Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра информатики

Отчет по производственной практике

Выполнил:

студент группы 253503

Абрамов Д.А.

Руководитель от вуза:

Новиков В.И.

Руководитель от предприятия:

Торопов В.В.

Минск 2016

Содержание

[Введение 3](#__RefHeading___Toc750_450552699)

[Структура предприятия](#__RefHeading___Toc754_450552699) 3

[Сведения о программной платформе и проекте](#__RefHeading___Toc756_450552699) 5

[NET Framework](#__RefHeading___Toc758_450552699) 5

[C Sharp](#__RefHeading___Toc760_450552699) 7

[WPF](#__RefHeading___Toc762_450552699) 9

[XAML 1](#__RefHeading___Toc766_450552699)1

[XML 13](#__RefHeading___Toc768_450552699)

[MVVM 14](#__RefHeading___Toc770_450552699)

Microsoft SQL Server…………………………………………………………..16

T-SQL…………………………………………………………………………..17

ADO.NET……………………………………………………………………....18

[Заключение 19](#__RefHeading___Toc778_450552699)

[Список использованных материалов 20](#__RefHeading___Toc780_450552699)

## 

## Введение

## Структура предприятия

ООО «Омегасофтвер» - компания в сегменте разработки программного обеспечения для промышленных предприятий.

Основным продуктом компании OmegaSoftware является Корпоративная информационная система предприятия “Omega Production”. На сегодняшний день этот программный продукт является уникальным в СНГ, так как охватывает все переделы производственного процесса промышленного предприятия: от разработки изделия и технической подготовки производства до сопровождения продукции в гарантийный и постгарантийный периоды.

Работники компании имеют обширный опыт в разработке решений, которые отвечают всем потребностям клиентов и даже превышают их ожидания. На протяжении многих лет компания сумела создать прочную команду IT-специалистов.

Компания постоянно совершенствует стандарты качества предлагаемых продуктов и услуг. Все это делается с целью сократить время производства и минимизировать возможные риски, связанные с разработкой комплексных программных продуктов.

Для разработки своих продуктов компания использует следующие технологии: Java SE/EE, мобильные технологии, .NET Framework.

Организация в своей деятельности руководствуется Законами Республики Беларусь, указами Президента Республики Беларусь, другими нормативными актами и уставом компании ООО «Омегасофтвер».

Компания была создана в 1992 году, и на сегодняшний день имеет представительства в Минске, Москве, Ижевске и Омске, где работают более 100 высококвалифицированных специалистов.

# Сведения о программной платформе и проекте

## Программная платформа

**.NET Framework**

Платформа .NET Framework — это интегрированный компонент Windows, который поддерживает создание и выполнение нового поколения приложений и веб-служб XML. При разработке платформы .NET Framework учитывались следующие цели.

* Обеспечение согласованной объектно-ориентированной среды программирования для локального сохранения и выполнения объектного кода, для локального выполнения кода, распределенного в Интернете, либо для удаленного выполнения.
* Обеспечение среды выполнения кода, минимизирующей конфликты при развертывании программного обеспечения и управлении версиями.
* Обеспечение среды выполнения кода, гарантирующей безопасное выполнение кода, включая код, созданный неизвестным или не полностью доверенным сторонним изготовителем.
* Обеспечение среды выполнения кода, исключающей проблемы с производительностью сред выполнения сценариев или интерпретируемого кода.
* Обеспечение единых принципов работы разработчиков для разных типов приложений, таких как приложения Windows и веб-приложения.
* Разработка взаимодействия на основе промышленных стандартов, которое обеспечит интеграцию кода платформы .NET Framework с любым другим кодом.

