Реферат

Курсовая работа 24 страницы, 12 рисунков, 1 приложение, 4 источника.

***Ключевые слова***: автоматизированная система, программное обеспечение, база данных, игры.

***Объект исследования***: Магазин игр.

***Предмет исследования*:** Создание приложения на языке программирования C# для автоматизации работы магазина компьютерных игр.

***Цель курсовой работы:*** Изучение языка программирования C#, изучение среды Microsoft Visual Studio, создание базы данных Access, разработка и проектирование приложения.

***Задачами курсовой работы являются*:**

Разработка автоматизированной информационной системы, проверка работы созданного приложения.

***Выводы:*** В процессе курсового проекта был изучен язык C#, проверена работоспособность приложения.

Содержание

[Введение 4](#_Toc51591832)

[1 Техническое задание 5](#_Toc51591833)

[1.1 Назначение системы 5](#_Toc51591834)

[1.2 Требования к программе 5](#_Toc51591835)

[2 Анализ предметной области и выбор технологии, языка разработки 7](#_Toc51591836)

[2.1 Анализ предметной области 7](#_Toc51591837)

[2.2 Обоснование выбора СУБД 7](#_Toc51591838)

[1.3 Обоснование выбора языка программирования 8](#_Toc51591839)

[2 Разработка диаграммы вариантов использования 10](#_Toc51591840)

[3 Определение структуры программного продукта 12](#_Toc51591841)

[3.1 Структурная схема 12](#_Toc51591842)

[3.2 Информационная схема 13](#_Toc51591843)

[4 Описание реализации программного продукта 16](#_Toc51591844)

[5 Тестирование программного средства 21](#_Toc51591845)

[Заключение 24](#_Toc51591846)

[Список литература 25](#_Toc51591847)

# Введение

Ввиду быстрого развития информационных систем, а также уверенного роста количества хранимой информации, организации разных масштабов и направлений деятельности все чаще приходят к необходимости автоматизации производственных и управленческих процессов. Основной целью таких внедрений является увеличение скорости и качества работы сотрудников предприятия, повышение производительности труда и эффективности работы всех структурных подразделений.

Автоматизация — внедрение в бизнес-процессы электронного торгового оборудования и современного программного обеспечения — на сегодняшний день, наверное, лучший способ достаточно быстро повысить качество и скорость обслуживания, эффективность использования ресурсов предприятия, сократить время, затрачиваемое на различные операции, осуществлять эффективный учет, контролировать все аспекты деятельности предприятия, устранять всевозможные ошибки.

Целью данного курсового проекта является проектирование и разработка автоматизированной справочной системы магазина компьютерных игр

# 1 Техническое задание

## Назначение системы

Разрабатываемая автоматизированная справочная система магазина компьютерных игр предназначается для работы сотрудника магазина, который выполняет соответствующую работу. Использование текущей программы позволит автоматизировать процесс хранения информации о сотрудниках, продажах товара и также самого товара. Выполнять поиск и сортировку необходимой информации. Внедрение автоматизированной информационной системы позволит увеличить скорость и точность работы, избежать ошибок при вводе информации.

## 1.2 Требования к программе

Автоматизированная система должна соответствовать требованиям заказчика и выполнять функции:

* Ввод и просмотр информации о жанрах игр
* Ввод и просмотр информации о должностях
* Ввод и просмотр информации об издательствах
* Ввод и просмотр информации о сотрудниках
* Ввод и просмотр информации о товарах
* Ввод и просмотр информации о продажах
* Поиск сотрудников по полям
* Поиск товаров по полям
* Поиск продаж по полям
* Сортировка данных по полям

Для полноценной работы информационной системы необходимо хранение информации о:

* сотрудниках
* должностях
* жанрах игр
* издательствах
* товарах
* продажах

Для правильного запуска и бесперебойной работы приложения, необходимо, чтобы на пользовательской машине были установлены следующие программные компоненты:

* Windows 7 и выше
* MS Access 2010
* Microsoft Access Database Engine Redistributable 2010
* MS .NET Framework 4.5

Аппаратные требования:

* Процессор AMD Athlon 2 X3 445 Processor 1.9GHz
* ОЗУ: не менее 1Гб
* HDD: не менее 50Гб
* Видеоадаптер: DirectX9
* Видеопамять: не менее 100 МБ

Язык программирования должен обеспечить возможность реализации всех заявленных заказчиком требований и позволить реализовать простой, удобный интерфейс интуитивно понятный конечному пользователю программы.

