*Жильцов А.А.*

**Платформа .Net 8**

.NET — это бесплатная  платформа разработчика с открытым кодом для создания программных приложений. Платформа .NET основана на высокопроизводительной среде выполнения, которая используется в рабочей среде многими высокомасштабируемыми приложениями.

Приложения и библиотеки .NET создаются из исходного кода и файла проекта с помощью .NET CLI или интегрированной среды разработки (IDE), например, Visual Studio, JetBrains Rider, которые широко используются при обучении программированию студентов СПО.

В ноябре 2023 года вышел релиз открытой платформы .NET 8, которая объединяет в себе продукты .NET Framework, .NET Core и Mono. С помощью .NET 8 можно разрабатывать кроссплатформенные приложения для браузера, облака, десктопа, IoT-устройств и мобильных платформ, используя общие библиотеки и единый процесс сборки, не зависимый от типа приложения. Для Lunux, macOS и Windows сформированы сборки .NET SDK 8, .NET Runtime 8 и ASP.NET Core Runtime 8. .NET Desktop Runtime 8 доступен только для Windows. Версия .NET 8 относится к категории LTS и будет поддерживаться до ноября 2026 года. Связанные с проектом наработки распространяются под лицензией MIT.

В состав .NET 8 входят Runtime с JIT-компилятором RyuJIT, API спецификации, библиотеки WPF, Windows Forms, WinUI, Entity Framework, командный интерфейс dotnet, а также инструменты для создания различных типов приложений. Отдельно выделены стеки для разработки web-приложений ASP.NET Core 8.0, ORM-слой Entity Framework Core 8.0, библиотека WPF 8, фреймворк Windows Forms 8, платформа Aspire 8 для создания приложений Cloud Native, фреймворк MAUI 8 для создания кроссплатформенных интерфейсов пользователя, а также релизы языков C# 12 и F# 8. Поддержка .NET 8.0 и C# 12 включена в свободный редактор кода Visual Studio Code.

Основные новшества платформы:

* Улучшена производительность работы за счёт нового генератора кода, поддержки инструкций AVX-512 и нового интерфейса для работы с данными в формате UTF-8.
* Добавлен новый стек .NET Aspire, предназначенный для разработки облачных приложений (cloud-native), используя технологии .NET. В .NET Aspire предоставляется большой набор компонентов, которые могут потребоваться при создании приложений, запускаемых в облачных окружения, включая средства для сбора телеметрии, настройки, проверки работоспособности и обеспечения отказоустойчивости.
* Улучшены инструменты для упаковки приложений в изолированные контейнеры. Уменьшен размер базового образа контейнера с компонентами платформы .NET. Усовершенствована компиляция в самодостаточные исполняемый файлы (Native AOT), в которых весь проект изначально скомпилирован в машинный код целевой платформы.
* Расширены возможности программных интерфейсов, которые можно использовать в системах машинного обучения. Добавлены примеры для создания чат-ботов, генерации контента и обращения к сервисам Azure AI.
* В Blazor уменьшено время загрузки страниц, увеличена масштабируемость, разрешено использование в одном приложении Blazor Server и Blazor WebAssembly. Добавлен Jiterpreter, новый более быстрый WebAssembly runtimе. Улучшены возможности аутентификации и авторизации.
* В Entity Framework Core предоставлены новые сложные типы, которые можно использовать как объекты. Добавлена поддержка иерархических данных в SQL Server.
* В пакетном менеджере NuGet упрощён аудит пакетов в проектах и обеспечено информирование о наличии известных уязвимостей.
* В .NET Runtime добавлен режим упреждающей компиляции (AOT) для платформ WebAssembly (WASM) и Android.
* В WPF добавлен диалог открытия каталога (OpenFolderDialog) и включена поддержка аппаратного ускорения в RDP.
* Значительно улучшена поддержка архитектуры ARM64 и повышено качество кода для ARM.
* В отладчике обеспечен показ сводных отладочных данных и реализованы упрощённые отладочные прослойки для обычно используемых типов .NET.
* Обеспечена интеграция больших языковых моделей (LLMs). Разработчики теперь смогут интегрировать LLMs в свои приложения, такие как, например, GPT4. Это позволит разработчикам привнести в свои приложения искусственный интеллект (AI) и значительно повысить их производительность, а также удобство использования.
* Произведены улучшения в сериализации System.Text.Json. System.Text.Json - это встроенная .NET библиотека, предоставляющая функциональность сериализации и десериализации JSON. Она позволяет разработчикам конвертировать .NET объекты, преобразуя их в данные JSON и наоборот.

Прогресс и темпы разработки программного обеспечения продолжают расти, при этом такая платформа, как .NET 8 позволяет повысить эффективность и качество разработки программных приложений. Целесообразно также применять возможности Net 8 в образовательном процессе при обучении студентов СПО программированию.