Двумя основными компонентами платформы .NET Framework являются общеязыковая среда выполнения (CLR) и библиотека классов .NET Framework. Основой платформы .NET Framework является среда CLR. Среду выполнения можно считать агентом, который управляет кодом во время выполнения и предоставляет основные службы, такие как управление памятью, управление потоками и удаленное взаимодействие. При этом накладываются условия строгой типизации и другие виды проверки точности кода, обеспечивающие безопасность и надежность. Фактически основной задачей среды выполнения является управление кодом. Код, который обращается к среде выполнения, называют управляемым кодом, а код, который не обращается к среде выполнения, называют неуправляемым кодом. Другой основной компонент платформы .NET Framework, библиотека классов, представляет полную объектно-ориентированную коллекцию типов, которые применяются для разработки приложений, начиная от обычных, запускаемых из командной строки или с графическим интерфейсом пользователя, и заканчивая приложениями, использующими последние технологические возможности ASP.NET, такие как Web Forms и веб-службы XML.

Платформа .NET Framework может размещаться неуправляемыми компонентами, которые загружают среду CLR в собственные процессы и запускают выполнение управляемого кода, создавая таким образом программную среду, позволяющую использовать средства как управляемого, так и неуправляемого выполнения. Платформа .NET Framework не только предоставляет несколько базовых сред выполнения, но также поддерживает разработку базовых сред выполнения независимыми производителями.

**C Sharp**

**C#** — [объектно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Разработан в [1998](https://ru.wikipedia.org/wiki/1998)—[2001 годах](https://ru.wikipedia.org/wiki/2001_%D0%B3%D0%BE%D0%B4) группой инженеров под руководством [Андерса Хейлсберга](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D0%B9%D0%BB%D1%81%D0%B1%D0%B5%D1%80%D0%B3,_%D0%90%D0%BD%D0%B4%D0%B5%D1%80%D1%81) в компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) как язык разработки приложений для платформы [Microsoft .NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework) и впоследствии был стандартизирован как [ECMA](https://ru.wikipedia.org/wiki/ECMA)-334 и [ISO](https://ru.wikipedia.org/wiki/ISO)/[IEC](https://ru.wikipedia.org/wiki/IEC) 23270.

C# относится к семье языков с [C-подобным синтаксисом](https://ru.wikipedia.org/wiki/C-%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81), из них его синтаксис наиболее близок к [C++](https://ru.wikipedia.org/wiki/C%2B%2B) и [Java](https://ru.wikipedia.org/wiki/Java). Язык имеет [статическую типизацию](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), поддерживает [полиморфизм](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BC%D0%BE%D1%80%D1%84%D0%B8%D0%B7%D0%BC_%28%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29), [перегрузку операторов](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BE%D0%B2) (в том числе операторов явного и неявного приведения типа), [делегаты](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BB%D0%B5%D0%B3%D0%B0%D1%82_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29), атрибуты, [события](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B9%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [свойства](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29), [обобщённые](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) типы и методы, [итераторы](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29#.D0.93.D0.B5.D0.BD.D0.B5.D1.80.D0.B0.D1.82.D0.BE.D1.80.D1.8B), [анонимные функции](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) с поддержкой [замыканий](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BC%D1%8B%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29), [LINQ](https://ru.wikipedia.org/wiki/Language_Integrated_Query), [исключения](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D1%81%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B9), [комментарии](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B8_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29) в формате [XML](https://ru.wikipedia.org/wiki/XML).

Переняв многое от своих предшественников — языков [C++](https://ru.wikipedia.org/wiki/C%2B%2B), [Pascal](https://ru.wikipedia.org/wiki/Pascal_%28%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%29), [Модула](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D0%B0-2), [Smalltalk](https://ru.wikipedia.org/wiki/Smalltalk) и, в особенности, [Java](https://ru.wikipedia.org/wiki/Java_%28%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%29) — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает [множественное наследование](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BD%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) классов (между тем допускается [множественное наследование интерфейсов](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%28%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29#.D0.9C.D0.BD.D0.BE.D0.B6.D0.B5.D1.81.D1.82.D0.B2.D0.B5.D0.BD.D0.BD.D0.BE.D0.B5_.D0.BD.D0.B0.D1.81.D0.BB.D0.B5.D0.B4.D0.BE.D0.B2.D0.B0.D0.BD.D0.B8.D0.B5_.D0.B8_.D1.80.D0.B5.D0.B0.D0.BB.D0.B8.D0.B7.D0.B0.D1.86.D0.B8.D1.8F_.D0.B8.D0.BD.D1.82.D0.B5.D1.80.D1.8)).

