**🧩 Слайд 1. Кроссплатформенность .NET 6.0+**

**Текст для выступления:**  
Платформа .NET 6.0 — это универсальная среда разработки от Microsoft, которая позволяет создавать приложения для различных операционных систем. Основная особенность этой версии — полная кроссплатформенность и объединение всех веток .NET в единую экосистему. Теперь один код можно запускать на Windows, Linux, macOS, Android и iOS.  
.NET 6 стал частью стратегии Microsoft по созданию единой, быстрой и открытой платформы для всех устройств и типов приложений.

**💡 Слайд 2. Что такое .NET**

**Расширенный текст:**  
.NET — это открытая платформа разработки, включающая язык программирования (например, C#), библиотеки, компилятор и виртуальную машину для выполнения кода.  
С помощью .NET можно создавать:

* веб-приложения (ASP.NET Core),
* мобильные приложения (MAUI),
* десктопные программы (WPF, WinUI),
* облачные и IoT-решения.

.NET сочетает мощь объектно-ориентированного подхода с удобством и безопасностью среды исполнения (CLR), что делает её популярной в образовании и промышленности.

**🌍 Слайд 3. Понятие кроссплатформенности**

**Расширенный текст:**  
Кроссплатформенность — это способность программы работать на разных операционных системах без изменения исходного кода.  
.NET 6.0 реализует это за счёт:

* единого **Common Language Runtime (CLR)**, адаптированного под каждую ОС,
* общей **Base Class Library (BCL)**,
* системы сборки и публикации, создающей исполняемые файлы под нужную платформу.

Благодаря этому один проект может работать на Windows, Linux, macOS, а также на мобильных устройствах Android и iOS. Это экономит время и упрощает поддержку кода.

**⚙️ Слайд 4. Эволюция .NET**

**Расширенный текст:**  
Платформа прошла большой путь:

1. **.NET Framework (2002–2019)** — предназначен только для Windows.
2. **.NET Core (2016)** — стал кроссплатформенным и открытым.
3. **.NET 5 (2020)** — попытка объединить все варианты .NET.
4. **.NET 6 (2021)** — завершённое объединение, единый SDK, долгосрочная поддержка (LTS).

.NET 6 объединяет преимущества предыдущих поколений: совместимость, скорость, простоту и открытость.

**🧱 Слайд 5. Архитектура и инструменты .NET 6**

**Расширенный текст:**  
Основные компоненты .NET 6:

* **CLR (Common Language Runtime)** — выполняет код, управляет памятью и безопасностью.
* **BCL (Base Class Library)** — стандартная библиотека классов (работа с файлами, сетью, коллекциями и т.д.).
* **SDK и CLI (Command Line Interface)** — инструменты для сборки и запуска приложений.
* **MAUI (Multi-platform App UI)** — создание кроссплатформенных приложений с одним интерфейсом.
* **Blazor** — позволяет писать веб-приложения на C#, без JavaScript.

Эта архитектура позволяет разработчику сосредоточиться на логике программы, а не на особенностях конкретной ОС.

**💻 Слайд 6. Примеры кроссплатформенных решений**

**Расширенный текст:**

1. **ASP.NET Core Web API** — серверное приложение, которое можно запустить и на Windows Server, и на Linux.
2. **.NET MAUI** — одно приложение, одинаково работающее на Android, iOS и Windows с общим кодом интерфейса.
3. **Blazor WebAssembly** — C#-код, исполняемый прямо в браузере без серверной зависимости.
4. **Консольные и облачные сервисы** — .NET-приложения легко контейнеризируются и разворачиваются в Docker и Kubernetes.

Таким образом, .NET 6 подходит для любых типов устройств и сценариев.

**⚖️ Слайд 7. Преимущества и сложности**

**Расширенный текст:**  
**Преимущества:**

* Универсальность — один SDK и язык для всех платформ.
* Высокая производительность благодаря JIT и AOT-компиляции.
* Поддержка контейнеров, микросервисов и облаков.
* Открытый исходный код и активное сообщество.

**Сложности:**

* Некоторые старые библиотеки работают только под Windows.
* MAUI пока не полностью стабилен на всех ОС.
* Необходимость настройки окружения при кросс-компиляции.

**🔮 Слайд 8. Будущее .NET**

**Расширенный текст:**  
Microsoft продолжает развивать .NET — уже вышли .NET 8 и 9, которые усилили кроссплатформенность и производительность.  
Основные направления развития:

* Улучшение поддержки ARM64 и мобильных устройств.
* Интеграция с искусственным интеллектом (ML.NET, ONNX).
* Совершенствование MAUI и Blazor Hybrid.
* Оптимизация сборки и уменьшение размера приложений.

.NET становится действительно универсальной платформой для всех типов разработчиков — от студентов до крупных компаний.