Областное государственное бюджетное профессиональное образовательное учреждение

«КОСТРОМСКОЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность: 09.02.07 «Информационные системы и программирование»

**Курсовая работа**

**по дисциплине «МДК 02.01 Технология разработки программного обеспечения»**

КП-02069846-09.02.07-11-81

Проверил(а):\_\_\_\_\_\_\_\_ Э.М. Павлова

Выполнил(а): \_\_\_\_\_\_\_\_ И. С. Котов

«11» декабря 2023 г.

Кострома, 2023

**Оглавление**

[ВВЕДЕНИЕ 4](#_heading=h.ywwpadw6pbfo)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_heading=h.9a5vv0x7s2cz)

[1.1 Техническое задание на разработку программного продукта 7](#_heading=h.ek6e4dyqvtnc)

[1.2 Организация ввода-вывода и пользовательского интерфейса 7](#_heading=h.b1of3qaotfvz)

[1.2.1 Входные данные 7](#_heading=h.1vxc607go7ua)

[1.2.2. Выходные данные 7](#_heading=h.x35dlu9rfp9k)

[1.2.3. Организация пользовательского интерфейса 8](#_heading=h.izxyns3xbc1p)

[2. СПЕЦИАЛЬНАЯ ЧАСТЬ 9](#_heading=h.oczl5o61a3d8)

[2.1 Выбор и обоснование языка программирования 10](#_heading=h.8lbxndgz64ot)

[2.2 Выбор и обоснование технологий и инструментов 10](#_heading=h.2o7upwi6tmak)

[2.3 Выбор стиля, методов и средств программирования 10](#_heading=h.l4rzurf8btt9)

[2.4 Описание программы 11](#_heading=h.wpf2lct36g1m)

[2.4.1 Общие сведения 11](#_heading=h.vc4fmmxolyjc)

[2.4.2 Функциональное назначение программы 12](#_heading=h.z9uw36l2aprx)

[2.4.3 Входные и выходные данные 12](#_heading=h.6w4myvvj00f8)

[2.4.4 Используемые технические средства 13](#_heading=h.1696thk707j1)

[2.4.5 Вызов и загрузка 13](#_heading=h.clbjbfrj86xl)

[2.5 Тестирование программы 14](#_heading=h.cmji7l1yxxgu)

[2.5.1 Стратегия проведения испытаний программного продукта 14](#_heading=h.ymvhmbgzph09)

[2.5.2 Этапы проведения испытаний программного продукта 14](#_heading=h.2ehel8ddl9i0)

[3. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ 15](#_heading=h.cydr1x7roc9h)

[1 Общие сведения 16](#_heading=h.35nkun2)

[2 Наиболее распространенные угрозы 16](#_heading=h.44sinio)

[3 Основные угрозы конфиденциальности 16](#_heading=h.2jxsxqh)

[4 Используемые методы информационной безопасности 17](#_heading=h.84bybkckadik)

[ЗАКЛЮЧЕНИЕ 18](#_heading=h.u9w1n4bkor22)

[Список литературы 20](#_heading=h.nqatmjp0vuxd)

[ПРИЛОЖЕНИЯ 21](#_heading=h.30j0zll)

[ПРИЛОЖЕНИЕ 1 22](#_heading=h.seww1g21h56p)

[1. Введение 23](#_heading=h.dae68mvwg0p8)

[2. Основание для разработки 23](#_heading=h.4mu2iawvgxsq)

[3. Назначение разработки 24](#_heading=h.5s01tohll1ew)

[4. Требования к программному продукту 24](#_heading=h.wwsrlhzfk9jl)

[4.1. Требования к функциональным характеристикам 24](#_heading=h.sauz9r1vvhzn)

[4.2. Требования к надежности 25](#_heading=h.v8qmxsf0lmlz)

[4.3. Условия эксплуатации 25](#_heading=h.h543y85ra7j1)

[4.4. Требования к составу и параметрам технических средств 25](#_heading=h.w2zdd57kpdnr)

[4.5. Требования к информационной и программной совместимости 26](#_heading=h.8xz9c0rshv3s)

[4.6. Требования к передаче программного продукта 26](#_heading=h.f254kf1cmou)

[5. Требования к программной документации 27](#_heading=h.xa80fgaqckkj)

[ПРИЛОЖЕНИЕ 2 28](#_heading=h.bnxs1ci6jff9)

[1. Назначение и условия применения программы 29](#_heading=h.framaopzt737)

[2. Характеристика программы 29](#_heading=h.xyskvikkairk)

[3. Обращение к программе 29](#_heading=h.icocpsqw0jm6)

[4. Входные и выходные данные 30](#_heading=h.ib5jyih53muu)

[5. Сообщение 30](#_heading=h.14t9bp6m441p)

[ПРИЛОЖЕНИЕ 3 31](#_heading=h.k601xjx742qz)

[1. Назначение программы 32](#_heading=h.wi5lyckpuly4)

[2. Условия выполнения программы 32](#_heading=h.fev9zi9qtb0p)

[3. Выполнения программы 32](#_heading=h.gnoaz5gmjk3i)

[4. Сообщение оператору 33](#_heading=h.6q87pbns09nd)

[ПРИЛОЖЕНИЕ 4 34](#_heading=h.tn466gcrl0i0)

[1. Назначение программы 35](#_heading=h.mkt2tdvx9f7k)

[2. Запуск программы 35](#_heading=h.72a47ujz2p0g)

[3. Объем программного продукта 36](#_heading=h.avcrk726rbtb)

[4. Специальные указания оператору 36](#_heading=h.e829tn7a5qte)

[6. Дата создания 36](#_heading=h.5wth1015e23v)

[ПРИЛОЖЕНИЕ 5 37](#_heading=h.utga4ehjqzqy)

[ПРИЛОЖЕНИЕ 6 43](#_heading=h.6iehil6502e9)

# ВВЕДЕНИЕ

|  |  |
| --- | --- |
|  |  |
|  | |
|  | |
|  | |

Проверил(а):\_\_\_\_\_\_\_\_ Э.М. Павлова

Выполнил(а): \_\_\_\_\_\_\_\_ И.С. Котов

«11» декабря 2023 г.

