">

<DockPanel>

<ContentPresenter x:Name="MainScreen"/>

</DockPanel>

</Window>

**Листинг кода главного окна**

using Cafe.Tools;

using Cafe.View;

using System.Windows;

namespace Cafe

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

AppMainWindow = this;

MainScreen.Content = new MainView();

DBContext.UpdateContext();

}

public static MainWindow AppMainWindow;

private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)

{

e.Cancel = NotificationActions.WindowClosing();

}

}

}

**Листинг отображения MainView**

<UserControl x:Class="Cafe.View.MainView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Cafe.View"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<UserControl.Background>

<ImageBrush ImageSource="/Images/Cafe\_Background.jpg" Stretch="UniformToFill"

Opacity="0.5"/>

</UserControl.Background>

<Grid Margin="15">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Width="200">

<Image Source="/Images/Products.png"

Height="75"/>

<Button x:Name="Products"

Content="Товары"

Margin="5,15"

Height="30" Click="Products\_Click"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Width="200"

Grid.Column="1">

<Image Source="/Images/Checks.png"

Height="75"/>

<Button x:Name="Checks"

Content="Чеки"

Margin="5,15"

Height="30" Click="Checks\_Click"/>

</StackPanel>

</Grid>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Width="275"

Grid.Row="1">

<Image Source="/Images/Measures.png"

Height="75"/>

<Button x:Name="Measures"

Content="Единицы измерения"

Margin="5,15"

Height="30"

Width="200" Click="Measures\_Click"/>

</StackPanel>

<Button x:Name="CloseWindow"

Background="Transparent"

Opacity="1"

Height="42"

Width="42"

BorderThickness="0"

HorizontalAlignment="Left"

VerticalAlignment="Bottom"

Grid.Row="1"

ToolTip="Отображает информацию о программе"

Click="CloseWindow\_Click"

IsCancel="True"

Style="{StaticResource ImageButton}">

<Button.Effect>

<DropShadowEffect BlurRadius="50"

Direction="90"

ShadowDepth="-25"/>

</Button.Effect>

<Image Source="/Images/Close.png"/>

</Button>

<Button x:Name="AboutProgram"

Background="Transparent"

Opacity="1"

Height="42"

Width="42"

BorderThickness="0"

HorizontalAlignment="Right"

VerticalAlignment="Bottom"

Grid.Row="1"

ToolTip="Отображает информацию о программе"

Click="AboutProgram\_Click"

Style="{StaticResource ImageButton}">

<Button.Effect>

<DropShadowEffect BlurRadius="50"

Direction="90"

ShadowDepth="-25"/>

</Button.Effect>

<Image Source="/Images/Developer.png"/>

</Button>

</Grid>

</UserControl>

**Листинг кода отображения MainView**

using System.Windows;

using System.Windows.Controls;

namespace Cafe.View

{

/// <summary>

/// Логика взаимодействия для MainView.xaml

/// </summary>

public partial class MainView : UserControl

{

public MainView()

{

InitializeComponent();

}

private void Products\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new ProductsViews.ProductsView();

}

private void Checks\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new ChecksViews.ChecksView();

}

private void Measures\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new MeasuresViews.MeasuresView();

}

private void AboutProgram\_Click(object sender, RoutedEventArgs e)

{

MessageBox.Show("Разработчик программы: Лопаткин Сергей Михайлович (GitHub.Name = \"JohnHaval\")\n\n" +

"Наименование программы: Кафе-столовая\n" +

"Кодовое наименование: Cafe\n" +

"Версия: 1.0", "О программе", MessageBoxButton.OK, MessageBoxImage.Information);

}

private void CloseWindow\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.Close();

}

}

}

**Листинг отображения ProductsView**

<UserControl x:Class="Cafe.View.ProductsViews.ProductsView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<UserControl.Background>

<ImageBrush ImageSource="/Images/Cafe\_Background.jpg" Stretch="UniformToFill"

Opacity="0.5"/>

</UserControl.Background>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="34\*" MaxHeight="88"/>

<RowDefinition Height="44"/>

<RowDefinition Height="30"/>

<RowDefinition Height="126\*"/>

<RowDefinition Height="44"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="96"/>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="406\*"/>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="96"/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Grid.Column="2">

<Image Source="/Images/Products.png"

Height="40"

Width="40" Margin="10"/>

<TextBlock Text="Товары"

FontSize="30"

Height="36"

FontFamily="SF Pro Display"

FontWeight="SemiBold">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

</StackPanel>

<TextBlock Text="Искать (№, Имя)"

FontSize="16"

Grid.Row="1"

Grid.ColumnSpan="2"

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontFamily="SF Pro Display"

FontWeight="SemiBold">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="15"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="Search"

Grid.Row="1"

Grid.ColumnSpan="2"

Grid.Column="2"

Opacity="0.9"

Margin="0,7,0,7"

TextChanged="Search\_TextChanged"

MaxLength="50"/>

<Border Grid.ColumnSpan="5"

Grid.Row="2"

Background="AliceBlue"

Opacity="0.9"

BorderThickness="1"

BorderBrush="Gray">

<DockPanel>

<TextBlock Text="№"

HorizontalAlignment="Center"

Width="120"

TextAlignment="Center"

VerticalAlignment="Center"/>

<Line X1="0" Y1="0" X2="0" Y2="95" Stroke="Black"/>

<TextBlock Text="Товар"

HorizontalAlignment="Center"

TextAlignment="Center"

VerticalAlignment="Center"/>

</DockPanel>

</Border>

<ListView x:Name="ProductsList"

Grid.Row="3"

Grid.ColumnSpan="5"

Opacity="0.9"

Grid.RowSpan="1"

Background="AliceBlue"

SelectionMode="Single">

<ListView.ItemTemplate>

<DataTemplate>

<DockPanel>

<TextBlock Text="{Binding ProductID}"

Width="104"

TextAlignment="Center"

VerticalAlignment="Center"></TextBlock>

<StackPanel>

<TextBlock Text="{Binding Name}"

Margin="15,0"

FontWeight="Bold"

FontSize="18"></TextBlock>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Left"

Margin="15,0">

<TextBlock FontSize="16">Остатки:</TextBlock>

<TextBlock Text="{Binding HoldCount}"

FontSize="16"

Margin="5,0"></TextBlock>

<TextBlock Text="{Binding Measures.Name}"

FontSize="16"></TextBlock>

</StackPanel>

</StackPanel>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Right">

<TextBlock Text="{Binding Price, StringFormat=F2}"

FontSize="18"></TextBlock>

<TextBlock Margin="5,0"

FontSize="18">руб.</TextBlock>

</StackPanel>

</DockPanel>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Button x:Name="RemoveObject"

Content="Удалить"

Grid.Row="4"

Grid.ColumnSpan="2"

HorizontalAlignment="Center"

Click="RemoveObject\_Click"/>

<Button x:Name="RefreshTable"

Content="Обновить"

Grid.Row="4"

Grid.Column="2"

HorizontalAlignment="Left"

Click="RefreshTable\_Click"/>

<Button x:Name="InsertObject"

Content="+"

Grid.Row="4"

Grid.Column="2"

HorizontalAlignment="Right"

Click="InsertObject\_Click"

IsDefault="True"/>

<Button x:Name="UpdateObject"

Content="Изменить"

Grid.Row="4"

Grid.Column="3"

HorizontalAlignment="Center"

Grid.ColumnSpan="2"

Click="UpdateObject\_Click"/>

<Button x:Name="Back"

HorizontalAlignment="Left"

VerticalAlignment="Top"

Height="40"

Width="40"

Opacity="1"

BorderBrush="Transparent"

Background="Transparent"

Margin="5,5,0,0"

Click="Back\_Click"

IsCancel="True">

<Image Source="/Images/Back.png"/>

</Button>

</Grid>

</UserControl>

**Листинг кода отображения ProductsView**

using Cafe.Models;

using Cafe.Tools;

using System.Data.Entity;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace Cafe.View.ProductsViews

{

/// <summary>

/// Логика взаимодействия для ProductsView.xaml

/// </summary>

public partial class ProductsView : UserControl

{

public ProductsView()

{

InitializeComponent();

RefreshAll();

}

private void Back\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new MainView();

}

private void InsertObject\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new ProductsControlView();

}

private void UpdateObject\_Click(object sender, RoutedEventArgs e)

{

if (ProductsList.SelectedItem != null)

{

MainWindow.AppMainWindow.MainScreen.Content = new ProductsControlView((Products)ProductsList.SelectedItem);

}

else

{

NotificationActions.NeedSelectForUpdate();

}

}

private void RemoveObject\_Click(object sender, RoutedEventArgs e)

{

try

{

if (ProductsList.SelectedItem != null)

{

if (NotificationActions.GetRemoveResponse())

{

DBContext.Context.Products.Remove((Products)ProductsList.SelectedItem);

DBContext.Context.SaveChanges();

}

}

else

{

NotificationActions.NeedSelectBeforeRemove();

}

}

catch

{

NotificationActions.KeyProblem();

}

RefreshAll();

}

private void Search\_TextChanged(object sender, TextChangedEventArgs e)

{

if (Search.Text.Count() != 0)

{

foreach (var item in ProductsList.Items)

{

var product = (Products)item;

if (product.ProductID.ToString().Contains(Search.Text) ||

product.Name.Contains(Search.Text))

{

ProductsList.SelectedItem = item;

ProductsList.ScrollIntoView(item);

return;

}

}

ProductsList.SelectedIndex = -1;

}

else ProductsList.SelectedIndex = -1;

}

private void RefreshTable\_Click(object sender, RoutedEventArgs e)

{

RefreshAll();

}

private void RefreshAll()

{

DBContext.Context.Products.Load();

ProductsList.ItemsSource = null;

ProductsList.ItemsSource = DBContext.Context.Products.Local.ToBindingList();

}

}

}

**Листинг отображения ProductsControlView**

<UserControl x:Class="Cafe.View.ProductsViews.ProductsControlView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800" Loaded="UserControl\_Loaded">

<UserControl.Background>

<ImageBrush ImageSource="/Images/Cafe\_Background.jpg" Stretch="UniformToFill"

Opacity="0.5"/>

</UserControl.Background>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="34\*"/>

<RowDefinition Height="126\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center">

<Image x:Name="ActionPicture"

Source="/Images/Insert.png"

Height="40"

Width="40" Margin="10"/>

<TextBlock x:Name="Title"

Text="Добавление товара"

FontSize="30"

Height="36"