C# разрабатывался как язык программирования прикладного уровня для [CLR](https://ru.wikipedia.org/wiki/Common_Language_Runtime) и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает [BCL](https://ru.wikipedia.org/wiki/Base_Class_Library). Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем (однако, эта закономерность была нарушена с выходом C# 3.0, представляющего собой расширения языка, не опирающиеся на расширения платформы .NET). CLR предоставляет C#, как и всем другим [.NET](https://ru.wikipedia.org/wiki/.NET_Framework)-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, [Сборка мусора](https://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29) не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на [VB.NET](https://ru.wikipedia.org/wiki/Visual_Basic_.NET), [J#](https://ru.wikipedia.org/wiki/Visual_J%E2%99%AF) и др.

**WPF**

*Windows Presentation Foundation* — система для построения клиентских приложений [Windows](http://ru.wikipedia.org/wiki/Windows) с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе [.NET Framework](http://ru.wikipedia.org/wiki/.NET_Framework) (начиная с версии [3.0](http://ru.wikipedia.org/wiki/.NET_Framework_3.0)), использующая язык [XAML](http://ru.wikipedia.org/wiki/XAML)[.](http://ru.wikipedia.org/wiki/Windows_Presentation_Foundation#cite_note-2)

WPF предустановлена в [Windows Vista](http://ru.wikipedia.org/wiki/Windows_Vista) ([.NET Framework](http://ru.wikipedia.org/wiki/.NET_Framework) 3.0), [Windows 7](http://ru.wikipedia.org/wiki/Windows_7) ([.NET Framework](http://ru.wikipedia.org/wiki/.NET_Framework) 3.5 SP1), [Windows 8](http://ru.wikipedia.org/wiki/Windows_8)(.NET Framework 4.0 и 4.5), [Windows 8.1](http://ru.wikipedia.org/wiki/Windows_8.1) (.NET Framework 4.5.1). С помощью WPF можно создавать широкий спектр как автономных, так и запускаемых в [браузере](http://ru.wikipedia.org/wiki/Браузер) приложений[.](http://ru.wikipedia.org/wiki/Windows_Presentation_Foundation#cite_note-msdn_aa970268-3)

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык [XAML](http://ru.wikipedia.org/wiki/XAML) (Extensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление[.](http://ru.wikipedia.org/wiki/Windows_Presentation_Foundation#cite_note-msdn_aa970268-3)

Графической технологией, лежащей в основе WPF, является [DirectX](http://ru.wikipedia.org/wiki/DirectX), в отличие от [Windows Forms](http://ru.wikipedia.org/wiki/Windows_Forms), где используется [GDI](http://ru.wikipedia.org/wiki/GDI)/[GDI+](http://ru.wikipedia.org/wiki/GDI%2B). Производительность [WPF](http://ru.wikipedia.org/wiki/WPF) выше, чем у [GDI+](http://ru.wikipedia.org/wiki/GDI%2B) за счёт использования аппаратного ускорения графики через [DirectX](http://ru.wikipedia.org/wiki/DirectX).

Также существует урезанная версия [CLR](http://ru.wikipedia.org/wiki/CLR), называющаяся [WPF/E](http://ru.wikipedia.org/wiki/WPF/E), она же известна как [Silverlight](http://ru.wikipedia.org/wiki/Silverlight).

[XAML](http://ru.wikipedia.org/wiki/XAML) представляет собой [XML](http://ru.wikipedia.org/wiki/XML), в котором фактически реализованы классы [.NET Framework](http://ru.wikipedia.org/wiki/.NET_Framework). Также реализована модель разделения кода и дизайна, позволяющая кооперироваться программисту и дизайнеру. Кроме того, есть встроенная поддержка стилей элементов, а сами элементы легко разделить на [элементы управления](http://ru.wikipedia.org/wiki/Элемент_управления) второго уровня, которые, в свою очередь, разделяются до уровня векторных фигур и свойств/действий. Это позволяет легко задать стиль для любого элемента.