Выбранная предметная область – аптека. Интерес к предметной области вызван тем, что эта сфера всегда пользуется спросом и никогда не потеряет своей актуальности.

Актуальность работы заключается в том, что разработка настольного приложения и управление им на сегодняшний день для большинства организаций становится необходимостью. Приложение выступает как средство коммуникации и помощи целевой аудитории. Он является важнейшим источником информации, который поможет улучшить выбор лекарств и медикаментов для многих.

За счет создания этого приложения компания получит увеличение клиентов, оказанных услуг и повышение конкурентоспособности.

Цель работы: разработка программного обеспечения для сети аптек. Анализ требований и информационных потребностей;

Для достижения цели поставлены следующие задачи:

1. Разработать дизайн и пользовательский интерфейс, который будет легко работать для пользователей;

2. Создать программу с минимально необходимым функционалом;

3. Провести тщательное тестирование приложения на различных устройствах и операционных системах;

4. Сделать анализ рынка, расчет финансов.

Объект – сфера сети аптек. Предмет исследования – настольное приложение для аптеки.

Практическая значимость работы заключается в том, что её можно применить в реальной жизни для создания своей сети аптек. Способствует повышению клиентской части, продаж оказанных услуг и прибыли аптеки.

# 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Руководитель: \_\_\_\_\_\_\_\_ Э.М. Павлова

Выполнил: \_\_\_\_\_\_\_\_ И.С. Котов

«11» декабря 2023 г.

## 1.1 Техническое задание на разработку программного продукта

В данной работе основанием для разработки программного продукта является задание на курсовую работу по предмету “Технология разработка программного обеспечения”, выданное преподавателем ОГБПОУ “Костромской политехнический колледж” Павловой Э.М

## 1.2 Организация ввода-вывода и пользовательского интерфейса

### 1.2.1 Входные данные

Входной информацией являются следующие данные для пользователя:

● ФИО клиента;

● Дата рождения;

● Логин;

● Пароль;

● Пол.

### 1.2.2. Выходные данные

Выходной информацией являются следующие данные:

● Специалист по данной услуге;

● Стоимость товара;

● Счет списания;

● Скидочная карта;

● Название товара;

● Адрес аптеки;

● История покупок.

### 1.2.3. Организация пользовательского интерфейса

Программное обеспечение имеет графический интерфейс.Последовательность появления форм у пользователя в соответствии с выполняемыми задачами. Каждая форма должна иметь: название, которое однозначно определяет ее назначение; привлекательный внешний вид; блок для ввода данных в таблицы или параметров и т.д.

Элементы управления, с помощью которых организуется взаимодействие программы с пользователем:

1. Меню;

2. Диалоговые окна;

3. Кнопки;

4. Списки;

Разработка форм может производиться различными средствами визуального проектирования, например :

● с помощью языков программирования;

● с помощью специальных компонентов СУБД.

# 2. СПЕЦИАЛЬНАЯ ЧАСТЬ

Руководитель: \_\_\_\_\_\_\_\_ Э.М. Павлова

Выполнил: \_\_\_\_\_\_\_\_ И.С. Котов «11» декабря 2023 г.

## 2.1 Выбор и обоснование языка программирования

В качестве средства разработки выбран объектно-ориентированный язык программирования – C#. Главным фактором языка C# является практичность. Он предоставляет средства для быстрого и эффективного решения поставленных задач.

## 2.2 Выбор и обоснование технологий и инструментов

Очень важным фактором, который необходимо будет учесть при разработке системы, является соответствие потребностей в ресурсах имеющемуся в аптеке техническому обеспечению. Для корректной работы системы, оборудование должно быть совместимо с ней.

Microsoft SQL Server – база данных менее уязвима, имеет инструменты бизнес-аналитики с поддержкой самообслуживания.

SQL – универсальна для работы с базой данных.

## 2.3 Выбор стиля, методов и средств программирования

Программа выполнена в стиле объектно-ориентированного программирования с элементами процедурного стиля.

Используемые среды разработки:

* Visual Studio – имеет расширяемую библиотеку дополнений и готовых решений, а также обладает мультифункциональностью;
* DB Browser для SQLite – это высококачественный визуальный инструмент с открытым исходным кодом для создания, дизайна и редактирования файлов базы данных, совместимый с SQLite;
* Git –почти все отрицательные черты CVS/SVN устранены + высокая скорость работы распределенной системы контроля версий, легкость проведения различных операций с ветками проекта.

## 2.4 Описание программы

### 2.4.1 Общие сведения

Аптека - это место, где можно приобрести лекарственные препараты и другие медицинские товары. Она обычно расположена рядом с медицинскими учреждениями или торговыми центрами. Основная цель аптеки - предоставить населению качественные и безопасные лекарства, а также предоставить консультацию фармацевта и дополнительные услуги, такие как измерение давления или приготовление рецептурных смесей.

В услуги аптеки входят:

● Продажа лекарств - это означает, что аптека предлагает широкий ассортимент лекарственных препаратов для лечения различных заболеваний;

● Консультации фармацевта - это означает, что клиенты могут получить профессиональные консультации по выбору и применению лекарств от квалифицированных специалистов;

● Продажа медицинских изделий и средств гигиены - это означает, что аптека предлагает широкий ассортимент медицинских товаров, таких как бинты, перевязочные материалы, средства гигиены и т.д.

● Услуги доставки лекарств - это означает, что аптека может предоставлять услугу доставки лекарственных препаратов клиентам, которым неудобно или невозможно самостоятельно приобрести лекарства;

### 2.4.2 Функциональное назначение программы

Основной вид деятельности аптеки заключается в оказании услуг клиентам и продажи медикаментов.

### 2.4.3 Входные и выходные данные

Входной информацией являются следующие данные для пользователя:

● ФИО клиента;

● Дата рождения;

● Логин;

● Пароль;

● Пол.

Выходной информацией являются следующие данные:

● Специалист по данной услуге;

● Стоимость товара;

● Счет списания;

● Скидочная карта;

● Название товара;

● Адрес аптеки;

● История покупок.

### 2.4.4 Используемые технические средства

Использование данного приложения будет предъявлять следующие требования к оборудованию и программному обеспечению:

● Операционная система Android или IOS;

● Необходимое свободное место для установки программного продукта 50 Mb;

### 2.4.5 Вызов и загрузка

Загрузка программного продукта осуществляется через Google Play Market и App Store, а также с официального сайта https://indigo.ru/dejavu. Для работы приложению необходимо подключение к сети Интернет и Excel.

