**СОДЕРЖАНИЕ**

[Введение 4](#_Toc72935595)

[1 Постановка задачи 5](#_Toc72935596)

[1.1 Сущность задачи 5](#_Toc72935597)

[1.2 Входная оперативная информация 6](#_Toc72935598)

[1.3 Постоянная информация задачи 7](#_Toc72935599)

[1.4 Выходная информация 7](#_Toc72935600)

[2 Объектно-ориентированный анализ задачи 8](#_Toc72935601)

[2.1 Описание задачи в ключевых абстракциях предметной области 8](#_Toc72935602)

[2.2 Описание интерфейса классов 8](#_Toc72935603)

[3 Описание программы 10](#_Toc72935604)

[3.1 Общие сведения 10](#_Toc72935605)

[3.2 Функциональное назначение 10](#_Toc72935606)

[3.3 Описание логической структуры 10](#_Toc72935607)

[3.4 Используемые технические средства 12](#_Toc72935608)

[3.5 Вызов и загрузка 13](#_Toc72935609)

[4 Программа и методика испытаний 15](#_Toc72935610)

[4.1 Цель испытаний 15](#_Toc72935611)

[4.2 Описание используемых методов испытаний и их результаты 16](#_Toc72935612)

[5 Описание применения 18](#_Toc72935613)

[5.1 Назначение программы 18](#_Toc72935614)

[5.2 Условия выполнения программы 18](#_Toc72935615)

[5.3 Выполнение программы 18](#_Toc72935616)

[5.3.1 Запуск программы 18](#_Toc72935617)

[5.3.2 Выполнение программы 19](#_Toc72935618)

[5.3.3 Завершение работы программы 23](#_Toc72935619)

[Заключение 24](#_Toc72935620)

[Список используемых источников 25](#_Toc72935621)

[Приложение А 26](#_Toc72935622)

# Введение

Сегодня в начале 21 века, компьютеры можно встретить почти во всех сферах деятельности. Они достаточно прочно укрепились в нашей жизни и сознании. В настоящее время происходит стремительное развитие вычислительной техники и программного обеспечения ЭВМ.

Все большему числу специалистов в самых различных областях требуется доступ к вычислительной технике для решения стоящих перед ними задач. Сейчас компьютеры превратились в универсальные средства для обработки всех видов информации, используемых человеком. С помощью компьютеров не только проводятся числовые расчеты, но и подготавливаются к печати книги, создаются рисунки, кинофильмы, музыка, осуществляется управление заводами и космическими кораблями, проводятся всевозможные тестирования, проверяющие знания людей в той или иной области и так далее.

Одной из значительных ниш, которую заняли ЭВМ, является автоматизация документооборота и хранение больших объемов данных. Информационные технологии и развитие возможностей самих компьютеров позволяет хранить огромное количество данных, обеспечивая при этом высокую (в некоторых случаях почти мгновенную) скорость доступа к информации, возможность корректировки и модификации, переносимости и множимости данных, а если требуется, то их защиту. Также работа с базой данных на ЭВМ обычно более проста и удобна, чем работами с бумажными и другими носителями информации [1].

Все вышеперечисленные возможности реализуются с помощью систем управления базами данных. Сфера применения СУБД практически безгранична. трудно представить себе область интересов человека, которая не требовала бы хранения информации для дальнейшего использования.

Целью курсовой работы является написания приложения электронного сборника книг по программированию.

# 1 Постановка задачи

Создание любой программы начинается с постановки задачи. Изначально задача формулируется в терминах предметной области, и необходимо перевести ее на язык понятий, более близких к программированию.

Поскольку программист редко досконально разбирается в предметной области, а заказчик – в программировании (простой пример: требуется написать бухгалтерскую программу), постановка задачи может стать весьма непростым итерационным процессом [2].

Кроме того, при постановке задачи заказчик зачастую не может четко и полно сформулировать свои требования и критерии.

## 1.1 Сущность задачи

Необходимо разработать приложение для хранение информации о книгах программирования, учитывая их разбиение по авторам, издателям и языкам программирования. Также стоит запомнить, что у одной книги может быть несколько авторов и в ней могут быть описаны работа с несколькими языками программирования.

Основные функции представлены в виде UML диаграммы вариантов использования.

При разработке диаграммы вариантов использования проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью вариантов использования. Актером называется любая сущность, взаимодействующая с системой извне . Вариант использования служит для описания сервисов, которые система предоставляет актеру. Диаграмма вариантов использования может дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов [3].

Диаграмма вариантов использования представлена на рисунке 1.1.

В разрабатываемой системе возможна работа одного типа пользователя, соответственно на диаграмме вариантов использования представлен один актер, который обладает всеми функциями приложения, что позволяет ему взаимодействовать с данными всех таблиц приложения.