# 2 Анализ предметной области и выбор технологии, языка разработки

## 2.1 Анализ предметной области

Объектом исследования является магазин игр.

Магазин игр представляет собой помещение, в котором предоставлены различные игры для разных систем в удобном для покупателя виде.

Задача сотрудников магазина (в частности продавцов), это продажа игр покупателям. Каждая игра представляет из себя набор множества данных, таких как название, жанры этой игры, цена продажи и прочее.

Эти игры продают сотрудники, которые могут быть как продавцами, так и стажерами, менеджерами, или даже самим директором.

## 2.2 Обоснование выбора СУБД

Microsoft Office Access — реляционная система управления базами данных (СУБД) корпорации Microsoft. Входит в состав пакета Microsoft Office. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самом Access можно писать приложения, работающие с базами данных.

Основными компонентами MS Access являются:

построитель таблиц;

построитель экранных форм;

построитель SQL-запросов;

построитель отчётов, выводимых на печать.

База данных — представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ). В виде базы данных может быть представлена любая информация: от коллекции фильмов до списка клиентов фирмы.

Microsoft Access удобен для фирм, которым необходимо классифицировать всю свою информацию: клиентов, партнеров, транспортные компании, сотрудников и многое другое. В данной программе есть возможность создания различных форм, отчётов, запросов и различных типов связей, которые позволяют работникам быстрее выполнять свои задания.

В данной программе есть возможность производить ввод информации, сортировать её, производить выборку, отображать фотографии и многое другое. Главное преимущество данной программы в том, что на экран компьютера или на печать выводится лишь информация, соответствующая тому или иному запросу.

Благодаря системе запросов можно делать выборку информации из одной таблицы, чтобы она отображалась в другой. Например, реляционная база данных может содержать таблицу со сведениями о покупателях, таблицу со сведениями о производителях и таблицу со сведениями о товаре. Таблица со сведениями о товаре может содержать поле для связи с таблицей с данными о покупателях и поле для связи с таблицей с данными о производителях. Таким образом, при изменении, например, номера телефона производителя данные можно изменить однократно только в таблице производителей, а не в описании каждого товара с участием данного производителя.

В Microsoft Access имеется большое количество шаблонов, позволяющих ускорить создание базы данных. Шаблон — это готовая база данных, содержащая все таблицы, запросы, формы и отчеты, необходимые для выполнения определенной задачи. Например, существуют шаблоны, используемые для отслеживания проблем, управления контактами или записи расходов. Некоторые шаблоны содержат несколько образцов записей, чтобы продемонстрировать их использование. Шаблоны баз данных можно использовать как есть или настроить в соответствии с конкретными требованиями.

С помощью Access можно создавать не только классические базы данных, но и удобные веб-приложения для работы с базами данных, которые значительно облегчают ведение бизнеса. Данные автоматически сохраняются в базе данных SQL, поэтому они надежно защищены, а приложения можно с легкостью использовать совместно с коллегами.

## 1.3 Обоснование выбора языка программирования

Разработка приложения будет выполняться на языке С# в среде Visual Studio 2019.

C - объектно-ориентированный язык программирования для платформы .NET. Он разработан в 2000 году Андерсом Хейлсбергом, Скоттом Вилтамутом и Питером Гольде под эгидой Microsoft Research. Основным постулатом С# является высказывание: «всякая сущность есть объект». Язык основан на строгой компонентной архитектуре и реализует передовые механизмы обеспечения безопасности кода.

C# был создан специально для технологии ASP.NET. В то же время на C# полностью написана и сама ASP.NET.

C# -- это полнофункциональный объектно-ориентированный язык, который поддерживает все три «столпа» объектно-ориентированного программирования: инкапсуляцию, наследование и полиморфизм. Он имеет прекрасную поддержку компонентов, надежен и устойчив благодаря использованию «сборки мусора», обработки исключений, безопасности типов.