## 2.5 Тестирование программы

### 2.5.1 Стратегия проведения испытаний программного продукта

Был использован статический метод тестирования – один из наиболее эффективных способов обнаружения дефектов на ранних стадиях разработки ПО. Для повышения эффективности тестирования использовались автоматизированные инструменты, такие как Selenium, чтобы ускорить процесс тестирования и позволить быстрее выявлять ошибки и дефекты.

### 2.5.2 Этапы проведения испытаний программного продукта

Были проделаны следующие действия:

* Планирование: в этом этапе определялись цели, область тестирования, требования к продукту и методы тестирования.
* Подготовка тестовых данных: подготовка необходимых данных для проведения тестирования.
* Оценка результатов тестирования: анализировались полученные результаты тестирования, выявлены проблемы и ошибки в работе программного продукта.
* Устранение ошибок: исправлялись ошибки и выполнялись доработки, необходимые для улучшения качества программного продукта.
* Запуск в эксплуатацию: программный продукт выпускается в эксплуатацию после проведения всех этапов тестирования и устранения всех ошибок.

# **3.** **ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ**

Руководитель: \_\_\_\_\_\_\_\_ Э.М. Павлова

Выполнили: \_\_\_\_\_\_\_\_ И.С. Котов «11» декабря 2023 г.

## 1 Общие сведения

Информационная безопасность – это всесторонняя защищённость информации и поддерживающей её инфраструктуры от любых случайных или злонамеренных воздействий, результатом которых может явиться нанесение ущерба самой информации, ее владельцам или поддерживающей инфраструктуре.

 Целью информационной безопасности является минимизация ущерба, а также прогнозирование и предотвращение таких воздействий.

## 2 Наиболее распространенные угрозы

Наиболее распространенные угрозы в аптеке:

* Низкая защищенность данных, особенно если приложение требует доступ к личной информации пользователей, такой как данные о здоровье и финансовые сведения;
* Несанкционированное использование приложения или его функций, например, для проведения мошеннических операций или нарушения правил пользования приложения;
* Повышение стоимости сырья (косметические средства, инструмент, реклама).

## 3 Основные угрозы конфиденциальности

Основными угрозами конфиденциальности являются:

* Потеря персональных данных (ФИО, номер телефона, данные счета карты);
* Кража персональных данных (ФИО, номер телефона, данные счета карты).

## 4 Используемые методы информационной безопасности

В целях обеспечения конфиденциальности информации и защиты персональных данных компания аптеки поддерживает соответствующую ИТ-среду и принимает все меры, необходимые для предотвращения несанкционированного доступа (хакерства).

1. Для использования персональных данных, для любой иной цели будет создан запрос “Согласие на обработку персональных данных”.
2. Предоставлять каждому сотруднику минимально необходимый уровень доступа к данным – ровно столько, сколько ему нужно для выполнения должностных обязанностей. Этот принцип позволяет избежать многих проблем, таких как утечка конфиденциальных данных, удаление или искажение информации из-за нарушений в работе с ней и т. д. (технический метод);
3. Безопасная система оплаты. Сервис будет соответствовать платежному стандарту – PCI DSS.

# ЗАКЛЮЧЕНИЕ

Руководитель: \_\_\_\_\_\_\_\_ Э.М. Павлова

Выполнили: \_\_\_\_\_\_\_\_ И.С. Котов

«11» декабря 2023 г.

В любой фирме, рано или поздно, возникает проблема такой организации управления данными, которая позволит обеспечить наиболее эффективную работу. Исходя из потребностей аптеки в автоматизации приема поступающих заявок и продвижении предоставляемых услуг, было разработано настольное приложение, которое позволяет:

* принимать, вводить, накапливать, хранить и редактировать информацию о поступивших от клиентов;
* информировать клиентов о предоставляемых услугах и акциях;
* информировать клиентов о сотрудниках аптеки;
* привлекать новых клиентов;
* продвигать аптеку в просторах сети Интернет.

Были выполнены все поставленные задачи и цели в полной мере. В частности, были реализованы функции, которые позволяют автоматизировать многие процессы в аптеке, снизить количество ошибок и повысить эффективность работы.

На данный момент настольное приложение требует небольшой доработки, его преимуществами являются:

* адаптивность под все устройства (персональный компьютер, планшетный компьютер, смартфон);
* легкость в навигации;
* не требует высоких знаний уровня использования приложения.

В общем, разработанное программное обеспечение значительно упростило и ускорило работу аптеке, повысило качество предоставляемых услуг и сократило время, необходимое для выполнения рутинных задач.

Цель данной работы – разработка программного обеспечения для аптеки, была выполнена.

# Список литературы

1. ГОСТ Р 51904-2002 Программное обеспечение встроенных систем. Общие требования к разработке и документированию: [Электронный ресурс] Скачать ГОСТ Р 51904-2002 Программное обеспечение встроенных систем. Общие требования к разработке и документированию. https://files.stroyinf.ru/Index2/1/4294817/4294817035.htm : (10.12.2023)​
2. ГОСТ 19.504-79. ЕСПД. Руководство программиста. Требования к содержанию и оформлению: [Электронный ресурс] ГОСТ 19.504-79. ЕСПД. Руководство программиста. Требования к содержанию и оформлению. <https://www.swrit.ru/doc/espd/19.504-79.pdf> : (10.12.2023)
3. ГОСТ 19.505-79. ЕСПД. Руководство оператора. Требования к содержанию и оформлению: [Электронный ресурс] ГОСТ 19.505-79. ЕСПД. Руководство оператора. Требования к содержанию и оформлению. <https://www.swrit.ru/doc/espd/19.505-79.pdf> : (10.12.2023)
4. Стратегия тестирования информационных систем управления вузом на основе документов с теговой разметкой: [Электронный ресурс] Стратегия тестирования информационных систем управления вузом на основе документов с теговой разметкой. <https://cyberleninka.ru/article/n/strategiya-testirovaniya-informatsionnyh-sistem-upravleniya-vuzom-na-osnove-dokumentov-s-tegovoy-razmetkoy/viewer> :  (10.12.2023)
5. Входные данные,Выходные данные: [Электронный ресурс] Входные данные, Выходные данные - Программное обеспечение информационных технологий. <https://studwood.net/1706755/informatika/vhodnye_dannye> : (10.12.2023)
6. Входные данные программы, Выходные данные программы: [Электронный ресурс] Входные данные программы, Выходные данные программы, Результаты тестирования программы, Инструкция оператору по работе с программой, Руководство пользователя, Руководство администратора, Выводы - Разработка программного комплекса для автоматизации информационных процессов служб сбыта пищевой продукции. <https://studbooks.net/2224777/informatika/vhodnye_dannye_programmy> : (10.12.2023)