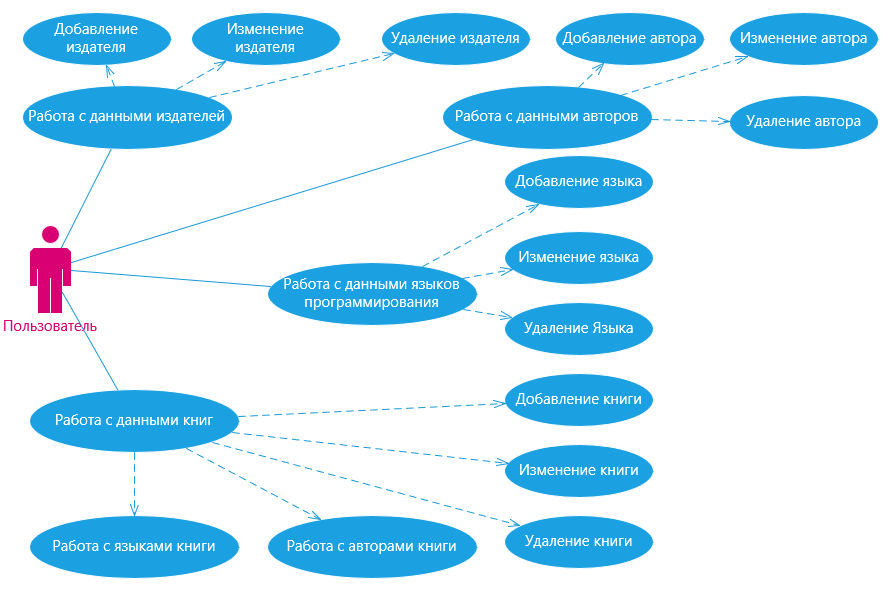


Рисунок 1.1 – Диаграмма вариантов использования

Для полноценной работы системы компьютер пользователя должен удовлетворять следующим аппаратным требованиям:

* процессор AMD Athlon X4 840 3.10 GHz;
* ОЗУ: не менее 4Гб;
* HDD: не менее 250Гб;
* видеоадаптер: DirectX9;
* видеопамять: не менее 512 МБ;

Кроме того, необходимы следующие программные компоненты:

* Windows 7 и выше;
* Microsoft Access для хранения базы данных и драйвера к нему;
* MS .NET Framework 4.6.1.

## 1.2 Входная оперативная информация

Входными параметрами для работы приложения будут информация о книгах, авторах, издательствах и языках программирования.

Об издательствах будет получена информация о названии издательства.

Об языках программирования будет получена информация о названии языка программирования

Об авторах будет получены ФИО авторов

О книгах будет получена следующая информация: название книги, кто публиковал в каком году была публикация, какие авторы у этой книги, о каких языках программирования в ней идет речь и описание данной книги.

## 1.3 Постоянная информация задачи

Постоянной информацией для разрабатываемой информационной системы являются справочники. Постоянная информация задачи храниться в базе данных Database.access.

## 1.4 Выходная информация

Выходной информацией для задачи являются табличные данные о книгах, издательствах, авторах и языках программирования.

# 2 Объектно-ориентированный анализ задачи

## 2.1 Описание задачи в ключевых абстракциях предметной области

При решении задачи воспользуемся многими плюсами структурного программирования и для удобства разработки включим классы. Так как проект невелик по объёму и не получит дальнейшего развития не имеет смысла делать программу полностью объектной: с помощью подпрограмм структурного программирования можно решить данную задачу быстрее и проще.

Проектирование классов производилось с учетом потребностей данной программы в объектах. При анализе задачи в ключевых абстракциях выделяем: пользователь, данные, база для хранения данных. При этом пользователь взаимодействует с базой, обмениваясь с ней данными. Значит нам нужен класс данных. Благодаря развитию языка C#, управление базой данных производится посредством стандартных объектов, нужно лишь настроить подключение к ней.

Объекты этого класса будут хранится в ОЗУ компьютера при работе программы (что позволяет организовать к ним быстрый доступ), или на диске при работе других приложений или выключенном ПК. Особенности программы удобнее реализовать с помощью структурного программирования, поэтому иерархии классов в программе нет.

## 2.2 Описание интерфейса классов

Поскольку мы уже имеем знание о том, какие сущности будем хранить и использовать, можем описать какими классами будем их обрабатывать.

Приложение хорошо делить на модули, в зависимости от объектов, с которыми они работают. У нас четы отдельных объекта, с которыми можно ввести работу, а именно: книги, авторы, издательства, языки программирования.

Для каждого из этих объектов создадим форму работы именно с ними. Работа с объектами языки книг и авторы книг будет проводиться при работе с объектом книга, т.к. они зависят напрямую от выбранного объекта.

На рисунке 2.2 отображены классы разрабатываемого приложения.

