# Глава 1
# Разработка приложений с использованием .NET Framework

# Скотт Гатри

**Скотт Гатри (род. 1975) является вице-президентом в Microsoft Developer Division. Он руководит командой разработчиков, которые строят ASP.NET , Common Language Runtime (CLR), Core. NET Base Class Library , Silverlight , Windows Forms , WPF , Internet Information Services 7.5, Commerce Server, . NET Compact Framework , Visual Web Developer и Visual Studio Инструменты для WPF.**

# Андерс Хейлсберг

**Андерс Хейлсберг (дат. Anders Hejlsberg; род. в декабре 1960, Копенгаген) — датский инженер-программист. В 1980 году написал свой первый компилятор языка Паскаль, который после портирования под операционную систему MS-DOS продал фирме Borland. До 1996 года Хейлсберг был главным инженером фирмы Borland, где создал новое поколение компиляторов Паскаля — язык Delphi, компилятор которого работал уже под операционной системой Windows. Позже возглавил группу по созданию и проектированию языка C#.В 2000 году Андерс Хейлсберг получил награду популярного журнала Dr. Dobb's Journal за создание Turbo Pascal, Delphi и C#. Он является одним из отцов-основателей .NET Framework.**

# Среда разработки

*Среда разработки* или *оболочка* - это совокупность средств, с помощью которых пишут, корректируют, преобразуют в машинные коды, отлаживают и запускают программы.

## Содержание среды разработки

Среда разработки содержит:

- текстовый редактор, предназначенный для ввода и корректировки текста программы;
- компилятор, предназначенный для перевода исходного текста в код, понятный компьютеру;
- средства отладки и запуска программ;
- общие библиотеки, содержащие часто используемые элементы программ;
- справочную систему и другие элементы.

# Основные задачи

Важнейшими задачами при создании программ являются:

- безопасность — невозможность несанкционированных действий;
- надежность — способность выполнять необходимые функции в предопределенных условиях;
- переносимость — возможность выполнения на различных типах компьютеров;
- использование готовых компонентов — для ускорения разработки;
- межъязыковое взаимодействие — возможность применять одновременно несколько языков программирования.

# Платформа .NET

Платформа .NET – это нечто большее, чем среда разработки для одного языка, это новая модель для создания приложений под Windows (в будущем, предполагается, и под другими ОС). Включает не только среду разработки для нескольких языков программирования, называемую Visual Studio.NET, но и множество других средств, например, механизмы поддержки баз данных, электронной почты и коммерции.

# Основные возможности .NET

**Платформа.NET**

Windows (а в будущем, предполагается, и под другими ОС).

Основные возможности .NET:

- полные возможности взаимодействия с существующим кодом. Например, существующие двоичные компоненты COM отлично работают вместе с двоичными файлами .NET;
- полное и абсолютное межъязыковое взаимодействие.
- общая среда выполнения для любых приложений .NET, вне зависимости от того, на каких языках они были созданы. Один из важных моментов при этом – то, что для всех языков используется один и тот же набор встроенных типов данных;
- библиотека базовых классов.
- упрощение процесса развертывания приложений. В .NET нет необходимости регистрировать двойные типы в системном реестре.

# Основные компоненты .NET

## Библиотека базовых классов (The Base Class Library)

| | | | |
|---|---|---|---|
| Доступ к данным | GUI | Безопасность | XML/SOAP |
| Управление потоками | Файловый ввод-вывод | Отладка | Прочее |

(SOAP – Simple Object Access Protocol - простой протокол доступа к объектам)
(XML- eXtensible Markup Language- один из языков для создания Web-страниц)
(GUI - Graphical user interface - разновидность пользовательского интерфейса)

## Стандартная среда выполнения для языков (The Common Language Runtime, CLR)

| | |
|---|---|
| Стандартная система типов (Common Type System, CTS) | Система правил работы с типами (Common Language Specification, CLS) |

# Основные компоненты .NET

Среда выполнения .NET обеспечивается с помощью Common Language Runtime (CLR, стандартная среда выполнения для языков). Главная роль CLR заключается в том, чтобы обнаруживать и загружать типы .NET и производить управление ими в соответствии с командами программиста.

Еще один строительный блок платформы .NET – это Common Type System (CTS стандартная система типов). CTS полностью описывает все типы данных, поддерживаемые средой выполнения, определяет, как одни типы данных могут взаимодействовать с другими.

***Common Language Specification (CLS)***
правил, определяющих подмножество общих типов данных, в отношении которых гарантируется, что они безопасны при использовании во всех языках .NET.