FontFamily="SF Pro Display"

FontWeight="SemiBold">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<Image Source="/Images/Products.png"

Height="40"

Width="40" Margin="10"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Grid.Row="1">

<TextBlock Text="Наименование"

FontSize="24"

FontWeight="Medium"

TextAlignment="Center"

Margin="10">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="ObjectName"

Width="512"

Margin="10"

TextAlignment="Center"

MaxLength="50"/>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<StackPanel>

<TextBlock Text="Цена (рубли)"

FontSize="16"

FontWeight="Medium"

TextAlignment="Center"

Margin="14.5">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="Price"

Width="128"

Margin="10"

TextAlignment="Center"

MaxLength="9"/>

</StackPanel>

<StackPanel Grid.Column="1">

<TextBlock Text="Остатки"

FontSize="24"

FontWeight="Medium"

TextAlignment="Center"

Margin="10">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="HoldCount"

Width="128"

Margin="10"

TextAlignment="Center"

MaxLength="9"/>

</StackPanel>

<StackPanel Grid.Column="2">

<TextBlock Text="Единица измерения"

FontSize="16"

FontWeight="Medium"

TextAlignment="Center"

Margin="14.5">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<ComboBox x:Name="Measures"

DisplayMemberPath="Name"

Width="128"

Height="30"

Margin="10"

HorizontalAlignment="Center"

HorizontalContentAlignment="Center"

VerticalContentAlignment="Center"/>

</StackPanel>

</Grid>

<Grid Margin="0,50">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Button x:Name="AddObject"

Content="Добавить"

Grid.Column="1"

IsDefault="True"

Click="AddObject\_Click"/>

<Button x:Name="Cancel"

Content="Отмена"

IsCancel="True"

Click="Cancel\_Click"/>

</Grid>

</StackPanel>

</Grid>

</UserControl>

**Листинг кода отображения ProductsControlView**

using Cafe.Models;

using Cafe.Tools;

using System;

using System.Data.Entity;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media.Imaging;

namespace Cafe.View.ProductsViews

{

/// <summary>

/// Логика взаимодействия для ProductsControlView.xaml

/// </summary>

public partial class ProductsControlView : UserControl

{

public ProductsControlView()

{

InitializeComponent();

Measures.ItemsSource = DBContext.Context.Measures.Local.ToBindingList();

Measures.SelectedIndex = 0;

}

public ProductsControlView(Products product)

{

InitializeComponent();

Measures.ItemsSource = DBContext.Context.Measures.Local.ToBindingList();

Measures.SelectedItem = product.Measures;

ActionPicture.Source = new BitmapImage(new Uri("/Images/Update.png", UriKind.Relative));

Title.Text = "Изменение товара";

AddObject.Content = "Сохранить";

SelectedProduct = product;

ObjectName.Text = product.Name;

Price.Text = product.Price.ToString("F2");

HoldCount.Text = product.HoldCount.ToString();

}

Products SelectedProduct;

private void Cancel\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new ProductsView();

}

private void AddObject\_Click(object sender, RoutedEventArgs e)

{

try

{

if (DBContext.Context.Products == null)

{

NotificationActions.NoComboList();

return;

}

int id = 1, count = 0;

decimal price = 0;

try

{

DBContext.Context.Products.Load();

id = DBContext.Context.Products.ToList().Max(p => p.ProductID) + 1;

}

catch { }

try

{

price = Convert.ToDecimal(Price.Text);

count = Convert.ToInt32(HoldCount.Text);

}

catch

{

NotificationActions.IntError();

return;

}

if (SelectedProduct == null)

{

DBContext.Context.Products.Add(new Products()

{

ProductID = id,

Name = ObjectName.Text,

Price = price,

HoldCount = count,

MeasureID = ((Measures)Measures.SelectedItem).MeasureID

});

}

else

{

SelectedProduct.Name = ObjectName.Text;

SelectedProduct.Price = price;

SelectedProduct.HoldCount = count;

SelectedProduct.MeasureID = ((Measures)Measures.SelectedItem).MeasureID;

}

DBContext.Context.SaveChanges();

MainWindow.AppMainWindow.MainScreen.Content = new ProductsView();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void UserControl\_Loaded(object sender, RoutedEventArgs e)

{

ObjectName.Focus();

}

}

}

**Листинг отображения MeasuresView**

<UserControl x:Class="Cafe.View.MeasuresViews.MeasuresView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<UserControl.Background>

<ImageBrush ImageSource="/Images/Cafe\_Background.jpg" Stretch="UniformToFill"

Opacity="0.5"/>

</UserControl.Background>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="34\*" MaxHeight="88"/>

<RowDefinition Height="44"/>

<RowDefinition Height="30"/>

<RowDefinition Height="126\*"/>

<RowDefinition Height="44"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="96"/>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="406\*"/>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="96"/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Grid.Column="2">

<Image Source="/Images/Measures.png"

Height="40"

Width="40" Margin="10"/>

<TextBlock Text="Единицы измерения"

FontSize="30"

Height="36"

FontFamily="SF Pro Display"

FontWeight="SemiBold">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

</StackPanel>

<Border Grid.ColumnSpan="5"

Grid.Row="2"

Background="AliceBlue"

Opacity="0.9"

BorderThickness="1"

BorderBrush="Gray">

<DockPanel>

<TextBlock Text="№"

HorizontalAlignment="Center"

Width="120"

TextAlignment="Center"

VerticalAlignment="Center"/>

<Line X1="0" Y1="0" X2="0" Y2="95" Stroke="Black"/>

<TextBlock Text="Единица измерения"

HorizontalAlignment="Center"

TextAlignment="Center"

VerticalAlignment="Center"/>

</DockPanel>

</Border>

<TextBlock Text="Искать (№, Имя)"

FontSize="16"

Grid.Row="1"

Grid.ColumnSpan="2"

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontFamily="SF Pro Display"

FontWeight="SemiBold">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="15"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="Search"

Grid.Row="1"

Grid.ColumnSpan="2"

Grid.Column="2"

Opacity="0.9"

Margin="0,7,0,7"

TextChanged="Search\_TextChanged"

MaxLength="50"/>

<ListView x:Name="MeasuresList"

Grid.Row="3"

Grid.ColumnSpan="5"

Opacity="0.9"

Grid.RowSpan="1"

Background="AliceBlue"

SelectionMode="Single">

<ListView.ItemTemplate>

<DataTemplate>

<DockPanel>

<TextBlock Text="{Binding MeasureID}"

Width="104"

TextAlignment="Center"

VerticalAlignment="Center"></TextBlock>

<TextBlock Text="{Binding Name}"

Margin="15,0"

VerticalAlignment="Center"></TextBlock>

</DockPanel>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Button x:Name="RemoveObject"

Content="Удалить"

Grid.Row="4"

Grid.ColumnSpan="2"

HorizontalAlignment="Center"

Click="RemoveObject\_Click"/>

<Button x:Name="RefreshTable"

Content="Обновить"

Grid.Row="4"

Grid.Column="2"

HorizontalAlignment="Left"

Click="RefreshTable\_Click"/>

<Button x:Name="InsertObject"

Content="+"

Grid.Row="4"

Grid.Column="2"

HorizontalAlignment="Right"

Click="InsertObject\_Click"

IsDefault="True"/>

<Button x:Name="UpdateObject"

Content="Изменить"

Grid.Row="4"

Grid.Column="3"

HorizontalAlignment="Center"

Grid.ColumnSpan="2"

Click="UpdateObject\_Click"/>

<Button x:Name="Back"

HorizontalAlignment="Left"

VerticalAlignment="Top"

Height="40"

Width="40"

Opacity="1"

BorderBrush="Transparent"

Background="Transparent"

Margin="5,5,0,0"

Click="Back\_Click"

IsCancel="True">

<Image Source="/Images/Back.png"/>

</Button>

</Grid>

</UserControl>

**Листинг кода отображения MeasuresView**

using Cafe.Models;

using Cafe.Tools;

using System.Data.Entity;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace Cafe.View.MeasuresViews

{

/// <summary>

/// Логика взаимодействия для MeasuresView.xaml

/// </summary>

public partial class MeasuresView : UserControl

{

public MeasuresView()

{

InitializeComponent();

RefreshAll();

}

private void Back\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new MainView();

}

private void InsertObject\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new MeasuresControlView();

}

private void UpdateObject\_Click(object sender, RoutedEventArgs e)

{

if (MeasuresList.SelectedItem != null)

{

MainWindow.AppMainWindow.MainScreen.Content = new MeasuresControlView((Measures)MeasuresList.SelectedItem);

}

else

{

NotificationActions.NeedSelectForUpdate();

}

}

private void RemoveObject\_Click(object sender, RoutedEventArgs e)

{

try

{

if (MeasuresList.SelectedItem != null)

{

if (NotificationActions.GetRemoveResponse())

{

DBContext.Context.Measures.Remove((Measures)MeasuresList.SelectedItem);

DBContext.Context.SaveChanges();

}

}

else

{

NotificationActions.NeedSelectBeforeRemove();

}

}

catch

{

NotificationActions.KeyProblem();

}

RefreshAll();

}

private void Search\_TextChanged(object sender, TextChangedEventArgs e)

{

if (Search.Text.Count() != 0)

{

foreach (var item in MeasuresList.Items)

{

var measure = (Measures)item;

if (measure.MeasureID.ToString().Contains(Search.Text) ||

measure.Name.Contains(Search.Text))

{

MeasuresList.SelectedItem = item;

MeasuresList.ScrollIntoView(item);

return;

}

}

MeasuresList.SelectedIndex = -1;

}

else MeasuresList.SelectedIndex = -1;

}

private void RefreshTable\_Click(object sender, RoutedEventArgs e)

{

RefreshAll();

}

private void RefreshAll()

{

DBContext.Context.Measures.Load();

MeasuresList.ItemsSource = null;

MeasuresList.ItemsSource = DBContext.Context.Measures.Local.ToBindingList();

}

}

}

**Листинг отображения MeasuresControlView**

<UserControl x:Class="Cafe.View.MeasuresViews.MeasuresControlView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800" Loaded="UserControl\_Loaded">

<UserControl.Background>

<ImageBrush ImageSource="/Images/Cafe\_Background.jpg" Stretch="UniformToFill"

Opacity="0.5"/>

</UserControl.Background>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="34\*"/>

<RowDefinition Height="126\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center">

<Image x:Name="ActionPicture"

Source="/Images/Insert.png"

Height="40"