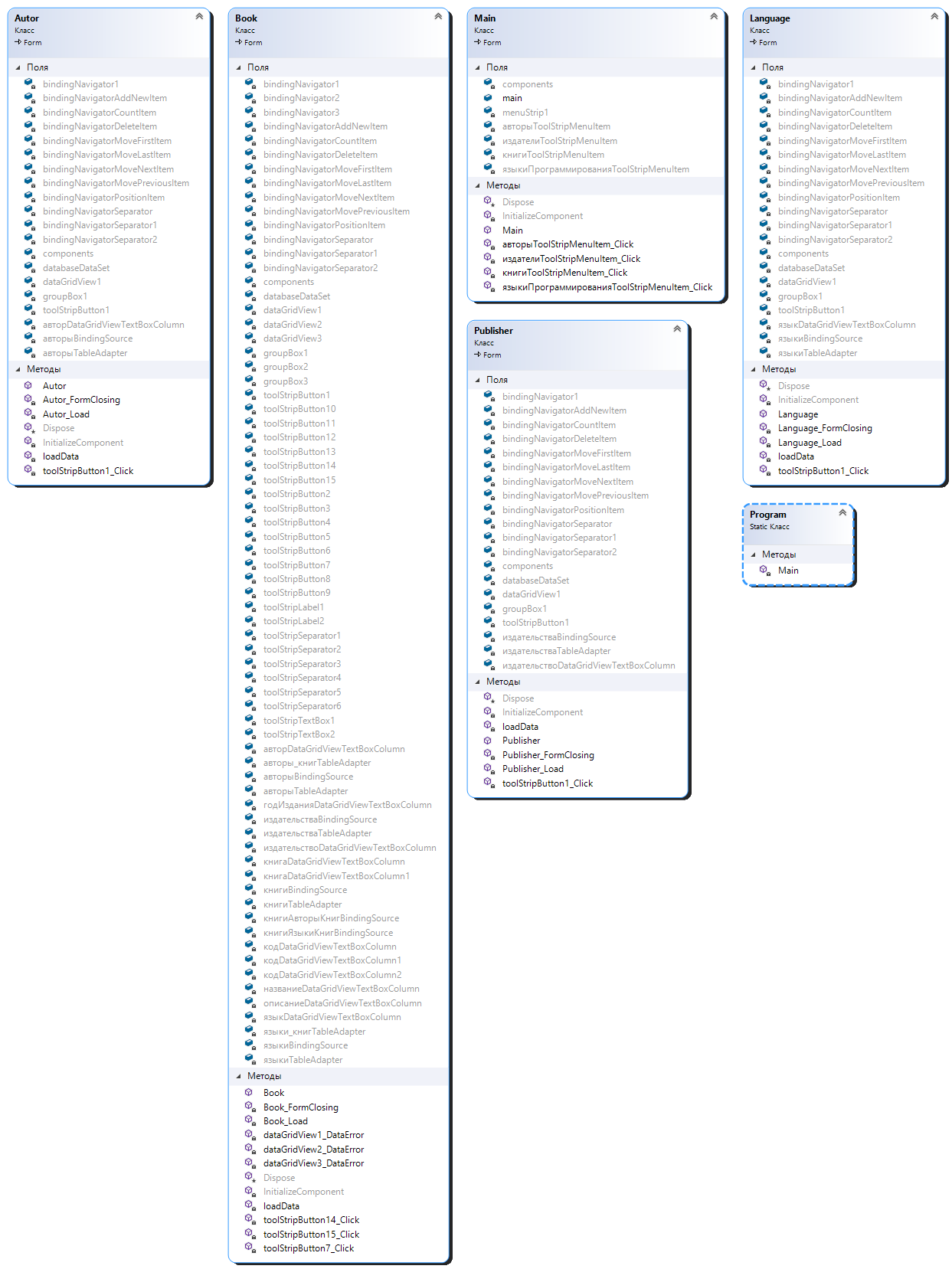


Рисунок 2.1 – Диаграмма классов

# 3 Описание программы

## 3.1 Общие сведения

Программное средство создается с целью облегчить взаимодействие с базой данных по хранению и книг по программированию, автоматизировав процесс добавления, изменения и удаления информации.

Книги по программированию имеют множество повторяющихся атрибутов, таких издатель, автор и язык программирования. С целью уменьшения повторения данных, все эти данные хранятся в разных таблицах и имеют взаимоотношения один ко многим или многие ко многим (используя дополнительную сводную таблицу).

## 3.2 Функциональное назначение

Функциональное назначение разрабатываемого курсового проекта заключается в создании приложения электронного сборника книг по программированию.

Функциональное назначение разрабатываемой системы состоит в следующем:

* хранении и организации информации;
* модификации информации: добавлении, редактировании, удалении.

## 3.3 Описание логической структуры

Для реализаций действий над объектами, нужно понять, с каким объектами будет работать наше приложение. Затем, используя эти объекты создать базу данных для приложения. Анализ объектов для базы данных лучше всего начинать с построения логической модели.

Логическая модель данных – описание объектов предметной области, их атрибутов и взаимосвязей между ними в том объеме, в котором они подлежат непосредственному хранению в базе данных системы [4].

Для разрабатываемой информационной системы было выделено 6 сущностей.

Для хранения информации об языках программирования была выделена сущность «Языки» со следующими атрибутами:

* Язык

Для хранения информации об авторах книг была выделена сущность «Авторы» со следующими атрибутами:

* Автор

Для хранения информации об издательствах была выделена сущность «Издательства» со следующими атрибутами:

* Издательство

Для хранения информации о книгах была выделена сущность «Книги» со следующими атрибутами:

* Код
* Название
* Год издания
* Издательство
* Описание

Для хранения информации об языках конкретной книги была выделена сущность «Языки книг» со следующими атрибутами:

* Код
* Книга
* Язык

Для хранения информации об авторах конкретной книги была выделена сущность «Авторы книг» со следующими атрибутами:

* Код
* Книга
* Автор

На основании выявленных сущностей и связей между ними была разработана и спроектирована база данных.

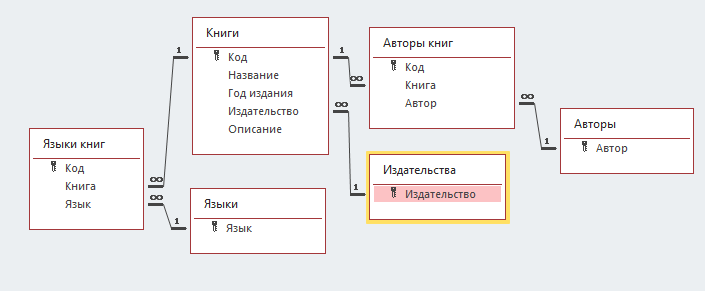


Рисунок 3.1 – Диаграмма базы данных

## 3.4 Используемые технические средства

Для разработки клиентского приложения была выбрана среда MicrosoftVisualStudio 2019. VisualStudio – это средство для разработки программного обеспечения, которое имеет большой функционал и возможности для отладки и развертывания приложений на различных платформах [5]. В комплект разработчика входят интегрированная среда разработки (IDE) и инструменты тестирования для всех этапов процесса разработки программного обеспечения. Интегрированная среда разработки представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

MicrosoftVisualStudio поддерживает широкий спектр популярных языков программирования. Это хороший инструмент для профессиональных разработчиков программного обеспечения, которые хотят создавать коммерческие программы.

Основные преимущества MicrosoftVisualStudio:

- поддержка множества языков при разработке.