Язык C# разрабатывался «с нуля» и вобрал в себя много полезных свойств таких языков, как C++, Java, Visual Basic, а также Pascal, Delphi и др. При этом необходимость обратной совместимости с предыдущими версиями отсутствовала, что позволило языку C# избежать многих отрицательных сторон своих предшественников.

Как и Java, C# разрабатывался для Интернет и примерно 75% его синтаксических возможностей аналогичны языку программирования Java, его также называют «очищенной версией Java». 10% подобны языку программирования C++, а 5% -- заимствованы из языка программирования Visual Basic. Объем новых концептуальных идей в языке C# около 10%.

Выделение и объединение лучших идей современных языков программирования делает язык C# не просто суммой их достоинств, а языком программирования нового поколения, поэтому я выбрал именно этот язык для написания программы

.

# 2 Разработка диаграммы вариантов использования

Диаграмма вариантов использования представляет собой основную концептуальную модель системы.

Назначение данной диаграммы состоит в следующем: проектируемая программная система представляется в форме так называемых вариантов использования, с которыми взаимодействуют внешние сущности или актеры. При этом актером или действующим лицом называется любой объект, субъект или система, взаимодействующая с моделируемой бизнес-системой извне.

Это может быть человек, техническое устройство, программа или любая другая система, которая служит источником воздействия на моделируемую систему так, как определит разработчик. Вариант использования служит для описания сервисов, которые система предоставляет актеру. Другими словами каждый вариант использования определяет набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой и собственно выполнение вариантов использования.

В соответствии с техническим заданием, выделим ряд функций, которыми должна обладать разрабатываемая автоматизированная система:

* Хранение жанров для работы с товаром
* Хранение должностей и заработной платы сотрудников
* Хранение издателей
* Хранение товаров
* Доступ к списку продаж
* Работа с сотрудниками
* Поиск данных в таблицах по полям.

Таким образом, можно сформировать диаграмму вариантов использования для разрабатываемой системы (рис. 2.1):

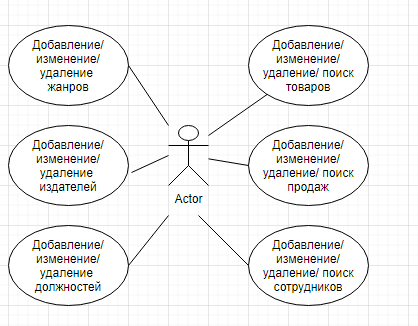


Рисунок 2.1 – Диаграмма вариантов использования

# 3 Определение структуры программного продукта

## 3.1 Структурная схема

В соответствии с указанным выше функционалом, программное средство будет представлять собой набор классов и их форм для обеспечения взаимодействия пользователь – программа.

Формы, подлежащие разработке:

* Главная – загружаемая при старте программы форма. Она будет являться сопроводительной формой, а так же предоставлять меню для доступа к другим формам.
* Продажи – форма работа с продажами. На ней можно как добавить новую продажу, так и изменить/удалить старую. Кроме того, можно посмотреть какой сотрудник, что и когда продал, какие товары продаются лучше других.
* Товары – форма работы с товарами. Основная работа формы заключается в добавлении/изменении/удалении товаров, с которыми ведется работа. На форме также присутствует поиск для быстрого доступа к определенному товару.
* Сотрудники – форма, отображающая всех сотрудников. Использовать ее можно при приходе/уходе сотрудника, а также для доступа к их данным, таким как адрес или мобильный телефон.
* Жанры – форма для работы с жанрами. Так как каждая игра содержит минимум несколько жанров, нужно место, где можно их добавлять или удалять.
* Должности – форма работы с должностями и их заработной платой.
* Издатели – форма для работы с данными по издателям игр

Одним из наиболее важных показателей в работе пользователя с программой является удобство ее использования и «понятность» интерфейса, грамотное расположение и взаимодействие структурных элементов программы. Для правильной организации перехода между указанными формами необходимо составить структурную схему программного обеспечения (рис. 3.1):