Width="40" Margin="10"/>

<TextBlock x:Name="Title"

Text="Добавление единицы измерения"

FontSize="30"

Height="36"

FontFamily="SF Pro Display"

FontWeight="SemiBold">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<Image Source="/Images/Measures.png"

Height="40"

Width="40" Margin="10"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Grid.Row="1">

<TextBlock Text="Наименование"

FontSize="24"

FontWeight="Medium"

TextAlignment="Center"

Margin="15">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="ObjectName"

Width="512"

Margin="15"

TextAlignment="Center"

MaxLength="50"/>

<Grid Margin="0,50">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Button x:Name="AddObject"

Content="Добавить"

Grid.Column="1"

IsDefault="True"

Click="AddObject\_Click"/>

<Button x:Name="Cancel"

Content="Отмена"

IsCancel="True"

Click="Cancel\_Click"/>

</Grid>

</StackPanel>

</Grid>

</UserControl>

**Листинг кода отображения MeasuresControlView**

using Cafe.Models;

using System;

using System.Data.Entity;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media.Imaging;

namespace Cafe.View.MeasuresViews

{

/// <summary>

/// Логика взаимодействия для MeasuresControlView.xaml

/// </summary>

public partial class MeasuresControlView : UserControl

{

public MeasuresControlView()

{

InitializeComponent();

}

public MeasuresControlView(Measures measure)

{

InitializeComponent();

ActionPicture.Source = new BitmapImage(new Uri("/Images/Update.png", UriKind.Relative));

Title.Text = "Изменение единицы измерения";

AddObject.Content = "Сохранить";

SelectedMeasure = measure;

ObjectName.Text = measure.Name;

}

Measures SelectedMeasure;

private void Cancel\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new MeasuresView();

}

private void AddObject\_Click(object sender, RoutedEventArgs e)

{

try

{

if (SelectedMeasure == null)

{

int id = 1;

try

{

DBContext.Context.Measures.Load();

id = DBContext.Context.Measures.ToList().Max(p => p.MeasureID) + 1;

}

catch { }

DBContext.Context.Measures.Add(new Measures()

{

MeasureID = id,

Name = ObjectName.Text

});

}

else

{

SelectedMeasure.Name = ObjectName.Text;

}

DBContext.Context.SaveChanges();

MainWindow.AppMainWindow.MainScreen.Content = new MeasuresView();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void UserControl\_Loaded(object sender, RoutedEventArgs e)

{

ObjectName.Focus();

}

}

}

**Листинг отображения ChecksView**

<UserControl x:Class="Cafe.View.ChecksViews.ChecksView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<UserControl.Background>

<ImageBrush ImageSource="/Images/Cafe\_Background.jpg" Stretch="UniformToFill"

Opacity="0.5"/>

</UserControl.Background>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="69\*" MaxHeight="88"/>

<RowDefinition Height="44"/>

<RowDefinition Height="30"/>

<RowDefinition Height="102\*"/>

<RowDefinition Height="30"/>

<RowDefinition Height="118\*"/>

<RowDefinition Height="44"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="96"/>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="406\*"/>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="96"/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Grid.Column="2">

<Image Source="/Images/Checks.png"

Height="40"

Width="40" Margin="10"/>

<TextBlock Text="Чеки"

FontSize="30"

Height="36"

FontFamily="SF Pro Display"

FontWeight="SemiBold">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

</StackPanel>

<TextBlock Text="Искать (№, Итого)"

FontSize="16"

Grid.Row="1"

Grid.ColumnSpan="2"

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontFamily="SF Pro Display"

FontWeight="SemiBold">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="15"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="Search" Grid.Row="1"

Grid.ColumnSpan="2"

Grid.Column="2"

Opacity="0.9"

Margin="0,7,0,7"

TextChanged="Search\_TextChanged"

MaxLength="50"/>

<Border Grid.ColumnSpan="5"

Grid.Row="2"

Background="AliceBlue"

Opacity="0.9"

BorderThickness="1"

BorderBrush="Gray">

<DockPanel>

<TextBlock Text="№"

HorizontalAlignment="Center"

Width="120"

TextAlignment="Center"

VerticalAlignment="Center"/>

<Line X1="0" Y1="0" X2="0" Y2="95" Stroke="Black"/>

<TextBlock Text="Чек"

HorizontalAlignment="Center"

TextAlignment="Center"

VerticalAlignment="Center"/>

</DockPanel>

</Border>

<ListView x:Name="ChecksList"

Grid.Row="3"

Grid.ColumnSpan="5"

Opacity="0.9"

Background="AliceBlue"

SelectionMode="Single"

SelectionChanged="ChecksList\_SelectionChanged">

<ListView.ItemTemplate>

<DataTemplate>

<DockPanel>

<TextBlock Text="{Binding CheckID}"

Width="104"

TextAlignment="Center"

VerticalAlignment="Center"></TextBlock>

<StackPanel>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Left"

Margin="15,0">

<TextBlock FontSize="16">Комплексный обед:</TextBlock>

<TextBlock Text="{Binding IsComplexLunchString}"

FontSize="16"

Margin="5,0"></TextBlock>

</StackPanel>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Left"

Margin="15,0">

<TextBlock FontSize="16">Без скидки:</TextBlock>

<TextBlock Text="{Binding Cost, StringFormat=F2}"

FontSize="16"

Margin="5,0"></TextBlock>

<TextBlock Margin="5,0"

FontSize="18">руб.</TextBlock>

</StackPanel>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Left"

Margin="15,0">

<TextBlock FontSize="16">Скидка:</TextBlock>

<TextBlock Text="{Binding Discount, StringFormat=F2}"

FontSize="16"

Margin="5,0"></TextBlock>

<TextBlock Margin="5,0"

FontSize="18">руб.</TextBlock>

</StackPanel>

</StackPanel>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Right">

<TextBlock Margin="5,0"

FontSize="18">Итого:</TextBlock>

<TextBlock Text="{Binding CostNDiscount, StringFormat=F2}"

FontSize="18"></TextBlock>

<TextBlock Margin="5,0"

FontSize="18">руб.</TextBlock>

</StackPanel>

</DockPanel>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Border Grid.ColumnSpan="5"

Grid.Row="4"

Background="AliceBlue"

Opacity="0.9"

BorderThickness="1"

BorderBrush="Gray">

<DockPanel>

<TextBlock Text="№"

HorizontalAlignment="Center"

Width="120"

TextAlignment="Center"

VerticalAlignment="Center"/>

<Line X1="0" Y1="0" X2="0" Y2="95" Stroke="Black"/>

<TextBlock Text="Товары в выбранном чеке"

HorizontalAlignment="Center"

TextAlignment="Center"

VerticalAlignment="Center"/>

</DockPanel>

</Border>

<ListView x:Name="PurchasesList"

Grid.Row="5"

Grid.ColumnSpan="5"

Opacity="0.9"

Background="AliceBlue"

SelectionMode="Single">

<ListView.ItemTemplate>

<DataTemplate>

<DockPanel>

<TextBlock Text="{Binding PurchaseID}"

Width="104"

TextAlignment="Center"

VerticalAlignment="Center"></TextBlock>

<StackPanel>

<TextBlock Text="{Binding Products.Name}"

Margin="15,0"

FontWeight="Bold"

FontSize="18"></TextBlock>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Left"

Margin="15,0">

<TextBlock FontSize="16">Кол-во:</TextBlock>

<TextBlock Text="{Binding ProductCount}"

FontSize="16"

Margin="5,0"></TextBlock>

<TextBlock Text="{Binding Products.Measures.Name}"

FontSize="16"></TextBlock>

</StackPanel>

</StackPanel>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Right">

<TextBlock Text="{Binding ProductCost, StringFormat=F2}"

FontSize="18"></TextBlock>

<TextBlock Margin="5,0"

FontSize="18">руб.</TextBlock>

</StackPanel>

</DockPanel>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Button x:Name="RemoveObject"

Content="Удалить"

Grid.Row="6"

Grid.ColumnSpan="2"

HorizontalAlignment="Center"

Click="RemoveObject\_Click"

Margin="0,7,0,7"/>

<Button x:Name="RefreshTable"

Content="Обновить"

Grid.Row="6"

Grid.Column="2"

HorizontalAlignment="Left"

Click="RefreshTable\_Click"

Margin="0,7,0,7"/>

<Button x:Name="InsertObject"

Content="+"

Grid.Row="6"

Grid.Column="2"

HorizontalAlignment="Right"

Click="InsertObject\_Click"

Margin="0,7,0,7"

IsDefault="True"/>

<Button x:Name="UpdateObject"

Content="Изменить"

Grid.Row="6"

Grid.Column="3"

HorizontalAlignment="Center"

Grid.ColumnSpan="2"

Click="UpdateObject\_Click"/>

<Button x:Name="Back"

HorizontalAlignment="Left"

VerticalAlignment="Top"

Height="40"

Width="40"

Opacity="1"

BorderBrush="Transparent"

Background="Transparent"

Margin="5,5,0,0"

Click="Back\_Click"

IsCancel="True">

<Image Source="/Images/Back.png"/>

</Button>

</Grid>

</UserControl>

**Листинг кода отображения ChecksView**

using Cafe.Models;

using Cafe.Tools;

using System.Data.Entity;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace Cafe.View.ChecksViews

{

/// <summary>

/// Логика взаимодействия для ChecksView.xaml

/// </summary>

public partial class ChecksView : UserControl

{

public ChecksView()

{

InitializeComponent();

RefreshAll();

}

private void Back\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new MainView();

}

private void InsertObject\_Click(object sender, RoutedEventArgs e)

{

MainWindow.AppMainWindow.MainScreen.Content = new ChecksControlView();

}

private void UpdateObject\_Click(object sender, RoutedEventArgs e)

{

if (ChecksList.SelectedItem != null)

{

MainWindow.AppMainWindow.MainScreen.Content = new ChecksControlView((Checks)ChecksList.SelectedItem);

}

else

{

NotificationActions.NeedSelectForUpdate();

}

}

private void RemoveObject\_Click(object sender, RoutedEventArgs e)

{

try

{

if (ChecksList.SelectedItem != null)

{

if (NotificationActions.GetRemoveResponse())

{

var check = (Checks)ChecksList.SelectedItem;

foreach (var item in check.Purchases.ToList())

{

DBContext.Context.Purchases.Remove(item);

}

DBContext.Context.Checks.Remove(check);

DBContext.Context.SaveChanges();

}

}

else

{

NotificationActions.NeedSelectBeforeRemove();