- редактор VisualStudio оптимизирует работу программиста по кодированию. В ходе того, как вводится новый код, на экране будут появляться подсказки. Для ускорения выполнения многих задач в VisualStudio предоставлено большое количество клавиатурных комбинаций. Существует набор средств быстрой переработки, которые позволяют быстро усовершенствовать структуру кода, не отрываясь от процесса программирования.

- интуитивный стиль кодирования. По умолчанию VisualStudio форматирует код по мере его ввода, автоматически вставляя необходимые отступы и применяя цветовое кодирование для выделения элементов типа комментариев. Такие незначительные отличия делают код более удобным для чтения и менее подверженным ошибкам.

- более высокая скорость разработки. Многие из функциональных возможностей VisualStudio направлены на то, чтобы помогать разработчику делать свою работу как можно быстрее. Удобные функции, вроде функции IntelliSense, функции поиска и замены и функции автоматического добавления и удаления комментариев позволяют разработчику работать быстро и эффективно.

- возможности отладки. Предлагаемые в VisualStudio инструменты отладки являются наилучшим средством для отслеживания ошибок и диагностирования странного поведения. Разработчик может выполнять свой код по строке за раз, устанавливать интеллектуальные точки прерывания, при желании сохраняя их для использования в будущем, и в любое время просматривать текущую информацию из памяти.

Для разработки приложения был выбран язык C#. C# предназначен для разработки приложений, которые выполняются в среде .NET Framework. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую языкам C. C# — это объектно-ориентированный язык, который поддерживает все правила объектно-ориентированного программирования: инкапсуляцию, наследование и полиморфизм. Язык поддерживает перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML [6].

C# имеет множество функций, обеспечивающих надежность и устойчивость приложений. Сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами. Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и их восстановлению. Типобезопасная структура языка делает невозможным чтение из неинициализированных переменных, индексацию массивов за пределами их границ или выполнение непроверенных приведений типов.

## 3.5 Вызов и загрузка

Для запуска приложения необходимо 2 раза нажать на приложение «Electronic collection of books on programming.exe» или один раз для выделения, а затем нажать клавишу «Enter», после чего будет запущено главное окно приложения.

В зависимости от дельнейших действий, доступны те или иные варианты взаимодействий. Алгоритм работы приложения представлен на рисунке 3.2.

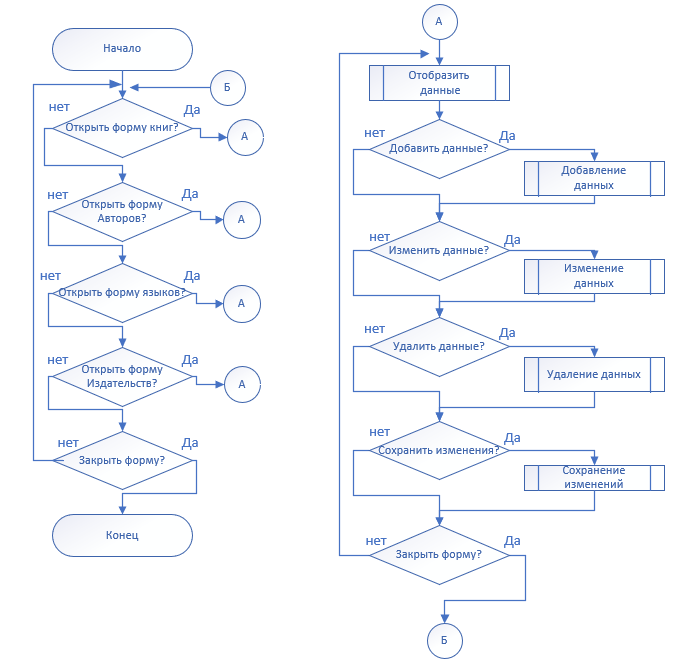


Рисунок 3.2 – Алгоритм программы

# 4 Программа и методика испытаний

## 4.1 Цель испытаний

Системное тестирование - это тестирование программного обеспечения, выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям [7]. Системное тестирование относится к методам тестирования чёрного ящика, и, тем самым, не требует знаний о внутреннем устройстве системы. Тестирование бывает:

* функциональное тестирование;
* тестирование пользовательского интерфейса;
* тестирование совместимости;
* тестирование безопасности;
* тестирование производительности;
* автоматическое тестирование;
* интеграционное тестирование.

Функциональное тестирование - это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определённых условиях решать задачи, нужные пользователям [8]. Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает.

Функциональные требования включают в себя:

* функциональная пригодность;
* точность;
* способность к взаимодействию;
* соответствие стандартам и правилам;
* защищённость.

Тестирование и отладка программы являются наиважнейшими этапами разработки любых программных продуктов. Цель этого этапа – проверка правильности и точности реализации функций, выполнение которых возлагается на данный программный продукт. В случае выявления некоторых неточностей и ошибок необходимо проведение работ по их исправлению и доработке программного продукта до требуемого уровня.

## 4.2 Описание используемых методов испытаний и их результаты

На основе функций, которые должны быть протестированы в разработанной информационной системе, составлен чек-лист (checklist), приведенный в таблице 4.1.

Таблица 4.1 – Чек-лист

|  |  |  |  |
| --- | --- | --- | --- |
| Тестируемый модуль | | Тестируемая функция | Результат |
| 1 | | 2 | 3 |
| Запуск приложения | | Функция запуска | Выполнено успешно |
| Выход из приложения | | Функция закрытия программы |
| Ввод данных | | Определение соответствие типа вводимых данных | Выполнено успешно |
| Ввод ошибочных данных | Вывод сообщении об ошибке | | Выполнено успешно |
| Сохранение и загрузка | Сохранение данных в базу данных | | Выполнена успешно |
| Чтение данных из базы данных | |

В качестве тестирования программного продукта был выбран тест-кейс (**Test Case)**.