Для работы с WPF требуется любой [.NET](http://ru.wikipedia.org/wiki/.NET_Framework)-совместимый язык. В этот список входит множество языков: [C#](http://ru.wikipedia.org/wiki/C_Sharp), [VB](http://ru.wikipedia.org/wiki/VB), [C++](http://ru.wikipedia.org/wiki/Visual_C%2B%2B), [Ruby](http://ru.wikipedia.org/wiki/IronRuby), [Python](http://ru.wikipedia.org/wiki/IronPython), [Delphi (Prism)](http://ru.wikipedia.org/wiki/Oxygene_(язык_программирования)), [Lua](http://ru.wikipedia.org/w/index.php?title=Lua.NET&action=edit&redlink=1) и многие другие. Для полноценной работы может быть использована как [Visual Studio](http://ru.wikipedia.org/wiki/Visual_Studio), так и [Expression Blend](http://ru.wikipedia.org/wiki/Expression_Blend). Первая ориентирована на программирование, а вторая — на дизайн и позволяет делать многие вещи, не прибегая к ручному редактированию [XAML](http://ru.wikipedia.org/wiki/XAML). Примеры этому — анимация, стилизация, состояния, создание элементов управления и так далее.

## 

XAML

**XAML**— расширяемый [язык разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) для приложений - основанный на [XML](https://ru.wikipedia.org/wiki/XML) язык разметки для декларативного программирования приложений, разработанный [Microsoft](https://ru.wikipedia.org/wiki/Microsoft).

Модель приложений Vista включает объект Application. Его набор свойств, методов и событий позволяет объединить веб-документы в связанное приложение. Объект Application контролирует выполнение программы и генерирует события для пользовательского кода. Документы приложения пишутся на XAML. Впрочем, с помощью XAML описывается, прежде всего, пользовательский интерфейс. Логика приложения по-прежнему управляется процедурным кодом (С#, VB, JavaScript и т. д.). XAML может использоваться как для браузер-базированных приложений, так и для локальных настольных приложений.

XAML включает основные четыре категории элементов: панели, элементы управления, элементы, связанные с документом и графические фигуры. Заявлено 7 классов панелей, которые задают принципы отображения вложенных в них элементов. Для задания положения элементов относительно границ родительской панели используются атрибуты на манер свойств в объектно-ориентированных языках. Подобный синтаксис не очень вяжется с рекомендациями [CSS](https://ru.wikipedia.org/wiki/CSS), но будет привычен программистам настольных приложений.

Приложения, объявленные в XAML, могут включать множество страниц. Элемент управления PageViewer позволяет разбивать содержание на страницы и обеспечивает навигацию по ним. Элемент ContextMenu помогает в создании навигационных меню приложения. Код процедурного языка может быть размещён непосредственно в файле XAML или же назначен при сборке проекта.

XAML широко используется в [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework) 3.0, в особенности в [Windows Presentation Foundation](https://ru.wikipedia.org/wiki/Windows_Presentation_Foundation) (WPF), [Windows Workflow Foundation](https://ru.wikipedia.org/wiki/Windows_Workflow_Foundation) (WWF) и [Silverlight](https://ru.wikipedia.org/wiki/Silverlight). В WPF XAML используется как язык разметки пользовательского интерфейса, для определения элементов пользовательского интерфейса, привязки данных, поддержки событий и др. свойств. В WWF, при помощи XAML можно определять последовательности выполняемых действий (workflows).

XAML файлы можно создавать и редактировать при помощи инструментов визуального конструирования, таких как: Microsoft Expression Blend, Microsoft Visual Studio, WPF visual designer. Также, их можно создавать при помощи стандартного текстового редактора, редактора кода такого как: XAMLPad, или графического редактора, такого как Vectropy.