}

}

catch

{

NotificationActions.KeyProblem();

}

RefreshAll();

}

private void ChecksList\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (ChecksList.SelectedItem != null)

{

PurchasesList.ItemsSource = ((Checks)ChecksList.SelectedItem).Purchases;

}

else PurchasesList.ItemsSource = null;

}

private void Search\_TextChanged(object sender, TextChangedEventArgs e)

{

if (Search.Text.Count() != 0)

{

foreach (var item in ChecksList.Items)

{

var check = (Checks)item;

if (check.CheckID.ToString().Contains(Search.Text) ||

check.CostNDiscount.ToString().Contains(Search.Text))

{

ChecksList.SelectedItem = item;

ChecksList.ScrollIntoView(item);

return;

}

}

ChecksList.SelectedIndex = -1;

}

else ChecksList.SelectedIndex = -1;

}

private void RefreshTable\_Click(object sender, RoutedEventArgs e)

{

RefreshAll();

}

private void RefreshAll()

{

DBContext.Context.Checks.Load();

DBContext.Context.Purchases.Load();

ChecksList.ItemsSource = null;

ChecksList.ItemsSource = DBContext.Context.Checks.Local.ToBindingList();

}

}

}

**Листинг отображения ChecksControlView**

<UserControl x:Class="Cafe.View.ChecksViews.ChecksControlView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Cafe.View.ChecksViews"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<UserControl.Background>

<ImageBrush ImageSource="/Images/Cafe\_Background.jpg" Stretch="UniformToFill"

Opacity="0.5"/>

</UserControl.Background>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="79\*" MaxHeight="88"/>

<RowDefinition Height="79"/>

<RowDefinition Height="82"/>

<RowDefinition Height="30"/>

<RowDefinition Height="136\*"/>

<RowDefinition Height="44"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="96"/>

<ColumnDefinition/>

<ColumnDefinition Width="407.34"/>

<ColumnDefinition/>

<ColumnDefinition Width="96"/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Grid.Column="2" Height="60" Width="364">

<Image x:Name="ActionPicture"

Source="/Images/Insert.png"

Height="40"

Width="40" Margin="10"/>

<TextBlock x:Name="Title"

Text="Добавление чека"

FontSize="30"

Height="36"

FontFamily="SF Pro Display"

FontWeight="SemiBold">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<Image Source="/Images/Checks.png"

Height="40"

Width="40" Margin="10"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Grid.Row="1"

Grid.ColumnSpan="3"

HorizontalAlignment="Left" Width="278">

<TextBlock Text="Комплексный обед"

FontSize="16"

FontWeight="Medium"

TextAlignment="Center"

VerticalAlignment="Center"

Margin="24,0">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<ComboBox x:Name="ComplexLunch"

MinWidth="80"

Height="30"

HorizontalAlignment="Center"

HorizontalContentAlignment="Center"

VerticalContentAlignment="Center"

SelectedIndex="0" SelectionChanged="ComplexLunch\_SelectionChanged">

<ComboBoxItem>Нет</ComboBoxItem>

<ComboBoxItem>Да</ComboBoxItem>

</ComboBox>

</StackPanel>

<StackPanel Grid.Column="2"

Grid.Row="1"

Orientation="Horizontal"

HorizontalAlignment="Center" Width="216">

<TextBlock Text="Скидка:"

FontSize="16"

FontWeight="Medium"

TextAlignment="Center"

VerticalAlignment="Center"

Margin="15,0">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="Discount"

Text="0"

Width="128"

TextAlignment="Right"

IsReadOnly="True"/>

</StackPanel>

<StackPanel Grid.Column="2"

Grid.ColumnSpan="3"

Grid.Row="1"

Orientation="Horizontal"

HorizontalAlignment="Right" Width="281">

<TextBlock Text="Сумма всего:"

FontSize="16"

FontWeight="Medium"

TextAlignment="Center"

VerticalAlignment="Center"

Margin="15,0">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="Cost"

Text="0"

Width="128"

TextAlignment="Right"

IsReadOnly="True"

Margin="0,0,24,0"/>

</StackPanel>

<Button x:Name="RemoveObject"

Content="Удалить товар"

Grid.Row="2"

Grid.ColumnSpan="2"

HorizontalAlignment="Center"

Click="RemoveObject\_Click"

Margin="0,39,0,13"

Width="150"/>

<Grid Grid.Row="2"

Grid.Column="2">

<StackPanel HorizontalAlignment="Left">

<TextBlock Text="Товар"

FontSize="24"

FontWeight="Medium"

TextAlignment="Center"

Margin="4.5">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<ComboBox x:Name="Products"

DisplayMemberPath="Name"

Width="250"

Height="30"

HorizontalContentAlignment="Center"

VerticalContentAlignment="Center"/>

</StackPanel>

<StackPanel HorizontalAlignment="Right">

<TextBlock Text="Количество"

FontSize="16"

FontWeight="Medium"

TextAlignment="Center"

VerticalAlignment="Center"

Margin="0,10">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="ProductCount"

Width="128"

TextAlignment="Center"/>

</StackPanel>

</Grid>

<Button x:Name="InsertObject"

Content="Добавить товар"

Grid.Row="2"

Grid.Column="3"

Grid.ColumnSpan="2"

HorizontalAlignment="Center"

Click="InsertObject\_Click"

Margin="23,39,0,13"

Width="150"/>

<Border Grid.ColumnSpan="5"

Grid.Row="3"

Background="AliceBlue"

Opacity="0.9"

BorderThickness="1"

BorderBrush="Gray">

<DockPanel>

<TextBlock Text="№"

HorizontalAlignment="Center"

Width="120"

TextAlignment="Center"

VerticalAlignment="Center"/>

<Line X1="0" Y1="0" X2="0" Y2="95" Stroke="Black"/>

<TextBlock Text="Товары в выбранном чеке"

HorizontalAlignment="Center"

TextAlignment="Center"

VerticalAlignment="Center"/>

</DockPanel>

</Border>

<ListView x:Name="PurchasesList"

Grid.Row="4"

Grid.ColumnSpan="5"

Opacity="0.9"

Background="AliceBlue"

SelectionMode="Single">

<ListView.ItemTemplate>

<DataTemplate>

<DockPanel>

<TextBlock Text="{Binding PurchaseID}"

Width="104"

TextAlignment="Center"

VerticalAlignment="Center"></TextBlock>

<StackPanel>

<TextBlock Text="{Binding Products.Name}"

Margin="15,0"

FontWeight="Bold"

FontSize="18"></TextBlock>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Left"

Margin="15,0">

<TextBlock FontSize="16">Кол-во:</TextBlock>

<TextBlock Text="{Binding ProductCount}"

FontSize="16"

Margin="5,0"></TextBlock>

<TextBlock Text="{Binding Products.Measures.Name}"

FontSize="16"></TextBlock>

</StackPanel>

</StackPanel>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Right">

<TextBlock Text="{Binding ProductCost, StringFormat=F2}"

FontSize="18"></TextBlock>

<TextBlock Margin="5,0"

FontSize="18">руб.</TextBlock>

</StackPanel>

</DockPanel>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Button x:Name="Cancel"

Content="Отмена"

Grid.Row="5"

HorizontalAlignment="Center"

Grid.ColumnSpan="2"

Width="150" Margin="0,7,0,7"

Click="Cancel\_Click"

IsCancel="True"/>

<StackPanel Grid.Column="2"

Grid.Row="5"

Orientation="Horizontal"

HorizontalAlignment="Center" Width="212">

<TextBlock Text="Итого:"

FontSize="16"

FontWeight="Black"

TextAlignment="Center"

VerticalAlignment="Center"

Margin="15,0">

<TextBlock.Effect>

<DropShadowEffect BlurRadius="10"

Direction="280"

ShadowDepth="2"

Color="White"/>

</TextBlock.Effect>

</TextBlock>

<TextBox x:Name="CostNDiscount"

Text="0"

Width="128"

TextAlignment="Right"

IsReadOnly="True"/>

</StackPanel>

<Button x:Name="InsertCheck"

Content="Добавить чек"

Grid.Row="5"

Grid.Column="3"

HorizontalAlignment="Center"

Grid.ColumnSpan="2"

Width="150" Margin="23,7,0,7"

Click="InsertCheck\_Click"

IsDefault="True"/>

</Grid>

</UserControl>

**Листинг кода отображения ChecksControlView**

using Cafe.Models;

using Cafe.Tools;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media.Imaging;

namespace Cafe.View.ChecksViews

{

/// <summary>

/// Логика взаимодействия для ChecksControlView.xaml

/// </summary>

public partial class ChecksControlView : UserControl

{

public ChecksControlView()

{

InitializeComponent();

Products.ItemsSource = DBContext.Context.Products.Local.ToBindingList();

Products.SelectedIndex = 0;

}

public ChecksControlView(Checks check)

{

InitializeComponent();

Products.ItemsSource = DBContext.Context.Products.Local.ToBindingList();

Products.SelectedIndex = 0;

ActionPicture.Source = new BitmapImage(new Uri("/Images/Update.png", UriKind.Relative));

Title.Text = "Изменение чека";

InsertCheck.Content = "Сохранить чек";

SelectedCheck = check;

Cost.Text = check.Cost.ToString("F2");

Discount.Text = check.Discount.ToString("F2");

CostNDiscount.Text = check.CostNDiscount.ToString("F2");

if (check.IsComplexLunch)

{

ComplexLunch.SelectedIndex = 1;

}

PurchasesList.ItemsSource = check.Purchases.ToList();

Purchases = check.Purchases.ToList();

}

Checks SelectedCheck;

List<Purchases> Purchases = new List<Purchases>();

private void Cancel\_Click(object sender, RoutedEventArgs e)

{

try

{

DBContext.UpdateContext();

MainWindow.AppMainWindow.MainScreen.Content = new ChecksView();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void InsertObject\_Click(object sender, RoutedEventArgs e)

{

try

{

long id = 1;

int count = 0;

try

{

DBContext.Context.Purchases.Load();

id = DBContext.Context.Purchases.ToList().Max(p => p.PurchaseID) + 1;

}

catch { }

try

{

if (id <= Purchases.Max(p => p.PurchaseID)) id = Purchases.Max(p => p.PurchaseID) + 1;

}

catch { }

try

{

count = Convert.ToInt32(ProductCount.Text);

}

catch

{

NotificationActions.IntError();

return;

}

if (!CheckActions.HasHoldCount((Products)Products.SelectedItem, count))