**Test Case** - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Тестирование производилось на ОС «Windws 10.

Таблица 4.2 – Тест-кейсы

|  |  |  |
| --- | --- | --- |
| **Краткое описание** | **Шаги для воспроизведения** | **Ожидаемый результат** |
| Добавление новой записи в базу данных | Заполнены все столбцы строки, нажата кнопка Enter, нажата кнопка сохранить, сохранение подтверждено | Уведомление об успешном сохранении. Новые данные отображаются в общем списке |
| Редактирование конкретного кейсв | Выбрана строка таблицы, изменены требуемые данные, нажата клавиша Enter, нажата клавиша сохранить и сохранение подтверждено» | Уведомление об успешном сохранении. Обновленные данные отображаются в таблице. |
| Удаление рейса | Нажата кнопка «Удалить», нажата клавиша сохранить и сохранение подтверждено» | Уведомление об успешном сохранении. Удаленные данные не отображаются в таблице. |
| Сортировка | Нажатие на название любого столбца | Данные отсортированы по столбцу |

|  |  |  |
| --- | --- | --- |
| Добавление новой записи в базу данных | Заполнены все столбцы строки, нажата кнопка Enter, нажата кнопка сохранить, сохранение подтверждено | Уведомление об успешном сохранении. Новые данные отображаются в общем списке |
| Редактирование конкретного кейсв | Выбрана строка таблицы, изменены требуемые данные, нажата клавиша Enter, нажата клавиша сохранить и сохранение подтверждено» | Уведомление об успешном сохранении. Обновленные данные отображаются в таблице. |
| Удаление рейса | Нажата кнопка «Удалить», нажата клавиша сохранить и сохранение подтверждено» | Уведомление об успешном сохранении. Удаленные данные не отображаются в таблице. |
| Сортировка | Нажатие на название любого столбца | Данные отсортированы по столбцу |

# 5 Описание применения

## 5.1 Назначение программы

Ожидается, что внедрение программы ускорит обработку и улучшит хранение информации о книгах по программированию за счет автоматизации таких операций как добавление, изменение и удаления данных, а также за счет поддержания непротиворечивости за счет связи таблиц между собой связями один ко многим.

Разрабатываемый продукт предназначен для использования любым желающим собрать или работать с электронным сборником книг по программированию.

## 5.2 Условия выполнения программы

Для полноценной работы системы компьютер пользователя должен удовлетворять следующим аппаратным требованиям:

* процессор AMD Athlon X4 840 3.10 GHz;
* ОЗУ: не менее 1Гб;
* HDD: не менее 250Гб;
* видеоадаптер: DirectX9;
* видеопамять: не менее 32 МБ;

Для запуска и бесперебойной работы приложения, на компьютере пользователя необходимы следующие программные компоненты:

* Windows 7 и выше;
* MS SQL 8 для базы данных;
* MS .NET Framework 4.6;
* Microsoft Access 2016 и драйвера к нему

## 5.3 Выполнение программы

### 5.3.1 Запуск программы

Для запуска приложения необходимо 2 раза нажать на приложение «Electronic collection of books on programming.exe» или один раз для выделения, а затем нажать клавишу «Enter», после чего будет запущено главное окно приложения.

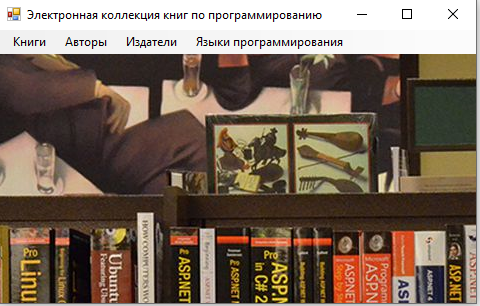


Рисунок 5.1 – Главное окно приложения

### 5.3.2 Выполнение программы

После запуска приложения, пользователь может перейти в одну из нескольких других форм или запершить работу приложения. Для работы с языками программирования следует нажать на соответствующее меню. Откроется форма для работы с языками

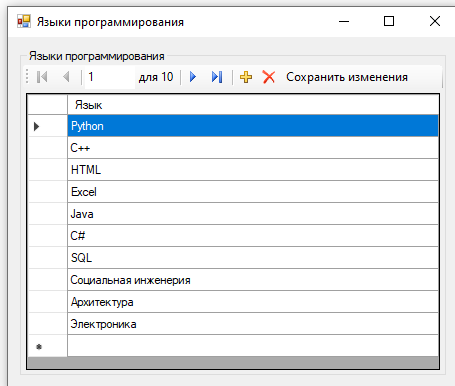


Рисунок 5.2 – Форма работы с языками

Все формы поддерживают изменение размета, т.е. если пользователю удобно сделать форму на весь экран, растянуть по горизонтали или вертикали, то содержимое форм также будет растягиваться.

Для добавления нового языка стоит внести значение в последнюю пустую строку и нажать кнопку сохранения. При этом у пользователя запросят разрешение на изменения данных в таблице. Изменением считается любое действие со значениями ячеек, будь это добавление, правка столбца или удаление строки.

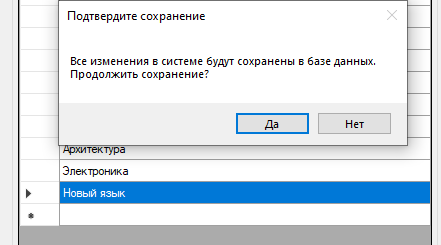


Рисунок 5.3 – Предупреждение об изменениях

После выдачи разрешения, наша запись сохранится.