Платформа .NET предоставляет в распоряжение программиста библиотеку базовых классов, доступную из любого языка программирования .NET. Эта библиотека не только прячет обычные низкоуровневые операции, такие как файловый ввод-вывод, обработка графики и взаимодействие с оборудованием компьютера, но и обеспечивает поддержку большого количества служб, используемых в современных приложениях.

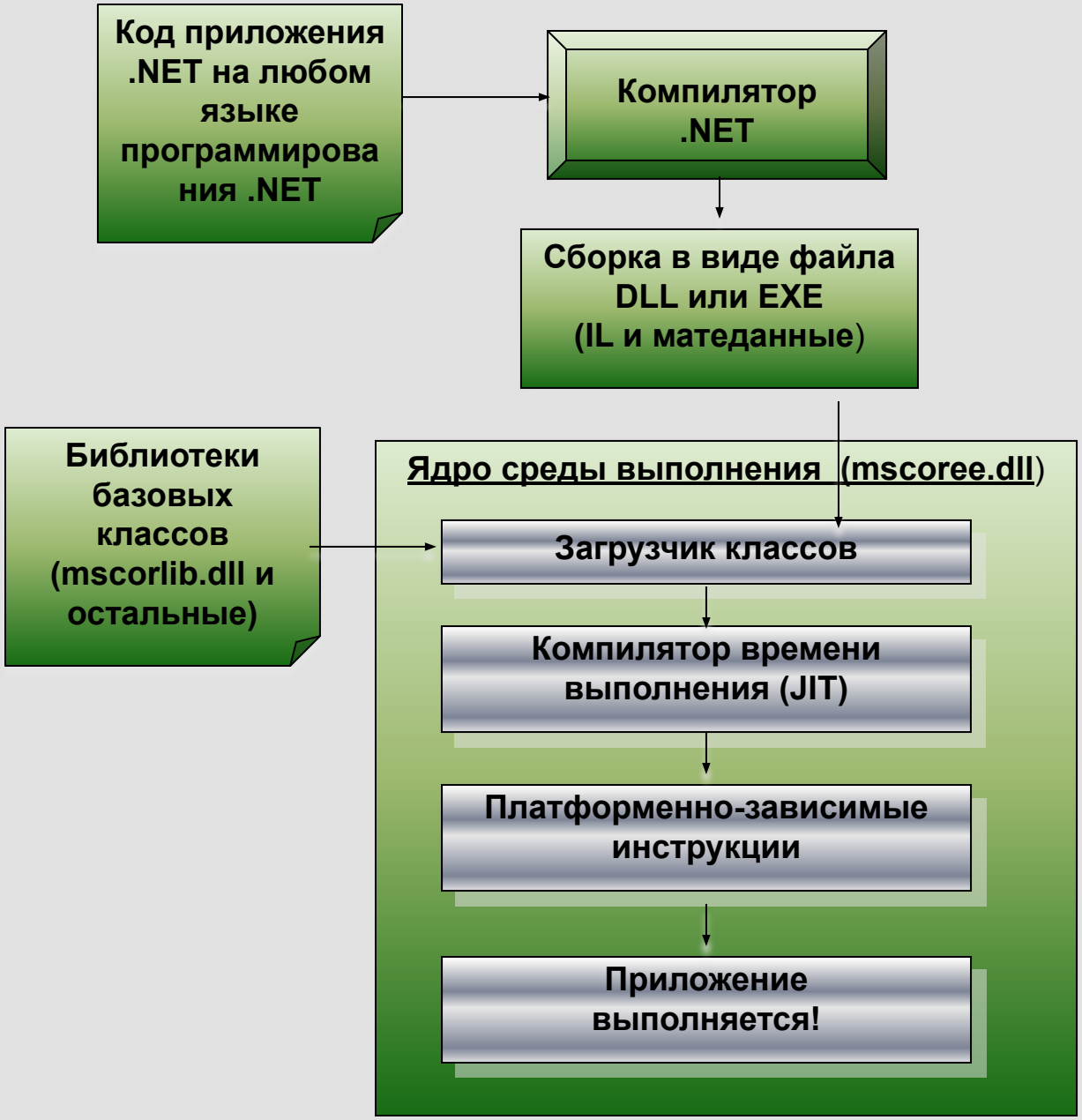
# Обзор двоичных файлов .NET (сборки)

Специально для платформы .NET Microsoft был разработан новый язык программирования C#, который генерирует код, предназначенный для выполнения только в среде выполнения .NET.

***Управляемый код*** *(managed code)* - это Microsoft код, предназначенный для работы в среде выполнения .NET.

***Сборка*** *(assembly)* – это двоичный файл, который содержит управляемый файл. Имеет расширение .exe и .dll. ***Содержимое двоичных файлов*** – это платформенно-независимый «промежуточный язык», который называется Microsoft Intermediate Language (MSIL или IL).