{

NotificationActions.NoHoldCount();

return;

}

if (SelectedCheck == null) CreateNewCheck();

var purchase = new Purchases()

{

PurchaseID = id,

CheckID = SelectedCheck.CheckID,

ProductID = ((Products)Products.SelectedItem).ProductID,

ProductCount = count

};

Purchases.Add(purchase);

DBContext.Context.Purchases.Add(purchase);

((Products)Products.SelectedItem).HoldCount -= count;//Списание товара с остатков

UpdateDisplayedData();

if (PurchasesList.ItemsSource != null) PurchasesList.ItemsSource = null;

PurchasesList.ItemsSource = Purchases;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void RemoveObject\_Click(object sender, RoutedEventArgs e)

{

try

{

if (PurchasesList.SelectedItem != null)

{

Products product = ((Purchases)PurchasesList.SelectedItem).Products;//Получение для будущего возврата к остаткам

product.HoldCount+=((Purchases)PurchasesList.SelectedItem).ProductCount;//Возвращение в остатки

Purchases.Remove((Purchases)PurchasesList.SelectedItem);

DBContext.Context.Purchases.Remove((Purchases)PurchasesList.SelectedItem);

UpdateDisplayedData();

if (PurchasesList.ItemsSource != null) PurchasesList.ItemsSource = null;

PurchasesList.ItemsSource = Purchases;

}

else NotificationActions.NeedSelectBeforeRemove();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void CreateNewCheck()

{

try

{

long id = 1;

try

{

DBContext.Context.Checks.Load();

id = DBContext.Context.Checks.ToList().Max(p => p.CheckID) + 1;

}

catch { }

SelectedCheck = new Checks()

{

CheckID = id,

Cost = 0,

Discount = 0,

CostNDiscount = 0,

IsComplexLunch = false

};

DBContext.Context.Checks.Add(SelectedCheck);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void InsertCheck\_Click(object sender, RoutedEventArgs e)

{

try

{

DBContext.Context.SaveChanges();

MainWindow.AppMainWindow.MainScreen.Content = new ChecksView();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

/// <summary>

/// Обновление отображаемых данных

/// </summary>

private void UpdateDisplayedData()

{

try

{

decimal cost = CalculateMainCost();

SelectedCheck.Cost = cost;

SelectedCheck.Discount = CheckActions.GetDiscount(cost);

if (ComplexLunch.SelectedIndex == 0)

{

SelectedCheck.CostNDiscount = SelectedCheck.Cost - SelectedCheck.Discount;

}

else

{

SelectedCheck.Discount = 0;

SelectedCheck.CostNDiscount = SelectedCheck.Cost;

}

Cost.Text = SelectedCheck.Cost.ToString("F2");

Discount.Text = SelectedCheck.Discount.ToString("F2");

CostNDiscount.Text = SelectedCheck.CostNDiscount.ToString("F2");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

/// <summary>

/// Рассчитывает стоимость всех товаров

/// </summary>

/// <returns>Стоимость всех товаров</returns>

private decimal CalculateMainCost()

{

try

{

decimal mainCost = 0;

foreach (var item in Purchases)

{

mainCost += item.ProductCost;

}

return mainCost;

}

catch

{

return 0;

}

}

private void ComplexLunch\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (SelectedCheck != null)

{

if (ComplexLunch.SelectedIndex == 0)

{

SelectedCheck.Discount = CheckActions.GetDiscount(SelectedCheck.Cost);

SelectedCheck.CostNDiscount = SelectedCheck.Cost - SelectedCheck.Discount;

SelectedCheck.IsComplexLunch = false;

}

else

{

SelectedCheck.Discount = 0;

SelectedCheck.CostNDiscount = SelectedCheck.Cost;

SelectedCheck.IsComplexLunch = true;

}

Discount.Text = SelectedCheck.Discount.ToString("F2");

CostNDiscount.Text = SelectedCheck.CostNDiscount.ToString("F2");

}

}

}

}

**Инструментальные классы**

**Листинг класса CheckActions**

using Cafe.Models;

namespace Cafe.Tools

{/// <summary>

/// Класс для работы с чеками. Включает в себя дополнительные модули.

/// </summary>

public static class CheckActions

{

/// <summary>

/// Позволяет получить рассчитанную скидку

/// </summary>

/// <param name="cost">Стоимость всех товаров</param>

/// <returns>Возвращает рассчитанную скидку.</returns>

public static decimal GetDiscount(decimal cost)

{

if (cost >= 300 && cost <= 500) return cost / 100 \* 3;

if (cost >= 501 && cost <= 1000) return cost / 100 \* 5;

if (cost >= 1001 && cost <= 5000) return cost / 100 \* 7;

if (cost > 5000) return cost / 100 \* 10;

return 0;

}

/// <summary>

/// Позволяет узнать, хватает ли остатков для покупки товара в указанном количестве

/// </summary>

/// <param name="product">Продукт, имеющий количество на остатках</param>

/// <param name="count">Количество для списания</param>

/// <returns>true, если хватает для списания, false - не хватает остатков для списания</returns>

public static bool HasHoldCount(Products product, int count)

{

if ((product.HoldCount - count) < 0) return false;

else return true;

}

}

}

**Листинг класса DBContext**

using Cafe.Models;

using System.Data.Entity;

namespace Cafe

{/// <summary>

/// Класс для работы с контекстом

/// </summary>

public static class DBContext

{

private static CafeEntities s\_context;

/// <summary>

/// Контекст, хранящий в себе данные, полученные из БД

/// </summary>

public static CafeEntities Context

{

get

{

if (s\_context == null)

{

UpdateContext();

return s\_context;

}

else return s\_context;

}

private set

{

s\_context = value;

}

}

/// <summary>

/// Обновляет данные контекста

/// </summary>

public static void UpdateContext()

{

s\_context = new CafeEntities();

s\_context.Measures.Load();

s\_context.Products.Load();

s\_context.Checks.Load();

s\_context.Purchases.Load();

}

}

}

**Листинг класса NotificationActions**

using System.Windows;

namespace Cafe.Tools

{

/// <summary>

/// Класс для отображения сообщений пользователю

/// </summary>

public static class NotificationActions

{

/// <summary>

/// Отображение о необходимости выбора объекта из списка для изменения

/// </summary>

public static void NeedSelectForUpdate()

{

MessageBox.Show("Для изменения элемента из списка, его необходимо выбрать!", "Изменение",

MessageBoxButton.OK, MessageBoxImage.Warning);

}

/// <summary>

/// Отображение об отсутствии записей в выпадающем списке

/// </summary>

public static void NoComboList()

{

MessageBox.Show("Выпадающий список пуст! Невозможно добавить запись без данных в нем!", "Ошибка",

MessageBoxButton.OK, MessageBoxImage.Error);

}

/// <summary>

/// Ошибка конвертации

/// </summary>

public static void IntError()

{

MessageBox.Show("Числовые поля имеют некоррекные значения! Проверьте правильность!", "Ошибка",

MessageBoxButton.OK, MessageBoxImage.Error);

}

/// <summary>

/// Недостаточно на остатках для списания

/// </summary>

public static void NoHoldCount()

{

MessageBox.Show("Продукта нет в таком количестве на остатках! Невозможно добавить.", "Ошибка",

MessageBoxButton.OK, MessageBoxImage.Error);

}

/// <summary>

/// Сообщение о закрытии окна

/// </summary>

/// <returns>Ответ пользователя, хочет ли он закрыть окно</returns>

public static bool WindowClosing()

{

MessageBoxResult result = MessageBox.Show("Вы действительно хотите закрыть программу?", "Закрытие программы",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.No) return true;

else return false;

}

#region Сообщения удаления

/// <summary>

/// Сообщение о необходимости выбрать объект из списка для удаления

/// </summary>

public static void NeedSelectBeforeRemove()

{

MessageBox.Show("Для удаления элемента из списка, его необходимо выбрать!", "Удаление", MessageBoxButton.OK, MessageBoxImage.Warning);

}

/// <summary>

/// Сообщение о проблеме с ключами

/// </summary>

public static void KeyProblem()

{

MessageBox.Show("Невозможно удалить, так как запись используется в других таблицах или она уже удалена!", "Удаление", MessageBoxButton.OK, MessageBoxImage.Stop);

}

/// <summary>

/// Сообщение об удалении объекта из списка

/// </summary>

/// <returns>Ответ-подтверждение</returns>

public static bool GetRemoveResponse()

{

MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить выбранное в списке?", "Удаление", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

return true;

}

else return false;

}

#endregion

}

}

**Листинг класса StylesActionClass (Используется для стилей)**

using Cafe.Models;

using Cafe.View.ChecksViews;

using Cafe.View.MeasuresViews;

using Cafe.View.ProductsViews;

using System.Windows.Controls;

using System.Windows.Input;

namespace Cafe.Tools

{

/// <summary>

/// Класс стилей для изменения объекта из списка двойным кликом

/// </summary>

public partial class StylesActionClass

{

private void MeasuresList\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

try

{

MainWindow.AppMainWindow.MainScreen.Content = new MeasuresControlView((Measures)((ListViewItem)sender).Content);

}

catch

{

try

{

MainWindow.AppMainWindow.MainScreen.Content = new ProductsControlView((Products)((ListViewItem)sender).Content);

}

catch

{

try

{

MainWindow.AppMainWindow.MainScreen.Content = new ChecksControlView((Checks)((ListViewItem)sender).Content);

}

catch { }

}

}

}

}

}

**Листинг словаря ресурсов со стилями Styles**

<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

x:Class="Cafe.Tools.StylesActionClass"

xmlns:theme="clr-namespace:Microsoft.Windows.Themes;assembly=PresentationFramework.Aero2">

<!--#region Стиль всех кнопок-->

<SolidColorBrush x:Key="Button.MouseOver.Background" Color="#FFBEE6FD"/>

<SolidColorBrush x:Key="Button.MouseOver.Border" Color="#FF3C7FB1"/>

<SolidColorBrush x:Key="Button.Pressed.Background" Color="#FFC4E5F6"/>

<SolidColorBrush x:Key="Button.Pressed.Border" Color="#FF2C628B"/>

<SolidColorBrush x:Key="Button.Disabled.Background" Color="#FFF4F4F4"/>

<SolidColorBrush x:Key="Button.Disabled.Border" Color="#FFADB2B5"/>

<SolidColorBrush x:Key="Button.Disabled.Foreground" Color="#FF838383"/>

<Style TargetType="Button">

<Style.Setters>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type ButtonBase}">