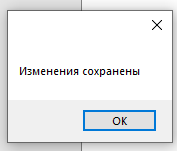


Рисунок 5.4 – Уведомление об успешном сохранении

Для сортировки данных в любой из таблиц достаточно нажать на заголовок столбца, по которому нужна сортировка

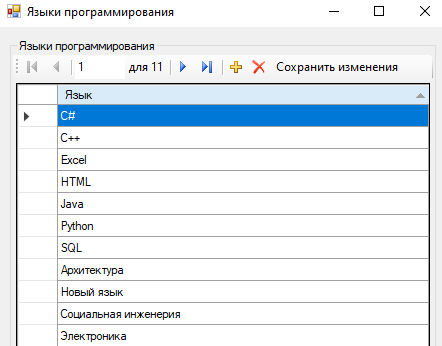


Рисунок 5.5 – Сортировка по языкам

Для выхода из формы нужно нажать крестик справа вверху экрана, произойдет переход на главную форму.

Формы работы с издателями и авторами устроена таким же образом

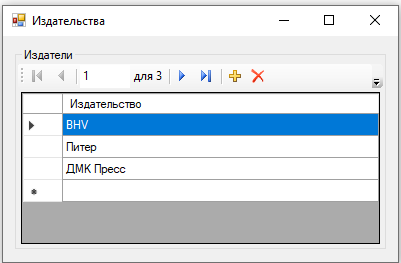


Рисунок 5.6 – Форма работы с издателями

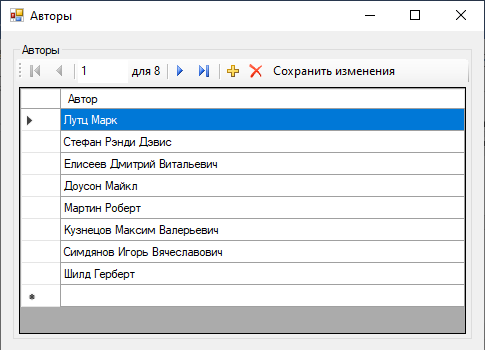


Рисунок 5.7 – Форма работы с авторами

При переходе на форму книг, при выборе какой-либо книги из списка, в таблицах ниже будут отображены все авторы этой книги и все языки, которые в ней упоминаются

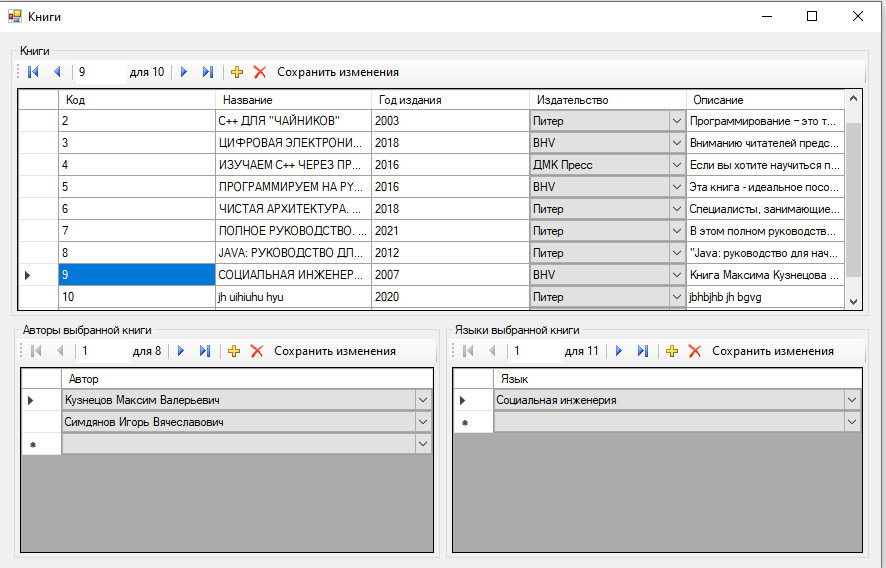


Рисунок 5.8 – Форма работы с книгами

Для удобства работы с таблицами, все заполненные ранее данные (авторы, языки, издательства) выбираются при помощи выпадающего списка

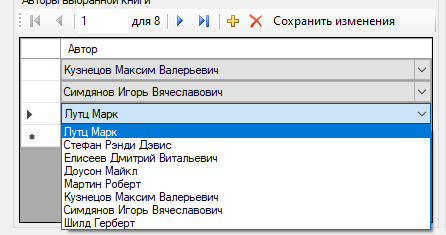


Рисунок 5.9 – Выбор из выпадающего списка

Книги могут иметь огромное описание, которое не вмещается в одну строку таблицы, поэтому, при изменении размера столбца или сортировке данных, добавлен перенос строк. Благодаря этому можно просматривать полное описание книги в удобном виде

