Частное учреждение образования

«Колледж бизнеса и права»

|  |
| --- |
| УТВЕРЖДАЮ  Заведующий  методическим кабинетом  \_\_\_\_\_\_\_\_\_\_\_\_\_Е.В.Фалей  « \_\_\_\_ » \_\_\_\_\_\_\_\_\_ 2016 |

|  |  |
| --- | --- |
| Специальность: 2-40 01 01 «Программное обеспечение информационных технологий» | Учебная практика по программированию |
| Составлена в соответствии с учебной программой, утвержденной директором Колледжа бизнеса и права 31.05.2016 | |

**Практическая работа №2**

**Инструкционно-технологическая карта**

Тема: Общие сведения о платформе .NET Framework. Возможности среды CLR. Библиотека классов .NET Framework. Примеры по платформе .NET Framework.

Цель: Закрепить общие сведения о платформе .NET Framework, возможностях среды CLR, библиотеке классов .NET Framework. Провести анализ примеров кода по платформе .NET Framework

Время выполнения: 2 часа

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомиться с теоретическими сведениями.
2. Проанализировать работу примеров кода по платформе .NET Framework

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**Общие сведения о платформе .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.

Например, ASP.NET размещает среду выполнения и обеспечивает масштабируемую среду для управляемого кода на стороне сервера. ASP.NET работает непосредственно со средой выполнения, чтобы обеспечить выполнение приложений ASP.NET и веб-служб XML, обсуждаемых ниже в этом разделе.

Обозреватель Internet Explorer может служить примером неуправляемого приложения, размещающего среду выполнения (в виде расширений типов MIME). Размещение среды выполнения в обозревателе Internet Explorer позволяет внедрять управляемые компоненты или элементы управления Windows Forms в HTML-документы. Такое размещение среды делает возможным выполнение управляемого мобильного кода (аналогичного элементам управления Microsoft® ActiveX®), но с существенными преимуществами управляемого кода, такими как выполнение в условиях неполного доверия и изолированное хранение файлов.

На следующем рисунке демонстрируется взаимосвязь среды CLR и библиотеки классов с пользовательскими приложениями и всей системой.На рисунке также показано, как управляемый код работает в пределах более широкой архитектуры.

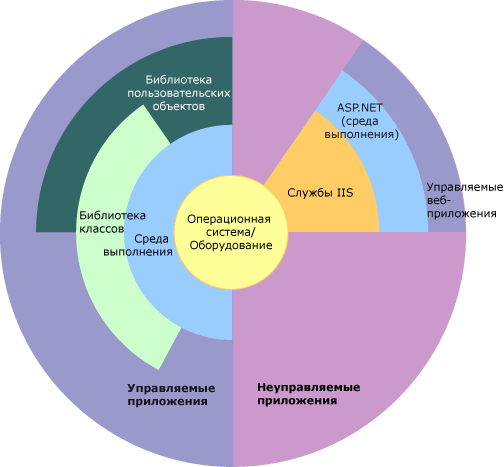


Рисунок 1 – .NET Framework в контексте

В следующих разделах приводится более подробное описание основных компонентов и средств платформы .NET Framework.

## **Возможности среды CLR**

Среда CLR управляет памятью, выполнением потоков, выполнением кода, проверкой безопасности кода, компиляцией и другими системными службами. Эти средства являются внутренними для управляемого кода, который выполняется в среде CLR.

По соображениям безопасности управляемым компонентам присваиваются разные степени доверия, зависящие от ряда факторов, в число которых входит их происхождение (например, Интернет, сеть предприятия или локальный компьютер). Это означает, что управляемый компонент может или не может выполнять операции доступа к файлам, операции доступа к реестру или другие важные функции, даже если он используется в одном активном приложении.