<Border x:Name="border" CornerRadius="6"

Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" SnapsToDevicePixels="true">

<ContentPresenter x:Name="contentPresenter" Focusable="False" HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" Margin="{TemplateBinding Padding}" RecognizesAccessKey="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="Button.IsDefaulted" Value="true">

<Setter Property="BorderBrush" TargetName="border" Value="{DynamicResource {x:Static SystemColors.HighlightBrushKey}}"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="true">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.MouseOver.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.MouseOver.Border}"/>

</Trigger>

<Trigger Property="IsPressed" Value="true">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.Pressed.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.Pressed.Border}"/>

</Trigger>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.Disabled.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.Disabled.Border}"/>

<Setter Property="TextElement.Foreground" TargetName="contentPresenter" Value="{StaticResource Button.Disabled.Foreground}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="FontSize" Value="14"/>

<Setter Property="FontFamily" Value="SF Pro Display"/>

<Setter Property="FontWeight" Value="Medium"/>

<Setter Property="Height" Value="30"/>

<Setter Property="Width" Value="100"/>

<Setter Property="Background" Value="AliceBlue"/>

<Setter Property="BorderBrush" Value="SaddleBrown"/>

<Setter Property="Opacity" Value="0.75"/>

</Style.Setters>

</Style>

<!--#endregion-->

<!--#region Для кнопок-иконок-->

<Style x:Key="ImageButton" TargetType="Button">

<Style.Setters>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type ButtonBase}">

<Border x:Name="border" CornerRadius="24"

Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" SnapsToDevicePixels="true"

Padding="5">

<ContentPresenter x:Name="contentPresenter" Focusable="False"

HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" Margin="{TemplateBinding Padding}" RecognizesAccessKey="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}">

<ContentPresenter.Effect>

<DropShadowEffect BlurRadius="50"

Direction="90"

ShadowDepth="-25"/>

</ContentPresenter.Effect>

</ContentPresenter>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="Button.IsDefaulted" Value="true">

<Setter Property="BorderBrush" TargetName="border" Value="{DynamicResource {x:Static SystemColors.HighlightBrushKey}}"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="true">

<Setter Property="Background" TargetName="border" Value="#AABEE6FD"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.MouseOver.Border}"/>

</Trigger>

<Trigger Property="IsPressed" Value="true">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.Pressed.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.Pressed.Border}"/>

</Trigger>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.Disabled.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.Disabled.Border}"/>

<Setter Property="TextElement.Foreground" TargetName="contentPresenter" Value="{StaticResource Button.Disabled.Foreground}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="FontSize" Value="14"/>

<Setter Property="FontFamily" Value="SF Pro Display"/>

<Setter Property="FontWeight" Value="Medium"/>

<Setter Property="Height" Value="30"/>

<Setter Property="Width" Value="100"/>

<Setter Property="Background" Value="AliceBlue"/>

<Setter Property="BorderBrush" Value="SaddleBrown"/>

<Setter Property="Opacity" Value="0.75"/>

</Style.Setters>

</Style>

<!--#endregion -->

<!--#region Стиль для полей-->

<SolidColorBrush x:Key="TextBox.Static.Border" Color="#FFABAdB3"/>

<SolidColorBrush x:Key="TextBox.MouseOver.Border" Color="#FF7EB4EA"/>

<SolidColorBrush x:Key="TextBox.Focus.Border" Color="#FF0bcdff"/>

<Style TargetType="TextBox">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type TextBoxBase}">

<Border Background="{TemplateBinding Background}"

x:Name="Bd" BorderBrush="{TemplateBinding BorderBrush}"

BorderThickness="{TemplateBinding BorderThickness}" CornerRadius="6">

<ScrollViewer x:Name="PART\_ContentHost" Margin="1,0,1,0"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsEnabled" Value="False">

<Setter Property="Background" Value="{DynamicResource {x:Static SystemColors.ControlBrushKey}}" TargetName="Bd"/>

<Setter Property="Foreground" Value="{DynamicResource {x:Static SystemColors.GrayTextBrushKey}}"/>

</Trigger>

<Trigger Property="Width" Value="Auto">

<Setter Property="MinWidth" Value="100"/>

</Trigger>

<Trigger Property="Height" Value="Auto">

<Setter Property="MinHeight" Value="20"/>

</Trigger>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Opacity" TargetName="Bd" Value="0.56"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="true">

<Setter Property="BorderBrush" TargetName="Bd" Value="{StaticResource TextBox.MouseOver.Border}"/>

</Trigger>

<Trigger Property="IsKeyboardFocused" Value="true">

<Setter Property="BorderBrush" TargetName="Bd" Value="{StaticResource TextBox.Focus.Border}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="FontSize" Value="14"/>

<Setter Property="FontFamily" Value="SF Pro Display"/>

<Setter Property="Height" Value="30"/>

<Setter Property="FontWeight" Value="Light"/>

<Setter Property="VerticalContentAlignment" Value="Center"/>

</Style>

<!--#endregion-->

<!--#region Стиль текста-->

<Style TargetType="TextBlock">

<Style.Setters>

<Setter Property="FontFamily" Value="SF Pro Display"/>

<Setter Property="FontSize" Value="14"/>

</Style.Setters>

</Style>

<!--#endregion-->

<!--#region Стиль элемента списка-->

<Style TargetType="ListViewItem">

<Setter Property="HorizontalContentAlignment" Value="Stretch"/>

<Setter Property="Background" Value="#eadbd4"/>

<Setter Property="Margin" Value="3"/>

<Setter Property="MinHeight" Value="28"/>

<EventSetter Event="MouseDoubleClick" Handler="MeasuresList\_MouseDoubleClick"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type ListBoxItem}">

<Border x:Name="brd" Background="{TemplateBinding Background}"

BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" CornerRadius="6">

<ContentPresenter/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsSelected" Value="True">

<Setter Property="Background" Value="#e6cfaa"/>

<Setter Property="Foreground" Value="Black"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="BorderBrush" Value="{StaticResource TextBox.Focus.Border}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="{StaticResource Button.MouseOver.Background}"/>

</Trigger>

</Style.Triggers>

</Style>

<!--#endregion-->

<!--#region Стиль выпадающего списка-->

<Style x:Key="FocusVisual">

<Setter Property="Control.Template">

<Setter.Value>

<ControlTemplate>

<Rectangle Margin="2" StrokeDashArray="1 2" Stroke="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}" SnapsToDevicePixels="true" StrokeThickness="1"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<SolidColorBrush x:Key="TextBox.Static.Background" Color="#FFFFFFFF"/>

<Style x:Key="ComboBoxEditableTextBox" TargetType="{x:Type TextBox}">

<Setter Property="OverridesDefaultStyle" Value="true"/>

<Setter Property="AllowDrop" Value="true"/>

<Setter Property="MinWidth" Value="0"/>

<Setter Property="MinHeight" Value="0"/>

<Setter Property="FocusVisualStyle" Value="{x:Null}"/>

<Setter Property="ScrollViewer.PanningMode" Value="VerticalFirst"/>

<Setter Property="Stylus.IsFlicksEnabled" Value="False"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type TextBox}">

<ScrollViewer x:Name="PART\_ContentHost" Background="Transparent" Focusable="false" HorizontalScrollBarVisibility="Hidden" VerticalScrollBarVisibility="Hidden"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<LinearGradientBrush x:Key="ComboBox.Static.Background" EndPoint="0,1" StartPoint="0,0">

<GradientStop Color="#FFF0F0F0" Offset="0.0"/>

<GradientStop Color="#FFE5E5E5" Offset="1.0"/>

</LinearGradientBrush>

<SolidColorBrush x:Key="ComboBox.Static.Border" Color="#FFACACAC"/>

<SolidColorBrush x:Key="ComboBox.Static.Glyph" Color="#FF606060"/>

<SolidColorBrush x:Key="ComboBox.Static.Editable.Background" Color="#FFFFFFFF"/>

<SolidColorBrush x:Key="ComboBox.Static.Editable.Border" Color="#FFABADB3"/>

<SolidColorBrush x:Key="ComboBox.Static.Editable.Button.Background" Color="Transparent"/>

<SolidColorBrush x:Key="ComboBox.Static.Editable.Button.Border" Color="Transparent"/>

<LinearGradientBrush x:Key="ComboBox.MouseOver.Background" EndPoint="0,1" StartPoint="0,0">

<GradientStop Color="#FFECF4FC" Offset="0.0"/>

<GradientStop Color="#FFDCECFC" Offset="1.0"/>

</LinearGradientBrush>

<SolidColorBrush x:Key="ComboBox.MouseOver.Border" Color="#FF7EB4EA"/>

<SolidColorBrush x:Key="ComboBox.MouseOver.Glyph" Color="#FF000000"/>

<SolidColorBrush x:Key="ComboBox.MouseOver.Editable.Background" Color="#FFFFFFFF"/>

<SolidColorBrush x:Key="ComboBox.MouseOver.Editable.Border" Color="#FF7EB4EA"/>

<LinearGradientBrush x:Key="ComboBox.MouseOver.Editable.Button.Background" EndPoint="0,1" StartPoint="0,0">

<GradientStop Color="#FFEBF4FC" Offset="0.0"/>

<GradientStop Color="#FFDCECFC" Offset="1.0"/>

</LinearGradientBrush>

<SolidColorBrush x:Key="ComboBox.MouseOver.Editable.Button.Border" Color="#FF7EB4EA"/>

<LinearGradientBrush x:Key="ComboBox.Pressed.Background" EndPoint="0,1" StartPoint="0,0">

<GradientStop Color="#FFDAECFC" Offset="0.0"/>

<GradientStop Color="#FFC4E0FC" Offset="1.0"/>

</LinearGradientBrush>

<SolidColorBrush x:Key="ComboBox.Pressed.Border" Color="#FF569DE5"/>

<SolidColorBrush x:Key="ComboBox.Pressed.Glyph" Color="#FF000000"/>

<SolidColorBrush x:Key="ComboBox.Pressed.Editable.Background" Color="#FFFFFFFF"/>

<SolidColorBrush x:Key="ComboBox.Pressed.Editable.Border" Color="#FF569DE5"/>

<LinearGradientBrush x:Key="ComboBox.Pressed.Editable.Button.Background" EndPoint="0,1" StartPoint="0,0">