Все созданное или реализованное в XAML может быть выражено при помощи более традиционных .NET языков, таких как: C# или Visual Basic.NET. Однако, ключевым аспектом технологии является уменьшение сложности используемых для обработки XAML инструментов, так как XAML основан на XML. В результате этого появляется множество продуктов, создающих основанные на XAML приложения. Поскольку XAML базируется на XML, у разработчиков и дизайнеров существует возможность одновременно работать над содержимым без необходимости компиляции.

XML

XML разрабатывался как язык с простым формальным [синтаксисом](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29), удобный для создания и обработки документов программам и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Сочетание простого формального синтаксиса, удобства для человека, расширяемости, а также базирование на кодировках [Юникод](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4) для представления содержания документов привело к широкому использованию как собственно XML, так и множества производных специализированных языков на базе XML в самых разнообразных программных средствах.

MVVM

Шаблон **Model-View-ViewModel** (MVVM) — применяется при проектировании архитектуры приложения. Первоначально был представлен сообществу Джоном Госсманом (John Gossman) в 2005 году как модификация шаблона Presentation Model. MVVM ориентирован на современные платформы разработки, такие как [Windows Presentation Foundation](https://ru.wikipedia.org/wiki/WPF), [Silverlight](https://ru.wikipedia.org/wiki/Silverlight) от компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft), [ZK framework](https://ru.wikipedia.org/wiki/ZK_%28framework%29).

MVVM используется для разделения модели и её представления, что необходимо для изменения их отдельно друг от друга. Например, разработчик задает логику работы с данными, а дизайнер соответственно работает с пользовательским интерфейсом.

В шаблонах проектирования [MVC](https://ru.wikipedia.org/wiki/Model-View-Controller)/[MVP](https://ru.wikipedia.org/wiki/Model-View-Presenter) изменения в пользовательском интерфейсе не влияют непосредственно на Mодель, а предварительно идут через Контроллер ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Controller*) или Presenter. В таких технологиях как [WPF](https://ru.wikipedia.org/wiki/WPF) и [Silverlight](https://ru.wikipedia.org/wiki/Silverlight) есть концепция «связывания данных», позволяющая связывать данные с визуальными элементами в обе стороны. Следовательно, при использовании этого приема применение модели MVC становится крайне неудобным из-за того, что привязка данных к представлению напрямую не укладывается в концепцию MVC/MVP.

Шаблон MVVM делится на три части:

* *Модель* ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Model*), так же, как в классической MVC, Модель представляет собой фундаментальные данные, необходимые для работы приложения.
* *Представление* ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *View*) — это графический интерфейс, то есть окно, кнопки и т. п. Представление является подписчиком на событие изменения значений свойств или команд, предоставляемых Моделью представления. В случае, если в Модели представления изменилось какое-либо свойство, то она оповещает всех подписчиков об этом, и Представление, в свою очередь, запрашивает обновленное значение свойства из Модели представления. В случае, если пользователь воздействует на какой-либо элемент интерфейса, Представление вызывает соответствующую команду, предоставленную Моделью представления.
* *Модель представления* ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *ViewModel*) является, с одной стороны, абстракцией Представления, а с другой, предоставляет обёртку данных из Модели, которые подлежат связыванию. То есть, она содержит Модель, которая преобразована к Представлению, а также содержит в себе команды, которыми может пользоваться Представление, чтобы влиять на Модель.

**Microsoft SQL Server**

## 

Microsoft SQL Server представляет собой платформу для работы с базами данных, обеспечивающую возможность крупномасштабной оперативной обработки транзакций (OLTP), хранения данных и работы с приложениями для электронной торговли; а также является платформой бизнес-аналитики для создания решений по интеграции данных, анализу и составлению отчетов.