# ПРИЛОЖЕНИЯ

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«КОСТРОМСКОЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 09.02.07 «Информационные системы и программирование»

| **УТВЕРЖДАЮ** |
| --- |
| Руководитель: Э.М. Павлова |
| «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

| **Техническое задание** |
| --- |
| ПРИЛОЖЕНИЕ 1 |
| КП-02069846-09.02.07-11-81  Программа на С# |
| Листов 5 |

| Выполнил: | \_\_\_\_\_\_\_\_\_\_И.С.Котов |
| --- | --- |

## 1. Введение

Выбранная предметная область – аптека. За счет создания настольного приложения, компания получит увеличение клиентов, оказанных услуг и повышение конкурентоспособности.

## 2. Основание для разработки

В данной работе основанием для разработки программного продукта является задание на курсовую работу по предмету “МДК 02.01 Технология разработки программного обеспечения”, выданное преподавателем ОГБПОУ “Костромской политехнический колледж” Павловой Э.М.

Стандарты, на основании которых ведется разработка программного продукта:

* ЕСПД ГОСТ 19.201-78. «Техническое задание. Требования к содержанию и оформлению»;
* ЕСПД ГОСТ 19.401-78. «Текст программы. Требования к содержанию и оформлению»;
* ЕСПД ГОСТ 19.404-78. «Пояснительная записка. Требования к содержанию и оформлению»;
* ЕСПД ГОСТ 19.301-79. «Программа и методика испытаний. Требования к содержанию и оформлению»;
* ЕСПД ГОСТ 19.502-78. «Описание применения. Требования к содержанию и оформлению»;
* ЕСПД ГОСТ 19.106-78. «Требования к программным документам, выполненным печатным способом».

Техническое задание представляет собой документ, в котором сформулированы основные цели разработки, требования к программному продукту, определены сроки и этапы разработки и регламентирован процесс приемосдаточных испытаний.

## 3. Назначение разработки

Предметом разработки является настольное приложение по услугам аптеки, функционирующий на основе системы динамического управления контентом.

Назначение настольного приложения: организация электронной торговли цифровыми услугами через сеть интернет.

## 4. Требования к программному продукту

### 4.1. Требования к функциональным характеристикам

Разработанная система должна отвечать следующим требованиям:

1)Предоставить посетителю возможность:

* просмотра ассортимента услуг, информации о аптеке;
* самостоятельно проходить процесс регистрации и иметь личный профиль в приложении;
* осуществлять поиск информации;
* осуществлять процесс оформления заказа с указанием контактных данных клиента и способа оплаты услуг;
* оставлять отзывы на каждый из услуг, а также в общем о аптеке;
* делиться ссылкой на описание услуги через социальные сети;
* осуществлять рассылку новостной информации на электронные ящики пользователей;
* обеспечивать взаимосвязь с администрацией посредством формы обратной связи;

2) Предоставить администратору возможность:

* добавлять, удалять и редактировать содержимое каталога услуг и другой информации, отображаемой на страницах мобильного приложения;
* просмотра информации о клиентах и их выборе;
* производить рассылку новостной информации;
* получать денежные средства от клиентов через платежную систему;
* рецензировать отзывы клиентов;
* изменения настроек приложения (цветового оформления, информации об организации, модулей и др.);

### 4.2. Требования к надежности

В целях надежности обучающей системы она должна удовлетворять следующим требованиям:

* разработанная программа должна обладать средствами защиты от ошибочных действий пользователей;
* все ошибки должны отображаться с комментариями или подсказками по их устранению;
* гарантировать сохранность данных при сбоях в работе внешних устройств.

Для повышения надежности необходимо принять следующие меры:

* сконфигурировать аппаратные и программные средства в соответствии с техническими требованиями;
* периодически осуществлять резервное копирование информации;
* регулярно проверять целостность базы данных;
* поддерживать исправность сетевого оборудования.

### 4.3. Условия эксплуатации

Скачать приложение на мобильное устройство на сервисе приложений Google Play Market или AppStore, или на официальном сайте, после запустить программу.

### 4.4. Требования к составу и параметрам технических средств

Исходные коды программы должны быть написаны на языке C#.

Для мобильного телефона необходимо:

* операционная система Android или iOS;
* необходимое свободное место для установки программного продукта 50Mb;
* подключение к сети Интернет;
* Excel.

### 4.5. Требования к информационной и программной совместимости

С целью обеспечения информационной совместимости, а также для однозначного описания данных, сокращении объёмов при хранении, обеспечения эффективного поиска и идентификации данных при разработке настольного приложения должны использоваться действующие в аптеке классификаторы.

В общем виде классификатор содержит наименование понятия и его код. Коды могут быть простые и составные.

Простой код представляет собой порядковый номер понятия в соответствующей области определения.

Составной код имеет составляющие, которые занимают определённые позиции кода.

### 4.6. Требования к передаче программного продукта

По требованию заказчика данное программное обеспечение разрабатывается под платформу Android, начиная с  4.1. или 5.0, и заканчивая Android 12, 11 и 10. Программа должна работать под основными версиями платформы iOS.

## 5. Требования к программной документации

Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа.

Разработчик передает заказчику документацию и программное обеспечение в сроки, определяемые настоящим техническим заданием.

Техническое задание, титульные листы других документов должны быть подписаны академическим руководителем образовательной программы «МДК 02.01 Технология разработки программного обеспечения».