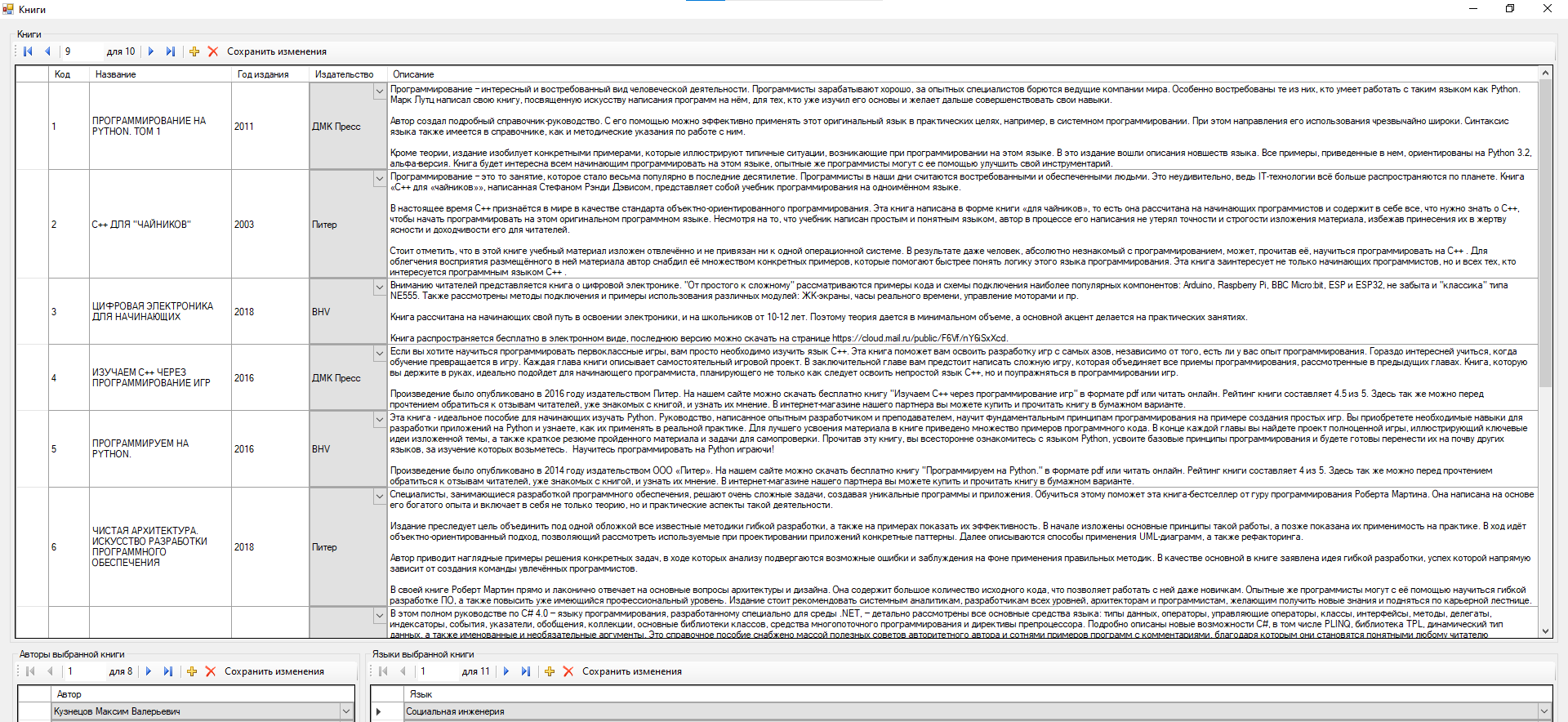


Рисунок 5.10 – Удобный просмотр описания книг

### 5.3.3 Завершение работы программы

Для завершения работы достаточно выйти в главное меню приложение и нажать на крестик справа в углу формы. Приложение завершит свою работу.

# Заключение

Главным итогом курсового проектирования является разработанное приложение, предназначенное для работы с базой данных, хранящей сведения о коллекции книг по программированию

В рамках данного курсового проекта была достигнута главная цель, которая заключалась в реализации базы данных и программного приложения для учета коллекции книг по программированию. Для ее достижения использовались стандартные компоненты среды Visual Studio и язык программирования высокого уровня C#.

Разработанное программное средство представляет собой систему, предназначенную для облегчения работы c базой данных по ведению коллекции книг по программированию.

Данная программа полностью соответствует требованиям технического задания. Программный продукт может эффективно функционировать на любом персональном компьютере, удовлетворяющем техническим требованиям.

# Список используемых источников

1. Информационные технологии [Электронный ресурс]. – Режим доступа: https://moodle.kstu.ru/pluginfile.php/301635/mod\_resource/content/2/Лекция%20№1.pdf. – Дата доступа: 26.05.2021.
2. Постановка задачи [Электронный ресурс]. – Режим доступа: https://3ys.ru/yazyki-i-sistemy-programmirovaniya/postanovka-zadachi.html. – Дата доступа: 26.05.2021.
3. Элементы графической нотации диаграммы вариантов использования [Электронный ресурс]. – Режим доступа: https://intuit.ru/studies/courses/32/32/lecture/1004. – Дата доступа: 26.05.2021.
4. Логическая модель данных [Электронный ресурс]. – Режим доступа: https://studbooks.net/2184322/informatika/logicheskaya\_model\_dannyh. – Дата доступа: 26.05.2021.
5. Новые возможности Visual Studio 2019 [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2019. – Дата доступа: 26.05.2021.
6. Объектно-ориентированное программирование на языке C# Источник: https://referatbooks.ru/kursovaya-rabota/obyektno-orientirovannoe-programmirovanie-na-yazyike-c/ [Электронный ресурс]. – Режим доступа: https://referatbooks.ru/kursovaya-rabota/obyektno-orientirovannoe-programmirovanie-na-yazyike-c/. – Дата доступа: 26.05.2021.
7. Системное тестирование программного обеспечения [Электронный ресурс]. – Режим доступа: https://unetway.com/tutorial/sistemnoe-testirovanie-programmnogo-obespecenia. – Дата доступа: 26.05.2021.
8. Функциональное тестирование [Электронный ресурс]. – Режим доступа: https://www.appline.ru/services/testing/functionalnoe-testirovanie. – Дата доступа: 26.05.2021.