Среда выполнения обеспечивает управление доступом для кода. Например, пользователи могут доверить исполняемому приложению, внедренному в веб-страницу, воспроизведение анимации на экране или звукозаписи, не позволяя ему при этом получить доступ к личным данным, файловой системе или сети. Таким образом, средства безопасности CLR предоставляют подлинному развернутому в Интернете программному обеспечению исключительно богатые функции.

Среда выполнения также обеспечивает надежность кода, реализуя инфраструктуру строгой типизации и проверки кода, которую называют системой общих типов (CTS). Система общих типов обеспечивает самоописание всего управляемого кода. Различные языковые компиляторы корпорации Майкрософт и независимых изготовителей создают управляемый код, удовлетворяющий системе общих типов .Это означает, что управляемый код может принимать другие управляемые типы и экземпляры, при этом обеспечивая правильность типов и строгую типизацию.

Кроме того, управляемая среда выполнения исключает многие часто возникающие проблемы с программным обеспечением. Например, среда выполнения автоматически управляет размещением объектов и ссылками на объекты, освобождая их, когда они больше не используются. Автоматическое управление памятью исключает две наиболее часто возникающие ошибки приложений: утечки памяти и недействительные ссылки на память.

Среда выполнения также повышает продуктивность разработчиков. Например, программисты могут писать приложения на привычном языке разработки, при этом используя все преимущества среды выполнения, библиотеку классов и компоненты, написанные другими разработчиками на других языках. Это доступно любому производителю компиляторов, обращающихся к среде выполнения. Языковые компиляторы, предназначенные для платформы .NET Framework, делают средства .NET Framework доступными для существующего кода, написанного на соответствующих языках, существенно облегчая процесс переноса существующих приложений.

Хотя среда выполнения разрабатывалась для будущего программного обеспечения, она также поддерживает сегодняшнее и вчерашнее программное обеспечение. Взаимодействие управляемого и неуправляемого кодов позволяет разработчикам использовать необходимые компоненты COM и библиотеки DLL.

Среда выполнения разработана для повышения производительности. Хотя общеязыковая среда выполнения предоставляет многие стандартные службы времени выполнения, управляемый код никогда не интерпретируется. Средство компиляции по требованию (JIT) позволяет всему управляемому коду выполняться на машинном языке компьютера. Между тем диспетчер памяти устраняет возможность фрагментации памяти и увеличивает объем адресуемой памяти для дополнительного повышения производительности.

И наконец, среда выполнения может размещаться в высокопроизводительных серверных приложениях, таких как Microsoft® SQL Server™ и Internet Information Services (IIS). Такая инфраструктура позволяет использовать управляемый код для написания собственной логики программ, пользуясь при этом высочайшей производительностью лучших производственных серверов, которые поддерживают размещение среды выполнения.

## **Библиотека классов .NET Framework**

Библиотека классов платформы .NET Framework представляет собой коллекцию типов, которые тесно интегрируются со средой CLR. Библиотека классов является объектно-ориентированной; предоставляя типы, из которых управляемый код пользователя может наследовать функции. Это не только упрощает работу с типами .NET Framework, но также уменьшает время, затрачиваемое на изучение новых средств платформы .NET Framework. Кроме того, компоненты независимых производителей можно легко объединять с классами платформы .NET Framework.

Например, в классах коллекций .NET Framework реализуется набор интерфейсов, которые можно использовать для разработки пользовательских классов коллекций. Пользовательские классы коллекций будут без затруднений объединяться с классами .NET Framework.

Как и ожидается от объектно-ориентированной библиотеки классов, типы .NET Framework позволяют решать типовые задачи программирования, включая работу со строками, сбор данных, подключения к базам данных и доступ к файлам. В дополнение к этим обычным задачам библиотека классов содержит типы, поддерживающие многие специализированные сценарии разработки. Например, можно использовать платформу .NET Framework для разработки следующих типов приложений и служб:

* консольные приложения
* приложения с графическим интерфейсом пользователя Windows (Windows Forms);
* приложения Windows Presentation Foundation (WPF);
* приложения ASP.NET;
* веб-службы; См
* службы Windows;
* сервисноориентированные приложения с помощью Windows Communication Foundation (WCF);
* приложения поддерживающие бизнес-процессы Windows Workflow Foundation (WF).