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«КОСТРОМСКОЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 09.02.07 «Информационные системы и программирование»

| **УТВЕРЖДАЮ** |
| --- |
| Руководитель: Э.М. Павлова |
| «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

| **Руководство программиста** |
| --- |
| ПРИЛОЖЕНИЕ 2 |
| КП-02069846-09.02.07-11-81  Программа на С# |
| Листов 2 |

| Выполнил: | \_\_\_\_\_\_\_\_ И.С. Котов |
| --- | --- |

## 1. Назначение и условия применения программы

Количество необходимой оперативной памяти определяется размером обрабатываемых данных

Требования к программному обеспечению:

Операционная система Android или iOS.

## 2. Характеристика программы

Характеристики программ для настольного приложения аптеки:

1. Работоспособность - программа должна быть полностью функциональной и позволять клиентам получать доступ ко всем услугам, представленным в приложении.

2. Удобность использования - программа должна облегчать каждый шаг, который клиенты предпринимают, используя мобильное приложение.

3. Скорость работы - программа должна загружаться быстро и не замедлять работу мобильного приложения.

4. Безопасность - программа должна обеспечивать защиту персональных данных клиента и сохранность финансовых средств.

5. Поддержка различных платежных систем - программа должна поддерживать различные способы оплаты, чтобы клиенты могли выбирать наиболее удобный способ оплаты для себя.

## 3. Обращение к программе

Обращение к программе – настольное приложение.

Описание процедур:

* Бронирование услуг с помощью онлайн-формы или контактной формы по электронной почте;
* Регистрация новых клиентов и сохранение их профиля в базу данных;
* Уведомление клиентов о подтверждении приобретения услуг и о напоминании даты и времени прибытия в аптеку.
* Передача параметров производится в режиме диалога программы с пользователем.

## 4. Входные и выходные данные

Входной информацией являются следующие данные для пользователя:

* ФИО клиента;
* Пол;
* Логин;
* Пароль;
* Дата рождения;

Выходной информацией являются следующие данные:

* Специалист по данной услуге;
* Стоимость товара;
* Счет списания;
* Скидочная карта;
* Название товара;
* Адрес аптеки;
* История покупок.

## 5. Сообщение

Визуально работоспособность программы программист может проверить следующим способом:

* при щелчке по пунктам меню происходят соответствующие действия;
* выдаются все предупреждения и сообщения об ошибках.

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«КОСТРОМСКОЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 09.02.07 «Информационные системы и программирование»

| **УТВЕРЖДАЮ** |
| --- |
| Руководитель: Э.М. Павлова |
| «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

| **Руководство оператора** |
| --- |
| ПРИЛОЖЕНИЕ 3 |
| КП-02069846-09.02.07-11-81  Программа на С# |
| Листов 2 |

| Выполнил: | \_\_\_\_\_\_\_\_ И.С.Котов |
| --- | --- |

## 1. Назначение программы

Назначение приложения: организация электронной торговли цифровыми услугами через сеть интернет или мобильные устройства.

Приложение предназначено для решения элементарных задач обработки данных, таких, как редактирование и сортировка.

Состав функций:

* Функция редактирования аккаунта;
* Функция удаления аккаунта;
* Функция обновления данных аккаунта;
* Функция сортировки данных.

## 2. Условия выполнения программы

Выполнение данного приложения будет предъявлять следующие требования к оборудованию и программному обеспечению:

* Операционная система Android или IOS;
* Необходимое свободное место для установки программного продукта 50 Mb.

## 3. Выполнения программы

Вызов и загрузка приложения осуществляется с помощью ввода в поисковой строке названия утилиты в магазине приложений Google Play Market или AppStore .

Закрытие приложения осуществляется по нажатию на кнопку “Закрыть”.

## 4. Сообщение оператору

Сообщение «Некорректный ввод выбираемой услуги»

Программа при добавлении или редактировании записи выдает сообщение об ошибке.

Причина: данное сообщение появляется, когда пользователь вводит для обозначения выбираемой услуги, отличные от русских букв.

Действия программы: программа не заносит ошибочные данные и повторно запрашивает у пользователя указать ту или иную услугу.

Действия оператора: убедиться в том, что вводятся именно русские строчные буквы , при необходимости сменить раскладку клавиатуры на русскоязычную.

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«КОСТРОМСКОЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 09.02.07 «Информационные системы и программирование»

| **УТВЕРЖДАЮ** |
| --- |
| Руководитель: Э.М. Павлова |
| «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

| **Паспорт программы** |
| --- |
| ПРИЛОЖЕНИЕ 4 |
| КП-02069846-09.02.07-11-81  Программа на C# |
| Листов 2 |

| Выполнил: | \_\_\_\_\_\_\_\_ И.С. Котов |
| --- | --- |

## 1. Назначение программы

Назначение: настольное приложение выступает как средство коммуникации привлечения целевой аудитории.

Требования к программе:

* Простота и удобство использования;
* Безопасность персональных данных пользователей;
* Доступность на различных устройствах и платформах;
* Быстрый и надежный доступ к данным;
* Отображение актуальной информации о доступных услугах.

Ожидаемые результаты:

* Улучшение удовлетворенности клиентов и повышения покупки качественных товаров;
* Уменьшение времени, затрачиваемого на обслуживание клиентов;
* Увеличение дохода аптеки.

## 2. Запуск программы

Для запуска приложения аптеки необходимо открыть Google Play Market и App Store, а также официальный сайт https://indigo.ru/dejavu. Для работы приложению необходимо подключение к сети Интернет.

## 3. Объем программного продукта

Вес настольного приложения на данный момент не известен.

## 4. Специальные указания оператору

Специальные указания оператору отсутствуют.