# Приложение А

(справочное)

**Исходный текст программы**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Electronic\_collection\_of\_books\_on\_programming

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Main());

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Electronic\_collection\_of\_books\_on\_programming

{

public partial class Main : Form

{

public static Main main;

public Main()

{

InitializeComponent();

main = this;

}

private void книгиToolStripMenuItem\_Click(object sender, EventArgs e)

{

new Book().Show();

this.Hide();

}

private void авторыToolStripMenuItem\_Click(object sender, EventArgs e)

{

new Autor().Show();

this.Hide();

}

private void издателиToolStripMenuItem\_Click(object sender, EventArgs e)

{

new Publisher().Show();

this.Hide();

}

private void языкиПрограммированияToolStripMenuItem\_Click(object sender, EventArgs e)

{

new Language().Show();

this.Hide();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Electronic\_collection\_of\_books\_on\_programming

{

public partial class Book : Form

{

public Book()

{

InitializeComponent();

}

private void Book\_FormClosing(object sender, FormClosingEventArgs e)

{

Main.main.Show();

}

private void Book\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "databaseDataSet.Издательства". При необходимости она может быть перемещена или удалена.

this.издательстваTableAdapter.Fill(this.databaseDataSet.Издательства);

loadData();

}

private void loadData()

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "databaseDataSet.Авторы". При необходимости она может быть перемещена или удалена.

this.авторыTableAdapter.Fill(this.databaseDataSet.Авторы);

// TODO: данная строка кода позволяет загрузить данные в таблицу "databaseDataSet.Языки". При необходимости она может быть перемещена или удалена.

this.языкиTableAdapter.Fill(this.databaseDataSet.Языки);

// TODO: данная строка кода позволяет загрузить данные в таблицу "databaseDataSet.Языки\_книг". При необходимости она может быть перемещена или удалена.

this.языки\_книгTableAdapter.Fill(this.databaseDataSet.Языки\_книг);

// TODO: данная строка кода позволяет загрузить данные в таблицу "databaseDataSet.Авторы\_книг". При необходимости она может быть перемещена или удалена.

this.авторы\_книгTableAdapter.Fill(this.databaseDataSet.Авторы\_книг);

// TODO: данная строка кода позволяет загрузить данные в таблицу "databaseDataSet.Книги". При необходимости она может быть перемещена или удалена.

this.книгиTableAdapter.Fill(this.databaseDataSet.Книги);

dataGridView1.RowTemplate.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

}

private void dataGridView3\_DataError(object sender, DataGridViewDataErrorEventArgs e){ }

private void dataGridView2\_DataError(object sender, DataGridViewDataErrorEventArgs e){ }

private void toolStripButton7\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Все изменения в системе будут сохранены в базе данных.\r\n" +

"Продолжить сохранение?",

"Подтвердите сохранение",

MessageBoxButtons.YesNo) == DialogResult.Yes)

try

{

авторы\_книгTableAdapter.Update(databaseDataSet.Авторы\_книг);

MessageBox.Show("Изменения сохранены");

loadData();

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

private void toolStripButton14\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Все изменения в системе будут сохранены в базе данных.\r\n" +

"Продолжить сохранение?",

"Подтвердите сохранение",

MessageBoxButtons.YesNo) == DialogResult.Yes)

try

{

языки\_книгTableAdapter.Update(databaseDataSet.Языки\_книг);

MessageBox.Show("Изменения сохранены");

loadData();

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

private void toolStripButton15\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Все изменения в системе будут сохранены в базе данных.\r\n" +

"Продолжить сохранение?",

"Подтвердите сохранение",

MessageBoxButtons.YesNo) == DialogResult.Yes)

try

{

книгиTableAdapter.Update(databaseDataSet.Книги);

MessageBox.Show("Изменения сохранены");

loadData();

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

private void dataGridView1\_DataError(object sender, DataGridViewDataErrorEventArgs e)

{

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Electronic\_collection\_of\_books\_on\_programming

{

public partial class Autor : Form

{

public Autor()

{

InitializeComponent();

}

private void Autor\_FormClosing(object sender, FormClosingEventArgs e)

{

Main.main.Show();

}

private void Autor\_Load(object sender, EventArgs e)

{

loadData();

}

private void loadData()

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "databaseDataSet.Авторы". При необходимости она может быть перемещена или удалена.

this.авторыTableAdapter.Fill(this.databaseDataSet.Авторы);

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Все изменения в системе будут сохранены в базе данных.\r\n" +

"Продолжить сохранение?",

"Подтвердите сохранение",

MessageBoxButtons.YesNo) == DialogResult.Yes)

try

{

авторыTableAdapter.Update(databaseDataSet.Авторы);

MessageBox.Show("Изменения сохранены");

loadData();

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Electronic\_collection\_of\_books\_on\_programming

{

public partial class Language : Form

{

public Language()

{

InitializeComponent();

}

private void Language\_FormClosing(object sender, FormClosingEventArgs e)

{

Main.main.Show();

}

private void Language\_Load(object sender, EventArgs e)

{

loadData();

}

private void loadData()

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "databaseDataSet.Языки". При необходимости она может быть перемещена или удалена.

this.языкиTableAdapter.Fill(this.databaseDataSet.Языки);

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Все изменения в системе будут сохранены в базе данных.\r\n" +

"Продолжить сохранение?",

"Подтвердите сохранение",

MessageBoxButtons.YesNo) == DialogResult.Yes)

try

{

языкиTableAdapter.Update(databaseDataSet.Языки);

MessageBox.Show("Изменения сохранены");

loadData();

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Electronic\_collection\_of\_books\_on\_programming

{

public partial class Publisher : Form

{

public Publisher()

{

InitializeComponent();

}

private void Publisher\_FormClosing(object sender, FormClosingEventArgs e)

{

Main.main.Show();

}

private void Publisher\_Load(object sender, EventArgs e)

{

loadData();

}

private void loadData()

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "databaseDataSet.Издательства". При необходимости она может быть перемещена или удалена.

this.издательстваTableAdapter.Fill(this.databaseDataSet.Издательства);

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Все изменения в системе будут сохранены в базе данных.\r\n" +

"Продолжить сохранение?",

"Подтвердите сохранение",

MessageBoxButtons.YesNo) == DialogResult.Yes)

try

{

издательстваTableAdapter.Update(databaseDataSet.Издательства);

MessageBox.Show("Изменения сохранены");

loadData();

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

}

}