Язык C# появился на свет в июне 2000 г. в результате кропотливой работы большой группы разработчиков компании Microsoft, возглавляемой Андерсом Хейлсбергом (Anders Hejlsberg). Этот человек известен как автор одного из первых компилируемых языков программирования для персональных компьютеров IBM -- Turbo Pascal. Наверное, на территории бывшего Советского Союза многие разработчики со стажем, да и просто люди, обучавшиеся в той или иной форме программированию в вузах, испытали на себе очарование и удобство использования этого продукта. Кроме того, во время работы в корпорации Borland Андерс Хейлсберг прославился созданием интегрированной среды Delphi (он руководил этим проектом вплоть до выхода версии 4.0).

Появление языка C# и инициативы .NET отнюдь не случайно пришлось на начало лета 2000 г. Именно к этому моменту компания Microsoft подготовила промышленные версии новых компонентных технологий и решений в области обмена сообщениями и данными, а также создания Internet-приложений (COM+, ASP+, ADO+, SOAP, Biztalk Framework). Несомненно, лучшим способом продвижения этих новинок является создание инструментария для разработчиков с их полноценной поддержкой. В этом и заключается одна из главных задач нового языка C#. Кроме того Microsoft не могла больше расширять все те же инструменты и языки разработки, делая их все более и более сложными для удовлетворения конфликтующих между собой требований поддержки современного оборудования и обеспечения обратной совместимости с теми продуктами, которые были созданы в начале 1990-х гг. во время первого появления Windows. Наступает момент, когда необходимо начать с чистого листа для того, чтобы создать простой, но имеющий сложную структуру набор языков, сред и средств разработки, которые позволят разработчику легко создавать современные программные продукты.

C# и .NET являются той самой отправной точкой. Если говорить упрощенно, то .NET представляет собой новую платформу, новый API для программирования в Windows, а C# е новый язык, созданный с нуля, для работы с этой платформой, а также для извлечения всех выгод из прогресса сред разработки и нашего понимания принципов объектно-ориентированного программирования в течение последних 20 лет.

Необходимо отметить, что обратная совместимость не потеряна. Существующие программы будут выполняться, а платформа .NET была спроектирована таким образом, чтобы она могла работать с имеющимся программным обеспечением. Связь между компонентами в Windows сейчас почти целиком осуществляется при помощи COM. С учетом этого .NET обладает способностью (а) создавать оболочки (wrappers) вокруг существующих компонентов COM, так что компоненты .NET могут общаться с ними, и (б) создавать оболочки вокруг компонентов .NET, что позволяет им выглядеть как обычные COM-компоненты.

Авторы C# стремились создать язык, сочетающий простоту и выразительность современных объектно-ориентированных языков (вроде Java) с богатством возможностей и мощью C++. По словам Андерса Хейлсберга, C# позаимствовал большинство своих синтаксических конструкций из C++. В частности, в нем присутствуют такие удобные типы данных, как структуры и перечисления (другой потомок C++ -- Java -- лишен этих элементов, что создает определенные неудобства при программировании). Синтаксические конструкции C# унаследованы не только от C++, но и от Visual Basic. Например, в C#, как и в Visual Basic, используются свойства классов. Как C++, C# позволяет производить перегрузку операторов для созданных вами типов Java не поддерживает ни ту, ни другую возможность). C# — это фактически гибрид разных языков. При этом C# синтаксически не менее (если не более) чист, чем Java, так же прост, как Visual Basic, и обладает практически той же мощью и гибкостью, что и C++.

Особенности C#:

- Полный и хорошо определенный набор основных типов.
- Встроенная поддержка автоматической генерации XML-документации. Автоматическое

освобождение динамически распределенной памяти.
- Возможность отметки классов и методов атрибутами, определяемыми пользователем. Это может быть полезно при документировании и способно воздействовать на процесс компиляции (например, можно пометить методы, которые должны компилироваться только в отладочном режиме).
- Полный доступ к библиотеке базовых классов .NET, а также легкий доступ к Windows API (если это действительно необходимо).
- Указатели и прямой доступ к памяти, если они необходимы. Однако язык разработан таким образом, что практически во всех случаях можно обойтись и без этого.
- Поддержка свойств и событий в стиле VB.
- Простое изменение ключей компиляции. Позволяет получать исполняемые файлы или библиотеки компонентов .NET, которые могут быть вызваны другим кодом так же, как элементы управления ActiveX (компоненты COM).
- Возможность использования C# для написания динамических web-страниц ASP.NET.

Одной из областей, для которых не предназначен этот язык, являются критичные по времени и высокопроизводительные программы, когда имеет значение, занимать исполнение цикла 1000 или 1050 машинных циклов, и освобождать ресурсы требуется немедленно. C++ остается в этой области наилучшим из языков низкого уровня. В C# отсутствуют некоторые ключевые моменты, необходимые для создания высокопроизводительных приложений, в частности подставляемые функции и деструкторы, выполнение которых гарантируется в определенных точках кода.