Например, классы Windows Forms представляют собой полный набор типов, существенно упрощающих разработку графических интерфейсов пользователя Windows. При написании приложения форм ASP.NET можно использовать классы Web Forms.

# Примеры по платформе .NET Framework

Документация платформы .NET Framework содержит примеры кода, которые можно копировать прямо из разделов справки и вставлять в собственные проекты. Кроме того, длинные и более сложные примеры и образцы приложений можно загрузить из коллекции примеров кода MSDN. В настоящий момент для платформы .NET Framework доступны следующие примеры:

* [Примеры кода ASP.NET](https://msdn.microsoft.com/ru-ru/library/ee332498(v=vs.100).aspx) – примеры, которые демонстрируют использование функций ASP.NET (например, платформы динамических данных и платформы Model-View-Controller).
* [CLR Code Samples](http://go.microsoft.com/fwlink/?LinkId=201802) – примеры, которые демонстрируют такие функции, как развертывание, шифрование, Managed Extensibility Framework (MEF), а также форматирование даты и времени или численных значений.
* [Образцы программирования сети](https://msdn.microsoft.com/ru-ru/library/ee890485(v=vs.100).aspx) – Образцы, демонстрирующие функции сетевых технологий, кроме ASP.NET, которые доступны в среды CLR.
* [WPF Documentation Samples](http://go.microsoft.com/fwlink/?LinkID=159044) — примеры, демонстрирующие функции Windows Presentation Foundation (WPF).
* [Windows Communication Foundation (WCF) Samples](https://msdn.microsoft.com/ru-ru/library/dd483346(v=vs.100).aspx) — Примеры, демонстрирующие функции WCF и сценарии использования.
* [Windows Workflow (WF) Samples](https://msdn.microsoft.com/ru-ru/library/dd483375(v=vs.100).aspx) — примеры, демонстрирующие функции и сценарии использования WF.
* [Коллекция примеров кода MSDN](http://go.microsoft.com/fwlink/?LinkID=211702) — репозиторий примеров от компании Microsoft и сообщества разработчиков. Для поиска примеров, относящихся к конкретным продуктам или технологиям, можно использовать поиск по всему репозиторию или по предусмотренным категориям.

ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

1. Дейтел, П. Как программировать на Visual C# 2012 / П. Дейтел, Х. Дейтел. – 5-е изд. – СПб.: Питер, 2014. – 864 с.: ил.
2. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5 / Э. Троелсен. – 6-е изд. – М.: ООО «И.Д. Вильямс», 2013. – 1312 с.: ил.
3. Шарп Джон. Microsoft Visual C#. Подробное руководство / Джон Шарп. – 8-е изд. – СПб.: Питер, 2017. – 848 с.
4. Загружаемые файлы с примерами кода ASP.NET [Электронный ресурс] / Microsoft. – Microsoft, 2016. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/ee332498(v=vs.100).aspx. – Дата доступа: 10.08.2016.

# Примеры Windows Presentation Foundation [Электронный ресурс] / Microsoft. – Microsoft, 2016. – Режим доступа: https://msdn.microsoft.com/ru-RU/Library/ms771633. – Дата доступа: 10.08.2016.

1. Примеры кода от разработчиков [Электронный ресурс] / Microsoft. – Microsoft, 2016. – Режим доступа: code.msdn.microsoft.com. – Дата доступа: 20.08.2016.

Преподаватель Е.В. Багласова

|  |
| --- |
| Рассмотрено на заседании цикловой  комиссии программного обеспечения информационных технологий № 10  Протокол №\_\_\_\_\_от «\_\_\_»\_\_\_\_2016  Председатель ЦК *( )* Т.Г.Багласова |