**5. Авторы**

Автор программы: Котов Илья Сергеевич. Контактная информация: группа – 3-2П9, учебное учреждение – Костромской политехнический колледж, почта – ilakotov7613@gmail.com

## 6. Дата создания

Техническая документация 09.12.23 -11.12.23

Создание программного продукта 01.12.23

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«КОСТРОМСКОЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 09.02.07 «Информационные системы и программирование»

| **УТВЕРЖДАЮ** |
| --- |
| Руководитель: Э.М. Павлова |
| «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

| **Пользовательские экраны** |
| --- |
| ПРИЛОЖЕНИЕ 5 |
| КП-02069846-09.02.07-11-81  Программа на С# |
| Листов 5 |

| Выполнил: | \_\_\_\_\_\_\_\_ И.С. Котов |
| --- | --- |

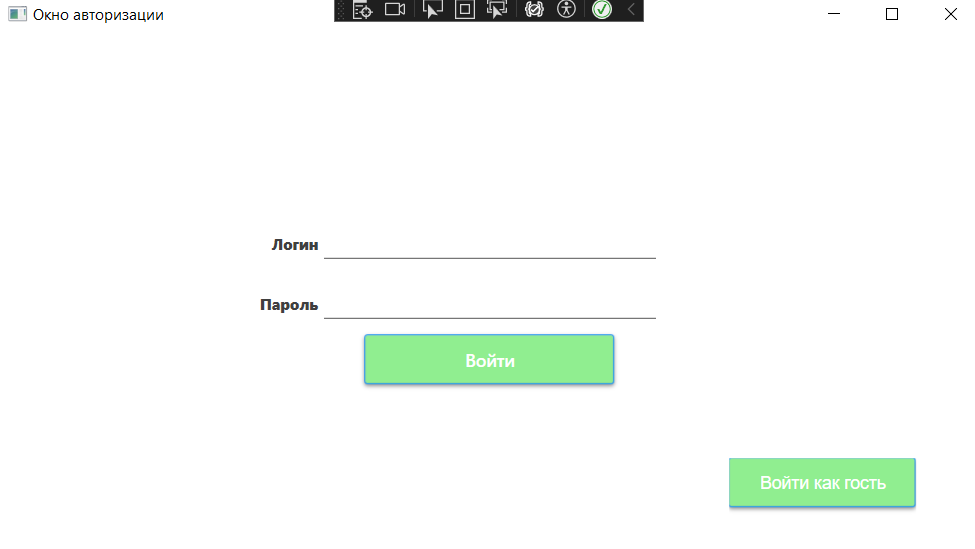


Рис.П5.1 Окно авторизации

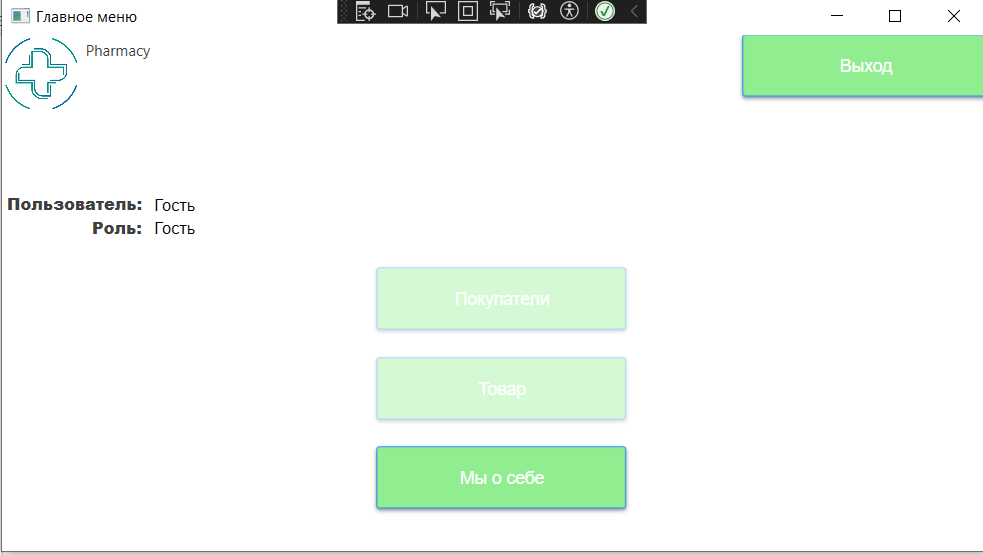


Рис.П5.2 Главное меню

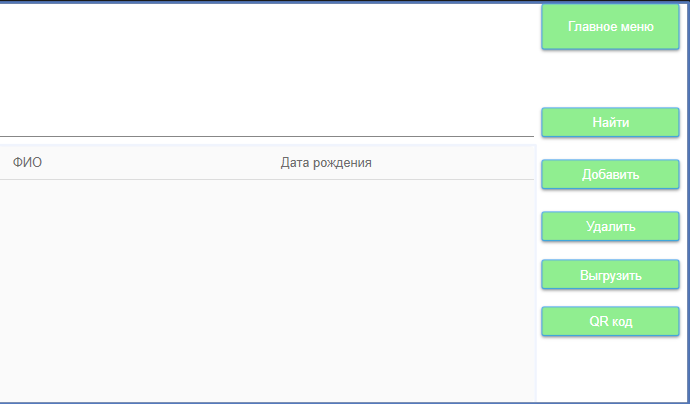


Рис.П5.3 Категория “Клиенты”

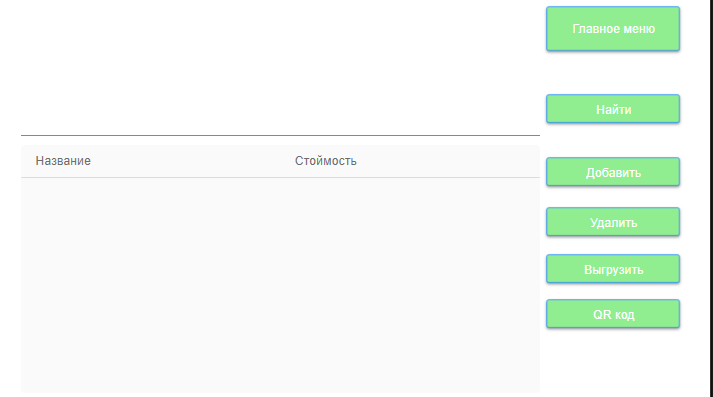


Рис.П5.4 Категория “Услуги”