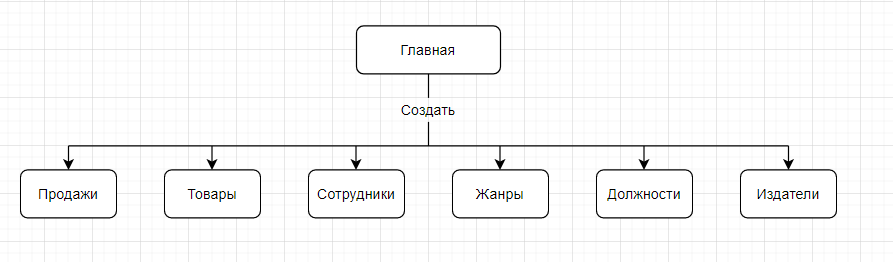


Рисунок 3.1 – Структурная схема программного продукта

## 3.2 Информационная схема

Предполагается, что разрабатываемое ПО должно хранить всю необходимую информацию в базе данных и по запросу предоставлять ее пользователю. Определим минимальный необходимый набор данных для хранения:

* Название товара
* Цена товара
* Жанры товара
* Издатель товара
* Год выпуска
* Ограничение по возрасту
* Название жанра
* Сотрудник что продал
* Должность сотрудника
* Заработная плата сотрудника
* ФИО сотрудника
* Адрес сотрудника
* Телефон сотрудника
* Дата продажи товара
* Сумма продажи
* Количество товара

В соответствии с требованиями 1НФ, 2НФ и 3НФ все атрибуты были распределены по таблицам, база данных нормализована. Таким образом, информационная схема принимает вид (рис. 3.2):

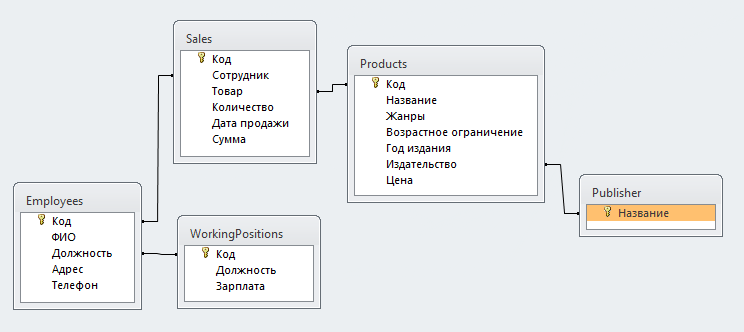


Рисунок 3.2 – Информационная модель

На информационной модели видно, что база данных состоит из 5 таблиц. Все таблицы связаны между собой не идентифицирующей связью мощностью «один-ко-многим» с помощью внешних ключей.

Для правильного хранения информации при создании базы данных необходимо указать типы данных, подлежащих хранению в текущем атрибуте. В таблицах 3.1 – 3.4 представлены структуры таблиц базы данных:

Таблица 3.1 – Структура полей таблицы «Employees»

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Ключ | Тип поля | Длина |
| Код | PK | Счетчик |  |
| ФИО |  | Текстовый | 20 |
| Должность |  | Текстовый | 150 |
| Адрес |  | Текстовый | 150 |
| Телефон |  | Текстовый | 20 |

Таблица 3.2 – Структура полей таблицы «Genres»

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Ключ | Тип поля | Длина |
| Название | PK | Текстовый | 255 |

Таблица 3.3 – Структура полей таблицы «Products»

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Ключ | Тип поля | Длина |
| Код | PK | Счетчик |  |
| Название |  | Текстовый | 255 |
| Жанры |  | Текстовый | 255 |
| Возрастное ограничение |  | Числовой | 15 |
| Год издания |  | Числовой | 15 |
| Издательство |  | Текстовый | 255 |
| Цена |  | Числовой | 15 |

Таблица 3.4 – Структура полей таблицы «Publisher»

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Ключ | Тип поля | Длина |
| Название | PK | Текстовый | 255 |

Таблица 3.3 – Структура полей таблицы «Sales»

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Ключ | Тип поля | Длина |
| Код | PK | Счетчик |  |
| Сотрудник |  | Числовой | 15 |
| Товар |  | Числовой | 15 |
| Количество |  | Числовой | 15 |
| Дата продажи |  | Дата/время |  |
| Сумма |  | Числовой | 15 |