<GradientStop Color="#FFDAEBFC" Offset="0.0"/>

<GradientStop Color="#FFC4E0FC" Offset="1.0"/>

</LinearGradientBrush>

<SolidColorBrush x:Key="ComboBox.Pressed.Editable.Button.Border" Color="#FF569DE5"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Background" Color="#FFF0F0F0"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Border" Color="#FFD9D9D9"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Glyph" Color="#FFBFBFBF"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Editable.Background" Color="#FFFFFFFF"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Editable.Border" Color="#FFBFBFBF"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Editable.Button.Background" Color="Transparent"/>

<SolidColorBrush x:Key="ComboBox.Disabled.Editable.Button.Border" Color="Transparent"/>

<Style x:Key="ComboBoxToggleButton" TargetType="{x:Type ToggleButton}">

<Setter Property="OverridesDefaultStyle" Value="true"/>

<Setter Property="IsTabStop" Value="false"/>

<Setter Property="Focusable" Value="false"/>

<Setter Property="ClickMode" Value="Press"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type ToggleButton}">

<Border x:Name="templateRoot" Background="{StaticResource ComboBox.Static.Background}" BorderBrush="{StaticResource ComboBox.Static.Border}" BorderThickness="{TemplateBinding BorderThickness}" SnapsToDevicePixels="true" CornerRadius="6">

<Border x:Name="splitBorder" BorderBrush="Transparent" BorderThickness="1" HorizontalAlignment="Right" Margin="0" SnapsToDevicePixels="true" Width="{DynamicResource {x:Static SystemParameters.VerticalScrollBarWidthKey}}">

<Path x:Name="arrow" Data="F1 M 0,0 L 2.667,2.66665 L 5.3334,0 L 5.3334,-1.78168 L 2.6667,0.88501 L0,-1.78168 L0,0 Z" Fill="{StaticResource ComboBox.Static.Glyph}" HorizontalAlignment="Center" Margin="0" VerticalAlignment="Center"/>

</Border>

</Border>

<ControlTemplate.Triggers>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="true"/>

<Condition Binding="{Binding IsMouseOver, RelativeSource={RelativeSource Self}}" Value="false"/>

<Condition Binding="{Binding IsPressed, RelativeSource={RelativeSource Self}}" Value="false"/>

<Condition Binding="{Binding IsEnabled, RelativeSource={RelativeSource Self}}" Value="true"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.Static.Editable.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.Static.Editable.Border}"/>

<Setter Property="Background" TargetName="splitBorder" Value="{StaticResource ComboBox.Static.Editable.Button.Background}"/>

<Setter Property="BorderBrush" TargetName="splitBorder" Value="{StaticResource ComboBox.Static.Editable.Button.Border}"/>

</MultiDataTrigger>

<Trigger Property="IsMouseOver" Value="true">

<Setter Property="Fill" TargetName="arrow" Value="{StaticResource ComboBox.MouseOver.Glyph}"/>

</Trigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsMouseOver, RelativeSource={RelativeSource Self}}" Value="true"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="false"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.MouseOver.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.MouseOver.Border}"/>

</MultiDataTrigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsMouseOver, RelativeSource={RelativeSource Self}}" Value="true"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="true"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.MouseOver.Editable.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.MouseOver.Editable.Border}"/>

<Setter Property="Background" TargetName="splitBorder" Value="{StaticResource ComboBox.MouseOver.Editable.Button.Background}"/>

<Setter Property="BorderBrush" TargetName="splitBorder" Value="{StaticResource ComboBox.MouseOver.Editable.Button.Border}"/>

</MultiDataTrigger>

<Trigger Property="IsPressed" Value="true">

<Setter Property="Fill" TargetName="arrow" Value="{StaticResource ComboBox.Pressed.Glyph}"/>

</Trigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsPressed, RelativeSource={RelativeSource Self}}" Value="true"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="false"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.Pressed.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.Pressed.Border}"/>

</MultiDataTrigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsPressed, RelativeSource={RelativeSource Self}}" Value="true"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="true"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.Pressed.Editable.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.Pressed.Editable.Border}"/>

<Setter Property="Background" TargetName="splitBorder" Value="{StaticResource ComboBox.Pressed.Editable.Button.Background}"/>

<Setter Property="BorderBrush" TargetName="splitBorder" Value="{StaticResource ComboBox.Pressed.Editable.Button.Border}"/>

</MultiDataTrigger>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Fill" TargetName="arrow" Value="{StaticResource ComboBox.Disabled.Glyph}"/>

</Trigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsEnabled, RelativeSource={RelativeSource Self}}" Value="false"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="false"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.Disabled.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.Disabled.Border}"/>

</MultiDataTrigger>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding IsEnabled, RelativeSource={RelativeSource Self}}" Value="false"/>

<Condition Binding="{Binding IsEditable, RelativeSource={RelativeSource AncestorType={x:Type ComboBox}}}" Value="true"/>

</MultiDataTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="{StaticResource ComboBox.Disabled.Editable.Background}"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="{StaticResource ComboBox.Disabled.Editable.Border}"/>

<Setter Property="Background" TargetName="splitBorder" Value="{StaticResource ComboBox.Disabled.Editable.Button.Background}"/>

<Setter Property="BorderBrush" TargetName="splitBorder" Value="{StaticResource ComboBox.Disabled.Editable.Button.Border}"/>

</MultiDataTrigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<ControlTemplate x:Key="ComboBoxEditableTemplate" TargetType="{x:Type ComboBox}">

<Grid x:Name="templateRoot" SnapsToDevicePixels="true">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition MinWidth="{DynamicResource {x:Static SystemParameters.VerticalScrollBarWidthKey}}" Width="0"/>

</Grid.ColumnDefinitions>

<Popup x:Name="PART\_Popup" AllowsTransparency="true" Grid.ColumnSpan="2" IsOpen="{Binding IsDropDownOpen, RelativeSource={RelativeSource TemplatedParent}}" Placement="Bottom" PopupAnimation="{DynamicResource {x:Static SystemParameters.ComboBoxPopupAnimationKey}}">

<theme:SystemDropShadowChrome x:Name="shadow" Color="Transparent" MinWidth="{Binding ActualWidth, ElementName=templateRoot}" MaxHeight="{TemplateBinding MaxDropDownHeight}">

<Border x:Name="dropDownBorder" Background="{DynamicResource {x:Static SystemColors.WindowBrushKey}}" BorderBrush="{DynamicResource {x:Static SystemColors.WindowFrameBrushKey}}" BorderThickness="1">

<ScrollViewer x:Name="DropDownScrollViewer">

<Grid x:Name="grid" RenderOptions.ClearTypeHint="Enabled">

<Canvas x:Name="canvas" HorizontalAlignment="Left" Height="0" VerticalAlignment="Top" Width="0">

<Rectangle x:Name="opaqueRect" Fill="{Binding Background, ElementName=dropDownBorder}" Height="{Binding ActualHeight, ElementName=dropDownBorder}" Width="{Binding ActualWidth, ElementName=dropDownBorder}"/>

</Canvas>

<ItemsPresenter x:Name="ItemsPresenter" KeyboardNavigation.DirectionalNavigation="Contained" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"/>

</Grid>

</ScrollViewer>

</Border>

</theme:SystemDropShadowChrome>

</Popup>

<ToggleButton x:Name="toggleButton" Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Grid.ColumnSpan="2" IsChecked="{Binding IsDropDownOpen, Mode=TwoWay, RelativeSource={RelativeSource TemplatedParent}}" Style="{StaticResource ComboBoxToggleButton}"/>

<Border x:Name="border" Background="{StaticResource TextBox.Static.Background}" Margin="{TemplateBinding BorderThickness}">

<TextBox x:Name="PART\_EditableTextBox" HorizontalContentAlignment="{TemplateBinding HorizontalContentAlignment}" IsReadOnly="{Binding IsReadOnly, RelativeSource={RelativeSource TemplatedParent}}" Margin="{TemplateBinding Padding}" Style="{StaticResource ComboBoxEditableTextBox}" VerticalContentAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Border>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Opacity" TargetName="border" Value="0.56"/>

</Trigger>

<Trigger Property="IsKeyboardFocusWithin" Value="true">

<Setter Property="Foreground" Value="Black"/>

</Trigger>

<Trigger Property="HasDropShadow" SourceName="PART\_Popup" Value="true">

<Setter Property="Margin" TargetName="shadow" Value="0,0,5,5"/>

<Setter Property="Color" TargetName="shadow" Value="#71000000"/>

</Trigger>

<Trigger Property="HasItems" Value="false">

<Setter Property="Height" TargetName="dropDownBorder" Value="95"/>

</Trigger>

<MultiTrigger>

<MultiTrigger.Conditions>

<Condition Property="IsGrouping" Value="true"/>

<Condition Property="VirtualizingPanel.IsVirtualizingWhenGrouping" Value="false"/>

</MultiTrigger.Conditions>

<Setter Property="ScrollViewer.CanContentScroll" Value="false"/>

</MultiTrigger>

<Trigger Property="ScrollViewer.CanContentScroll" SourceName="DropDownScrollViewer" Value="false">

<Setter Property="Canvas.Top" TargetName="opaqueRect" Value="{Binding VerticalOffset, ElementName=DropDownScrollViewer}"/>

<Setter Property="Canvas.Left" TargetName="opaqueRect" Value="{Binding HorizontalOffset, ElementName=DropDownScrollViewer}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

<ControlTemplate x:Key="ComboBoxTemplate" TargetType="{x:Type ComboBox}">

<Grid x:Name="templateRoot" SnapsToDevicePixels="true">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition MinWidth="{DynamicResource {x:Static SystemParameters.VerticalScrollBarWidthKey}}" Width="0"/>

</Grid.ColumnDefinitions>

<Popup x:Name="PART\_Popup" AllowsTransparency="true" Grid.ColumnSpan="2" IsOpen="{Binding IsDropDownOpen, Mode=TwoWay, RelativeSource={RelativeSource TemplatedParent}}" Margin="1" Placement="Bottom" PopupAnimation="{DynamicResource {x:Static SystemParameters.ComboBoxPopupAnimationKey}}">

<theme:SystemDropShadowChrome x:Name="shadow" Color="Transparent" MinWidth="{Binding ActualWidth, ElementName=templateRoot}" MaxHeight="{TemplateBinding MaxDropDownHeight}">

<Border x:Name="dropDownBorder" Background="{DynamicResource {x:Static SystemColors.WindowBrushKey}}" BorderBrush="{DynamicResource {x:Static SystemColors.WindowFrameBrushKey}}" BorderThickness="1">