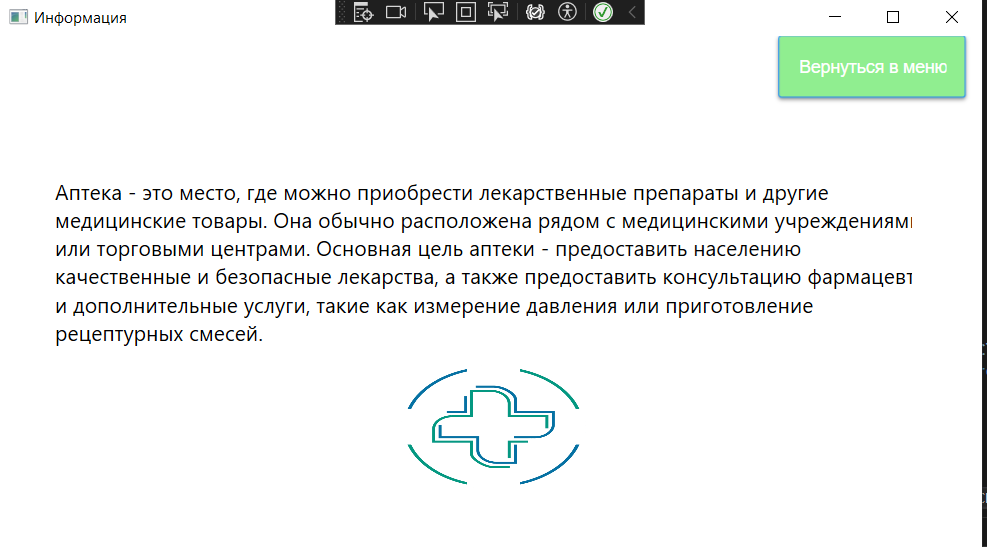


Рис.П5. 5 Категория «Мы о себе»

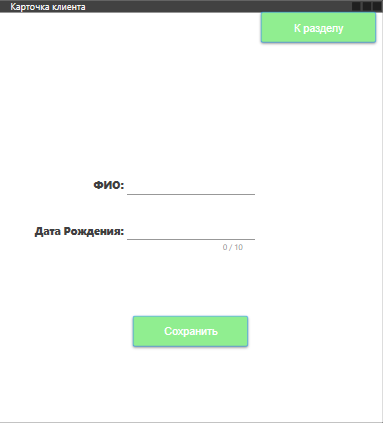
****

Рис.П5. 6 Категория «Карточка клиента»

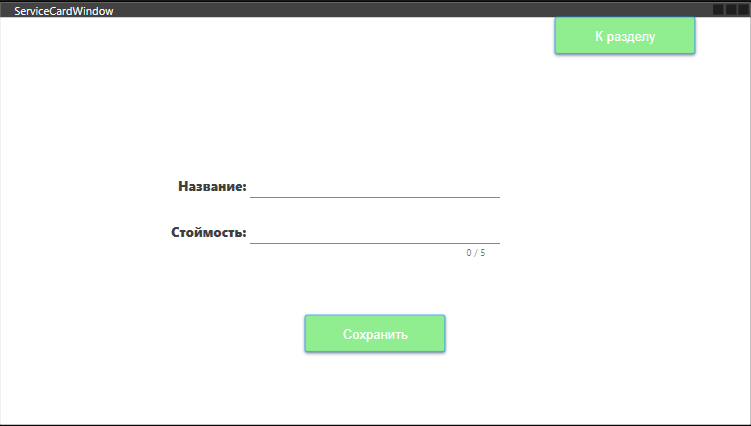
****

Рис.П5. 7 Категория “Выбор товара”

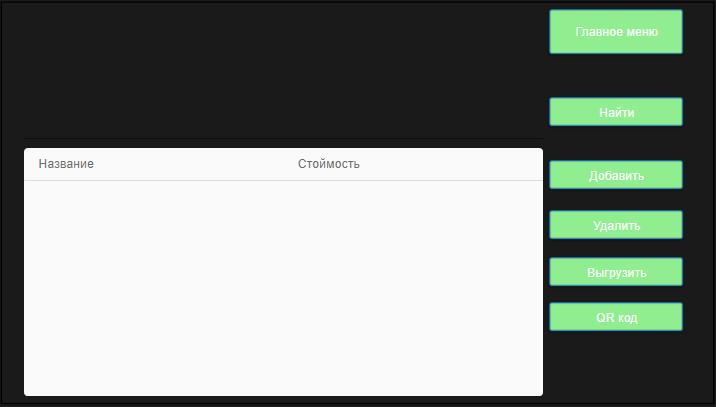
****

Рис.П5. 8 Категория «Товары»

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«КОСТРОМСКОЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 09.02.07 «Информационные системы и программирование»

| **УТВЕРЖДАЮ** |
| --- |
| Руководитель: Э.М. Павлова |
| «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

| **Текст программы** |
| --- |
| ПРИЛОЖЕНИЕ 6 |
| КП-02069846-09.02.07-11-81  Программа на C# |
| Листов 3 |

| Выполнил: | \_\_\_\_\_\_\_\_ И.С. Котов |
| --- | --- |

[global::System.Serializable()]

[global::System.ComponentModel.DesignerCategoryAttribute("code")]

[global::System.ComponentModel.ToolboxItem(true)]

[global::System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedDataSetSchema")]

[global::System.Xml.Serialization.XmlRootAttribute("apteka")]

[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.DataSet")]

public partial class apteka : global::System.Data.DataSet {

private global::System.Data.SchemaSerializationMode \_schemaSerializationMode = global::System.Data.SchemaSerializationMode.IncludeSchema;

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

public apteka() {

this.BeginInit();

this.InitClass();

global::System.ComponentModel.CollectionChangeEventHandler schemaChangedHandler = new global::System.ComponentModel.CollectionChangeEventHandler(this.SchemaChanged);

base.Tables.CollectionChanged += schemaChangedHandler;

base.Relations.CollectionChanged += schemaChangedHandler;

this.EndInit();

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

protected apteka(global::System.Runtime.Serialization.SerializationInfo info, global::System.Runtime.Serialization.StreamingContext context) :

base(info, context, false) {

if ((this.IsBinarySerialized(info, context) == true)) {

this.InitVars(false);

global::System.ComponentModel.CollectionChangeEventHandler schemaChangedHandler1 = new global::System.ComponentModel.CollectionChangeEventHandler(this.SchemaChanged);

this.Tables.CollectionChanged += schemaChangedHandler1;

this.Relations.CollectionChanged += schemaChangedHandler1;

return;