Таблица 3.4 – Структура полей таблицы «WorkingPositions»

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Ключ | Тип поля | Длина |
| Код | PK | Счетчик |  |
| Должность |  | Текстовый | 255 |
| Зарплата |  | Числовой | 15 |

# 4 Описание реализации программного продукта

Реализация программы заключается прежде всего не в построении форм и классов, а создании логики взаимодействия данных с конечным пользователем. В среде проектирования Visual Studio существует ряд компонентов, позволяющих взаимодействовать с таблицами базы данных. В отдельных свойствах данных компонентов можно прописывать запросы и команды к базе данных.

Рассмотрим подробнее данные запросы и команды.

1. Запрос на получение списка сотрудников

*SELECT Код, ФИО, Должность, Адрес, Телефон FROM Employees*

1. Запрос на получение списка сотрудников с условием совпадения части имени

*SELECT Код, ФИО, Должность, Адрес, Телефон*

*FROM Employees*

*WHERE (ФИО LIKE ?)*

1. Запрос на получение списка сотрудников с условием совпадения части адреса

*SELECT Код, ФИО, Должность, Адрес, Телефон*

*FROM Employees*

*WHERE (Адрес LIKE ?)*

1. Запрос на получение списка сотрудника с условием на совпадении должности

*SELECT Код, ФИО, Должность, Адрес, Телефон*

*FROM Employees*

*WHERE (Должность = ?)*

1. Запрос на получение списка сотрудников с условием совпадения телефона

*SELECT Код, ФИО, Должность, Адрес, Телефон*

*FROM Employees*

*WHERE (Телефон LIKE ?)*

1. Запрос на получение списка должностей и заработных плат

*SELECT Код, Должность, Зарплата FROM WorkingPositions*

1. Запрос на получение списка жанров

*SELECT Название FROM Genres*

1. Запрос на получение списка издателей

*SELECT Название FROM Publisher*

1. Запрос на получение списка продаж

*SELECT Код, Сотрудник, Товар, Количество, [Дата продажи], Сумма FROM Sales*

1. Запрос на получение списка продаж с условием совпадения даты

*SELECT Код, Сотрудник, Товар, Количество, [Дата продажи], Сумма*

*FROM Sales*

*WHERE ([Дата продажи] BETWEEN ? AND ?)*

1. Запрос на получение списка продаж с условием совпадения количества

*SELECT Код, Сотрудник, Товар, Количество, [Дата продажи], Сумма*

*FROM Sales*

*WHERE (Количество = ?)*

1. Запрос на получение списка продаж с условием совпадения продавшего сотрудника

*SELECT Код, Сотрудник, Товар, Количество, [Дата продажи], Сумма*

*FROM Sales*

*WHERE (Сотрудник = ?)*

1. Запрос на получение списка продаж с условием совпадения суммы

*SELECT Код, Сотрудник, Товар, Количество, [Дата продажи], Сумма*

*FROM Sales*

*WHERE (Сумма = ?)*

1. Запрос на получение списка продаж с условием совпадения товара

*SELECT Код, Сотрудник, Товар, Количество, [Дата продажи], Сумма*

*FROM Sales*

*WHERE (Товар = ?)*

1. Запрос на получение списка товара

*SELECT Код, Название, Жанры, [Возрастное ограничение], [Год издания], Издательство, Цена FROM Products*

1. Запрос на получение списка товара с условием совпадения возрастного ограничения

*SELECT Код, Название, Жанры, [Возрастное ограничение], [Год издания], Издательство, Цена*

*FROM Products*

*WHERE ([Возрастное ограничение] = ?)*

1. Запрос на получение списка товара с условием совпадения года издания

*SELECT Код, Название, Жанры, [Возрастное ограничение], [Год издания], Издательство, Цена*

*FROM Products*

*WHERE ([Год издания] = ?)*

1. Запрос на получение списка товара с условием совпадения жанров

*SELECT Код, Название, Жанры, [Возрастное ограничение], [Год издания], Издательство, Цена*