<ScrollViewer x:Name="DropDownScrollViewer">

<Grid x:Name="grid" RenderOptions.ClearTypeHint="Enabled">

<Canvas x:Name="canvas" HorizontalAlignment="Left" Height="0" VerticalAlignment="Top" Width="0">

<Rectangle x:Name="opaqueRect" Fill="{Binding Background, ElementName=dropDownBorder}" Height="{Binding ActualHeight, ElementName=dropDownBorder}" Width="{Binding ActualWidth, ElementName=dropDownBorder}"/>

</Canvas>

<ItemsPresenter x:Name="ItemsPresenter" KeyboardNavigation.DirectionalNavigation="Contained" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"/>

</Grid>

</ScrollViewer>

</Border>

</theme:SystemDropShadowChrome>

</Popup>

<ToggleButton x:Name="toggleButton" Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Grid.ColumnSpan="2" IsChecked="{Binding IsDropDownOpen, Mode=TwoWay, RelativeSource={RelativeSource TemplatedParent}}" Style="{StaticResource ComboBoxToggleButton}"/>

<ContentPresenter x:Name="contentPresenter" ContentStringFormat="{TemplateBinding SelectionBoxItemStringFormat}" ContentTemplate="{TemplateBinding SelectionBoxItemTemplate}" Content="{TemplateBinding SelectionBoxItem}" ContentTemplateSelector="{TemplateBinding ItemTemplateSelector}" HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" IsHitTestVisible="false" Margin="{TemplateBinding Padding}" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="HasDropShadow" SourceName="PART\_Popup" Value="true">

<Setter Property="Margin" TargetName="shadow" Value="0,0,5,5"/>

<Setter Property="Color" TargetName="shadow" Value="#71000000"/>

</Trigger>

<Trigger Property="HasItems" Value="false">

<Setter Property="Height" TargetName="dropDownBorder" Value="95"/>

</Trigger>

<MultiTrigger>

<MultiTrigger.Conditions>

<Condition Property="IsGrouping" Value="true"/>

<Condition Property="VirtualizingPanel.IsVirtualizingWhenGrouping" Value="false"/>

</MultiTrigger.Conditions>

<Setter Property="ScrollViewer.CanContentScroll" Value="false"/>

</MultiTrigger>

<Trigger Property="ScrollViewer.CanContentScroll" SourceName="DropDownScrollViewer" Value="false">

<Setter Property="Canvas.Top" TargetName="opaqueRect" Value="{Binding VerticalOffset, ElementName=DropDownScrollViewer}"/>

<Setter Property="Canvas.Left" TargetName="opaqueRect" Value="{Binding HorizontalOffset, ElementName=DropDownScrollViewer}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

<Style TargetType="{x:Type ComboBox}">

<Setter Property="FocusVisualStyle" Value="{StaticResource FocusVisual}"/>

<Setter Property="Background" Value="{StaticResource ComboBox.Static.Background}"/>

<Setter Property="BorderBrush" Value="{StaticResource ComboBox.Static.Border}"/>

<Setter Property="Foreground" Value="{DynamicResource {x:Static SystemColors.WindowTextBrushKey}}"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="ScrollViewer.HorizontalScrollBarVisibility" Value="Auto"/>

<Setter Property="ScrollViewer.VerticalScrollBarVisibility" Value="Auto"/>

<Setter Property="Padding" Value="6,3,5,3"/>

<Setter Property="ScrollViewer.CanContentScroll" Value="true"/>

<Setter Property="ScrollViewer.PanningMode" Value="Both"/>

<Setter Property="Stylus.IsFlicksEnabled" Value="False"/>

<Setter Property="Template" Value="{StaticResource ComboBoxTemplate}"/>

<Style.Triggers>

<Trigger Property="IsEditable" Value="true">

<Setter Property="IsTabStop" Value="false"/>

<Setter Property="Padding" Value="2"/>

<Setter Property="Template" Value="{StaticResource ComboBoxEditableTemplate}"/>

</Trigger>

</Style.Triggers>

</Style>

<!--#endregion-->

</ResourceDictionary>

**Листинг модельный классов**

**Листинг класса Checks (первая часть)**

//------------------------------------------------------------------------------

// <auto-generated>

// Этот код создан по шаблону.

//

// Изменения, вносимые в этот файл вручную, могут привести к непредвиденной работе приложения.

// Изменения, вносимые в этот файл вручную, будут перезаписаны при повторном создании кода.

// </auto-generated>

//------------------------------------------------------------------------------

namespace Cafe.Models

{

using System;

using System.Collections.Generic;

public partial class Checks

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Checks()

{

this.Purchases = new HashSet<Purchases>();

}

public long CheckID { get; set; }

public decimal Cost { get; set; }

public decimal Discount { get; set; }

public decimal CostNDiscount { get; set; }

public bool IsComplexLunch { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<Purchases> Purchases { get; set; }

}

}

**Листинг класса Checks (вторая часть)**

namespace Cafe.Models

{

/// <summary>

/// Вторая часть класса чеков

/// </summary>

public partial class Checks

{

public string IsComplexLunchString { get => (IsComplexLunch) ? "Да" : "Нет"; }

}

}

**Листинг класса Purchases (первая часть)**

//------------------------------------------------------------------------------

// <auto-generated>

// Этот код создан по шаблону.

//

// Изменения, вносимые в этот файл вручную, могут привести к непредвиденной работе приложения.

// Изменения, вносимые в этот файл вручную, будут перезаписаны при повторном создании кода.

// </auto-generated>

//------------------------------------------------------------------------------

namespace Cafe.Models

{

using System;

using System.Collections.Generic;

public partial class Purchases

{

public long PurchaseID { get; set; }

public long CheckID { get; set; }

public int ProductID { get; set; }

public int ProductCount { get; set; }

public virtual Checks Checks { get; set; }

public virtual Products Products { get; set; }

}

}

**Листинг класса Purchases (вторая часть)**

namespace Cafe.Models

{

/// <summary>

/// Вторая часть классов покупок

/// </summary>

public partial class Purchases

{

public decimal ProductCost { get => (Products != null) ? Products.Price \* ProductCount : 0; }

}

}

**Листинг класса Measures**

//------------------------------------------------------------------------------

// <auto-generated>

// Этот код создан по шаблону.

//

// Изменения, вносимые в этот файл вручную, могут привести к непредвиденной работе приложения.

// Изменения, вносимые в этот файл вручную, будут перезаписаны при повторном создании кода.

// </auto-generated>

//------------------------------------------------------------------------------

namespace Cafe.Models

{

using System;

using System.Collections.Generic;

public partial class Measures

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Measures()

{

this.Products = new HashSet<Products>();

}

public int MeasureID { get; set; }

public string Name { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<Products> Products { get; set; }

}

}

**Листинг класса Products**

//------------------------------------------------------------------------------

// <auto-generated>

// Этот код создан по шаблону.

//

// Изменения, вносимые в этот файл вручную, могут привести к непредвиденной работе приложения.

// Изменения, вносимые в этот файл вручную, будут перезаписаны при повторном создании кода.

// </auto-generated>

//------------------------------------------------------------------------------

namespace Cafe.Models

{

using System;

using System.Collections.Generic;

public partial class Products

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Products()

{

this.Purchases = new HashSet<Purchases>();

}

public int ProductID { get; set; }

public string Name { get; set; }

public decimal Price { get; set; }

public int HoldCount { get; set; }

public int MeasureID { get; set; }

public virtual Measures Measures { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<Purchases> Purchases { get; set; }

}

}

# **ПРИЛОЖЕНИЕ Б**

**Листинг тестовых модулей**

**Листинг тестового класса CheckActionsUnitTest**

using Microsoft.VisualStudio.TestTools.UnitTesting;

using Cafe.Tools;

using System;

namespace CafeUnitTests

{

[TestClass]

public class CheckActionsUnitTest

{

[TestMethod]

public void \_\_1TestHasHoldCount()//Проверка работы спуска до 0 остатков

{

Assert.IsTrue(CheckActions.HasHoldCount(new Cafe.Models.Products()

{

HoldCount = 2//задаем остатки

}, 2));//задаем количество для списания

}

[TestMethod]

public void \_\_2TestHasHoldCount()//Проверка работы с остатками больше 0

{

Assert.IsTrue(CheckActions.HasHoldCount(new Cafe.Models.Products()

{

HoldCount = 3//задаем остатки

}, 2));//задаем количество для списания

}

[TestMethod]

public void \_\_3TestHasHoldCount()//Проверка работы с остатками меньше 0

{

try

{

Assert.IsTrue(CheckActions.HasHoldCount(new Cafe.Models.Products()

{

HoldCount = 1//задаем остатки

}, 2));//задаем количество для списания

throw new Exception();//Если ошибка метода - будет исключение здесь

}

catch { }

}

[TestMethod]

public void \_\_4TestGetDiscount()//Проверка работы скидки до 300 р. (должно быть 0)

{

Assert.IsTrue(CheckActions.GetDiscount(299) == 0);

}

[TestMethod]

public void \_\_5TestGetDiscount()//Проверка работы скидки от 300 р. (должно быть 9)

{

Assert.IsTrue(CheckActions.GetDiscount(300) == 9);

}

[TestMethod]

public void \_\_6TestGetDiscount()//Проверка работы скидки до 500 р. (должно быть 15)

{

Assert.IsTrue(CheckActions.GetDiscount(500) == 15);

}

[TestMethod]

public void \_\_7TestGetDiscount()//Проверка работы скидки от 501 р. (должно быть 25,05)

{

Assert.IsTrue(CheckActions.GetDiscount(501) == Convert.ToDecimal(25.05));

}

[TestMethod]

public void \_\_8TestGetDiscount()//Проверка работы скидки до 1000 р. (должно быть 50)

{

Assert.IsTrue(CheckActions.GetDiscount(1000) == 50);

}

[TestMethod]

public void \_\_9TestGetDiscount()//Проверка работы скидки от 1001 р. (должно быть 70,07)

{

Assert.IsTrue(CheckActions.GetDiscount(1001) == Convert.ToDecimal(70.07));

}

[TestMethod]

public void \_10TestGetDiscount()//Проверка работы скидки до 5000 р. (должно быть 350)

{

Assert.IsTrue(CheckActions.GetDiscount(5000) == 350);

}

[TestMethod]

public void \_11TestGetDiscount()//Проверка работы скидки от 5001 р. (должно быть 500,1)

{

Assert.IsTrue(CheckActions.GetDiscount(5001) == Convert.ToDecimal(500.1));

}

}

}