}

string strSchema = ((string)(info.GetValue("XmlSchema", typeof(string))));

if ((this.DetermineSchemaSerializationMode(info, context) == global::System.Data.SchemaSerializationMode.IncludeSchema)) {

global::System.Data.DataSet ds = new global::System.Data.DataSet();

ds.ReadXmlSchema(new global::System.Xml.XmlTextReader(new global::System.IO.StringReader(strSchema)));

this.DataSetName = ds.DataSetName;

this.Prefix = ds.Prefix;

this.Namespace = ds.Namespace;

this.Locale = ds.Locale;

this.CaseSensitive = ds.CaseSensitive;

this.EnforceConstraints = ds.EnforceConstraints;

this.Merge(ds, false, global::System.Data.MissingSchemaAction.Add);

this.InitVars();

}

else {

this.ReadXmlSchema(new global::System.Xml.XmlTextReader(new global::System.IO.StringReader(strSchema)));

}

this.GetSerializationData(info, context);

global::System.ComponentModel.CollectionChangeEventHandler schemaChangedHandler = new global::System.ComponentModel.CollectionChangeEventHandler(this.SchemaChanged);

base.Tables.CollectionChanged += schemaChangedHandler;

this.Relations.CollectionChanged += schemaChangedHandler;

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

[global::System.ComponentModel.BrowsableAttribute(true)]

[global::System.ComponentModel.DesignerSerializationVisibilityAttribute(global::System.ComponentModel.DesignerSerializationVisibility.Visible)]

public override global::System.Data.SchemaSerializationMode SchemaSerializationMode {

get {

return this.\_schemaSerializationMode;

}

set {

this.\_schemaSerializationMode = value;

}

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

[global::System.ComponentModel.DesignerSerializationVisibilityAttribute(global::System.ComponentModel.DesignerSerializationVisibility.Hidden)]

public new global::System.Data.DataTableCollection Tables {

get {

return base.Tables;

}

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

[global::System.ComponentModel.DesignerSerializationVisibilityAttribute(global::System.ComponentModel.DesignerSerializationVisibility.Hidden)]

public new global::System.Data.DataRelationCollection Relations {

get {

return base.Relations;

}

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

protected override void InitializeDerivedDataSet() {

this.BeginInit();

this.InitClass();

this.EndInit();

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

public override global::System.Data.DataSet Clone() {

apteka cln = ((apteka)(base.Clone()));

cln.InitVars();

cln.SchemaSerializationMode = this.SchemaSerializationMode;

return cln;

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

protected override bool ShouldSerializeTables() {

return false;

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

protected override bool ShouldSerializeRelations() {

return false;

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

protected override void ReadXmlSerializable(global::System.Xml.XmlReader reader) {

if ((this.DetermineSchemaSerializationMode(reader) == global::System.Data.SchemaSerializationMode.IncludeSchema)) {

this.Reset();

global::System.Data.DataSet ds = new global::System.Data.DataSet();

ds.ReadXml(reader);

this.DataSetName = ds.DataSetName;

this.Prefix = ds.Prefix;

this.Namespace = ds.Namespace;

this.Locale = ds.Locale;

this.CaseSensitive = ds.CaseSensitive;

this.EnforceConstraints = ds.EnforceConstraints;

this.Merge(ds, false, global::System.Data.MissingSchemaAction.Add);

this.InitVars();

}

else {

this.ReadXml(reader);

this.InitVars();

}

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

protected override global::System.Xml.Schema.XmlSchema GetSchemaSerializable() {

global::System.IO.MemoryStream stream = new global::System.IO.MemoryStream();

this.WriteXmlSchema(new global::System.Xml.XmlTextWriter(stream, null));

stream.Position = 0;

return global::System.Xml.Schema.XmlSchema.Read(new global::System.Xml.XmlTextReader(stream), null);

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

internal void InitVars() {

this.InitVars(true);

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

internal void InitVars(bool initTable) {

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

private void InitClass() {

this.DataSetName = "apteka";

this.Prefix = "";

this.Namespace = "http://tempuri.org/apteka.xsd";

this.EnforceConstraints = true;

this.SchemaSerializationMode = global::System.Data.SchemaSerializationMode.IncludeSchema;

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

private void SchemaChanged(object sender, global::System.ComponentModel.CollectionChangeEventArgs e) {

if ((e.Action == global::System.ComponentModel.CollectionChangeAction.Remove)) {

this.InitVars();

}

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "17.0.0.0")]

public static global::System.Xml.Schema.XmlSchemaComplexType GetTypedDataSetSchema(global::System.Xml.Schema.XmlSchemaSet xs) {

apteka ds = new apteka();

global::System.Xml.Schema.XmlSchemaComplexType type = new global::System.Xml.Schema.XmlSchemaComplexType();

global::System.Xml.Schema.XmlSchemaSequence sequence = new global::System.Xml.Schema.XmlSchemaSequence();

global::System.Xml.Schema.XmlSchemaAny any = new global::System.Xml.Schema.XmlSchemaAny();

any.Namespace = ds.Namespace;

sequence.Items.Add(any);

type.Particle = sequence;

global::System.Xml.Schema.XmlSchema dsSchema = ds.GetSchemaSerializable();

if (xs.Contains(dsSchema.TargetNamespace)) {

global::System.IO.MemoryStream s1 = new global::System.IO.MemoryStream();

global::System.IO.MemoryStream s2 = new global::System.IO.MemoryStream();

try {

global::System.Xml.Schema.XmlSchema schema = null;

dsSchema.Write(s1);

for (global::System.Collections.IEnumerator schemas = xs.Schemas(dsSchema.TargetNamespace).GetEnumerator(); schemas.MoveNext(); ) {

schema = ((global::System.Xml.Schema.XmlSchema)(schemas.Current));

s2.SetLength(0);

schema.Write(s2);

if ((s1.Length == s2.Length)) {

s1.Position = 0;

s2.Position = 0;

for (; ((s1.Position != s1.Length)

&& (s1.ReadByte() == s2.ReadByte())); ) {

;

}

if ((s1.Position == s1.Length)) {

return type;

}

}

}

}

finally {

if ((s1 != null)) {

s1.Close();

}

if ((s2 != null)) {

s2.Close();

}

}

}

xs.Add(dsSchema);

return type;

}

}

}

#pragma warning restore 1591