*FROM Products*

*WHERE (Жанры LIKE ?)*

1. Запрос на получение списка товара с условием совпадения издательства

*SELECT Код, Название, Жанры, [Возрастное ограничение], [Год издания], Издательство, Цена FROM Products*

1. Запрос на получение списка товара с условием продукции с условием совпадения названия

*SELECT Код, Название, Жанры, [Возрастное ограничение], [Год издания], Издательство, Цена FROM Products*

1. Запрос на получение списка товара с условием продукции с условием совпадения цены

*SELECT Код, Название, Жанры, [Возрастное ограничение], [Год издания], Издательство, Цена FROM Products*

Непосредственная разработка форм и классов приложения выполнялась в среде Microsoft Visual Studio, которая содержит множество компонентов для создания удобного интерфейса пользователя. Обработка событий управления элементами осуществляется с помощью методов классов. Диаграмма классов представлена в приложении 1.

Так как многие компоненты напрямую взаимодействую с данными, то при сохранении любых изменений, данные проходят проверку соответствия указанному типу данных и заполненности полей (допустимо ли значение null). Однако в программе есть алгоритм, который выполняет непосредственную проверку, программно. Это алгоритм проверки целостности данных перед добавлением/изменением. Блок-схема алгоритма приведена на рисунке 4.1.

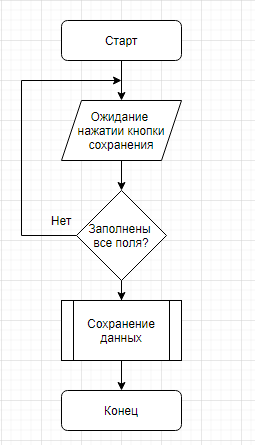


Рисунок 4.1 – Блок-схема алгоритма сохранения

# 5 Тестирование программного средства

Тестирование программного обеспечения — процесс исследования, испытания [программного продукта](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), имеющий две различные цели:

* продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
* выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

Существует множество различных видов тестирования, однако для проверки работоспособности разработанного программного средства применим функциональное тестирование.

Функциональное тестирование — это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает.

Результаты проведенного тестирования представлены в таблице 5.1:

Таблица 5.1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Название функции | Описание | Пример |
| 1 | 2 | 3 | 4 |
| 1 | Главное меню | Программа не должна вызывать трудности при запуске. При запуска первым делом отображается главное меню |  |
| 2 | Список продаж | При переходе на форму продаж, пользователю должен быть показан список совершенных продаж. Работа с данными продаж происходит на текущей форме |  |

Продолжение таблицы 5.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 3 | Товары | Сведения о товарах: их названии, цене, издателе, возрастном ограничении. Работа с данными товаров происходит на текущей форме |  |
| 4 | Сотрудники | Список всех зарегистрированных сотрудников и их должности |  |
| 5 | Выпадающее меню справочники | Список справочников, использующих в выпадающих списках |  |
| 6 | Жанры | Список жанров для игр |  |
| 7 | Издательства | Список игровых издательств |  |
| 8 | Должности | Список должностей сотрудников и заработная плата |  |

# Заключение

В рамках курсового проекта была разработана автоматизированная информационная система для работы магазина продажи игр. Программа позволяет хранить данные о произведенных продажах, текущих товарах и сотрудниках. Для всех хранящихся данных реализован удобный интерфейс не только просмотра, но также и редактирования данных, в том числе добавления и удаления.

По результатам тестирования можно сделать вывод, что информационная система работает корректно, выполняет все заявленные требования.

Программа является универсальной и может быть использована в любом магазине продажи игр.

# Список литературы

1. Троэлсен, Э. C# и платформа .NET. Питер, Санкт-Петербург, 2005. – 230 с.

2. Дюваль, П. Непрерывная интеграция. Улучшение качества программного обеспечения и снижение риска / П. Дюваль, С. Матиас, Э. Гловер. – М.: Вильямс, 2008. – 240 с.

3. Лахатин, А.С. Языки программирования. Учеб. пособие / А.С. Лахатин, Л.Ю. Искакова. − Екатеринбург, 1998. – 548 с.

4. Васильев, А. Программирование на C# для начинающих. Основные сведения. – Эксмо, 2018. – 592 с

# Приложение

Диаграмма классов