SQL Server представляет «среды» для облегчения выполнения задач разработки и управления: среда SQL Server Management Studio и среда Business Intelligence Development Studio. В среде Management Studio можно разрабатывать решения компонента SQL Server Database Engine и решения по уведомлению и управлять ими, управлять развернутыми решениями служб Analysis Services, запускать пакеты служб Integration Services и управлять ими, управлять серверами, отчетами и моделями отчетов служб Reporting Services. В среде BI Development Studio можно разрабатывать решения бизнес-аналитики: проекты служб Analysis Services используются для разработки кубов, измерений и структур интеллектуального анализа данных; проекты служб Reporting Services — для создания отчетов; проекты моделей отчетов — для определения моделей отчетов и проекты служб Integration Services — для пакетов.

## T-SQL

Transact-SQL (T-SQL) — процедурное расширение языка SQL, созданное компанией Microsoft (для Microsoft SQL Server) и Sybase (для Sybase ASE).

Язык Transact-SQL является ключом к использованию MS SQL Server. Все приложения, взаимодействующие с экземпляром MS SQL Server, независимо от их реализации и пользовательского интерфейса, отправляют серверу инструкции Transact-SQL.

SQL был расширен такими дополнительными возможностями как:

* управляющие операторы,
* локальные и глобальные переменные,
* различные дополнительные функции для обработки строк, дат, математики и т. п.,
* поддержка аутентификации Microsoft Windows.

**ADO.NET**

ADO.NET— технология, предоставляющая доступ к [данным](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5) для приложений, основанных на [Microsoft .NET](https://ru.wikipedia.org/wiki/Microsoft_.NET). Является не развитием более ранней технологии [ADO](https://ru.wikipedia.org/wiki/ADO), а самостоятельной технологией, частью [фреймворка .NET](https://ru.wikipedia.org/wiki/.NET_Framework). В отличие от классической ADO, которая была в основном предназначена для тесно связанных клиент-серверных систем, ADO.NET больше нацелена на автономную работу с помощью объектов DataSet. Эти типы представляют локальные копии любого количества взаимосвязанных таблиц данных, каждая из которых содержит набор строк и столбцов. Объекты DataSet позволяют вызывающей сборке (наподобие веб-страницы или программы, выполняющейся на настольном компьютере) работать с содержимым DataSet, изменять его, не требуя подключения к источнику данных, и отправлять обратно блоки измененных данных для обработки с помощью соответствующего адаптера данных. Но, пожалуй, самое фундаментальное различие между классической ADO и ADO.NET состоит в том, что ADO.NET является управляемой кодовой библиотекой, и, значит, подчиняется тем же правилам, что и любая управляемая библиотека. Типы, составляющие ADO.NET, используют протокол управления памятью CLR, принадлежат к той же системе типов (классы, интерфейсы, перечисления, структуры и делегаты), и доступ к ним возможен с помощью любого языка .NET. Классы ADO.NET находятся в сборке System.Data.dll.

От других API ADO.NET отличает то, что она не взаимодействует с системами управления базами данных напрямую. Вместо этого используются поставщики данных (data provider), которые инкапсулируют механизм работы с конкретной СУБД. Такой подход очень гибок, позволяет создавать адаптеры для любой СУБД и полностью использовать её особенности.

# Заключение

В ходе производственной практики была изучена структура предприятия, используемые технологии при разработке ПО. Был получен опыт в сфере промышленной разработки программного обеспечения.

В ходе практики были получены навыки работы по специальности. Был повышен уровень владения вышеизложенными технологиями разработки программного обеспечения, благодаря интересному и познавательному индивидуальному заданию, а также благодаря хорошем и ответственным руководителям практики.

# Список использованных материалов

1. Нейгел К. И др. C# 5.0 и платформа .NET 4.5 для профессионалов, М.: Диалектика, 2013

2. Рендольф Н., Гарднер Д., Минутилло М., Андерсон К. Microsoft Visual Studio 2010 для профессионалов, М.: Диалектика, 2011

3. Шилдт Г. C# 4.0: полное руководство, М.: ООО «И.Д. Вильямс», 2011

4. MSDN [Электронный ресурс]. – Сеть разработчиков Microsoft. – Режим доступа. : http://msdn.ru .