MSIL позволяет создавать приложения на ***любом*** языке, которые будут работать на *любой* платформе и под ***любой*** операционной системой. Поскольку в сборках содержится платформенно-независимый код IL, а выполняются в конечном итоге именно платформенно-зависимые инструкции, появляется необходимость в компиляторе времени выполнения –«just-in-time compiler» (JIT). Откомпилированные из IL платформенно-зависимые инструкции JIT помещаются в кэш-память, что очень сильно ускоряет работу приложения.


Код приложения .NET на любом языке программирования .NET
Компилятор .NET
Сборка в виде файла DLL или EXE (IL и матеданные)
Библиотеки базовых классов (mscorlib.dll и остальные)
Ядро среды выполнения (mscoree.dll)
Загрузчик классов
Компилятор времени выполнения (JIT)
Платформенно-зависимые инструкции
Приложение выполняется!

# Пространства имен .NET

**Сборка может содержать любое количество самых разных типов.**

***Тип*** **– это общий термин, который может относиться к классам, структурам, интерфейсам, перечислениям и прочему.**

**При создании приложения .NET требуется организовывать взаимодействие этих типов. При этом имеется возможность при создании собственных типов использовать пространства имен.**

***Пространство имен*** **– это логическая структура для организации имен, используемых в приложении .NET. Основное назначение пространств имен – предупредить возможные конфликты между именами в разных сборках. Определить использование в программе пространства имен можно с помощью служебного слова using.**

**Таким образом, пространство имен – это просто способ организации типов в единую группу.**

| Пространство имен .NET | Назначение |
|---|---|
| `System` | Внутри – множество низкоуровневых классов для работы с простыми типами, выполнения математических операций, сборки мусора и т.п. |
| `System.Collections` | Работа с контейнерами (массивы, очереди, списки и т.п.) |
| `System.Data` | Работа с базами данных |
| `System.Drawing`<br>`System.Drawing.Drawing2D`<br>`System.Drawing.Printing` | Типы для растровых изображений, шрифтов, значков, поддержки печати, а также специальные классы для вывода более сложных изображений |
| `System.IO` | Типы, отвечающие за операции ввода-вывода – в файл, буфер и т.п. |
| `System.Web` | Классы, используемые в web_приложениях, включая ASP.NET |
| `System.Windows.Forms` | Классы для работы с элементами интерфейса Windows – окнами, элементами управления и прочими |

# Объектно-ориентированное программирование

***Объект*** - совокупность данных, характеризующих его состояние, и функций их обработки, моделирующих его поведение.

Элементы структуры объекта скрыты и недоступны для непреднамеренного использования. Поэтому использование объекта происходит через его интерфейс.

***Интерфейс*** – это совокупность правил доступа.

***Инкапсуляция*** – это механизм скрытия внутренних элементов программы с целью защиты от внешнего вмешательства или неправильного использования.

В объектно-ориентированном программировании код и данные могут быть объединены вместе. Когда коды и данные объединяются таким способом, создаётся объект (object). Другими словами, объект - это то, что поддерживает инкапсуляцию.

***Полиморфизм*** - это свойство, которое позволяет одно и то же имя использовать для решения схожих, но технически разных задач. Целью полиморфизма является использование одного имени для задания общих для класса действий. Основная идея полиморфизма "один интерфейс, множество методов". Это означает, что можно создать общий интерфейс для группы близких по смыслу действий.

***Наследование*** - это процесс создания новых классов, посредством которого один объект может приобретать свойства другого. Точнее, класс – наследник может наследовать основные свойства класса - предка и добавлять к ним черты, характерные только для него. Наследование является важным, поскольку оно позволяет поддерживать концепцию иерархии классов. Применение иерархии классов делает управляемыми большие потоки информации.

***Visual Studio.NET*** – среда разработки, предоставляющая мощные и удобные средства написания, корректировки, компиляции, отладки и запуска приложений, использующих .NET-совместимые языки.

Приложение в процессе разработки называется проектом. Среда Visual Studio.NET позволяет создавать проекты различных типов, например:

- ***консольное приложение*** — выполняет вывод «на консоль», то есть в окно командного процессора;
- ***Windows*** — приложение использует элементы интерфейса Windows, включая формы, кнопки, флажки и пр.;
- ***веб-приложение*** — это приложение, доступ к которому выполняется через браузер (например, Internet Explorer) и которое по запросу формирует веб-страницу и отправляет ее клиенту по сети;
- ***веб-сервис*** — компонент, методы которого могут вызываться через Интернет;
- ***библиотека классов*** — объединяет классы, которые предназначены для использования в других приложениях.


Microsoft
.net