

VAWI OBJC WS 23/24 Block B

# Hausarbeit

Aufgabe 3

von: Schubert, Patrick

# 1 Grundlagen .NET

".NET ist eine kostenlose, plattformübergreifende und frei verfügbare Entwicklungsumgebung, mit der unterschiedlichste Applikationen erstellt werden können [2]. Microsoft pflegt dabei verschiedene Varianten von .NET. Die wichtigsten zwei Varianten sind einerseits der Vorgänger, das .NET-Framework (dt. Software-Bibliothek, wörtlich Rahmenwerk), und andererseits ein moderneres .NET Core. Letzteres wird oft kurz als .Net bezeichnet und stellt unter anderem auch cloudbasierte Applikationen durch Microsoft Azure zur Verfügung. Das .NET-Framework wird lediglich durch das Betriebssystem Windows unterstützt, während .NET Core plattformunabhängig ist [6]. Beide Versionen sind voll kompatibel miteinander [2].

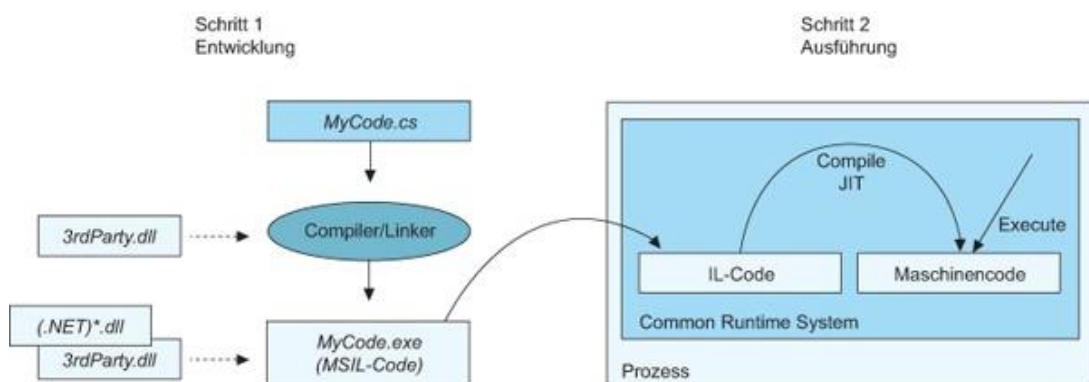
Grundlegend werden mit dem Begriff Framework Software-Artefakte assoziiert, die zur Verfügung gestellt werden und bestimmte Funktionalitäten mitbringen. Wenn sie in eigenen Softwareprojekten eingesetzt werden, können sie dabei helfen, bestimmte Probleme zu lösen [8].

Wesentliche Vorteile von .NET Core bietet unter anderem ein für die Webentwicklung entwickeltes Framework namens ASP.NET Core. Es sorgt beispielsweise für eine einheitliche Umgebung, in der Webbenutzeroberflächen generiert werden können. Außerdem kann im Browser ausschließlich mit C# gearbeitet werden, ohne eine weitere Programmiersprache wie zum Beispiel JavaScript erlernen zu müssen. Zudem können damit webfähige, programmierbare Schnittstellen (sog. APIs) erstellt werden [6]. Allerdings soll im Folgenden die Webentwicklung nicht weiter fokussiert werden.

Durch das .NET-Framework wurde .NET ursprünglich für das Betriebssystem Windows entwickelt. Mit .NET Core wurde eine cloudbasierte und plattformübergreifende Entwicklungsumgebung geschaffen, die Nutzer unter Betriebssystemen wie Linux oder MacOS unterstützt [7]. Der Quellcode liegt dabei offen in mehreren GitHub-Containern und wird von einer regen Community stets weiterentwickelt [2]. Es können beispielsweise clientseitige-, serverseitige-, mobile- oder cloudfähige Applikationen entwickelt werden [3]. Die erste marktfähige Version von .NET-Framework 1.0 entstand im Jahr 2002 [9]. Heute, im April 2024 mit dem Betriebssystem Windows 11, liegt das .NET-Framework in der Version 4.8.1 vor, während .NET Core aktuell in der Version 9 vorliegt [4]. Jede dieser Versionen enthält eine sogenannte Common

Language Runtime (CLR) sowie eine Base Class Library (BCL), also eine Basisklassenbibliothek, die grundlegende Funktionalitäten zur Verfügung stellt. Weitere verwaltete Bibliotheken sind darin ebenfalls enthalten [4]. Informationen zu den einzelnen Versionen von .NET können zum Beispiel in [9] oder [4] nachgelesen werden.

Die CLR bildet eine sogenannte virtuelle Maschine (VM), mit deren Hilfe .NET-Programme übersetzt und ausgeführt werden [9]. Grundsätzlich ist eine VM eine Software, die eine Hardware abbildet und damit eine gekapselte Umgebung schafft [11]. Durch diese immer gleiche Umgebung funktionieren Programme auf verschiedenen Betriebssystemen mit unterschiedlicher Hardware (32 Bit, 64 Bit) [9]. Schließlich wird der Code nicht direkt auf einem Prozessor ausgeführt. In einem ersten Schritt wird ein prozessorunabhängiger Code (das Programm) mittels einer Entwicklungsumgebung wie zum Beispiel Visual Studio Code erstellt. Dieser unabhängige Code wird IL-Code bzw. Intermediate Language Code oder auch MSIL-Code (Microsoft Intermediate Language Code) genannt. Dieser IL-Code wird von einem .NET-Compiler in eine .exe- oder .dll-Datei kompiliert und nach der Erstellung auf der Maschine des Benutzers installiert [9]. Die gespeicherte .exe- oder .dll-Datei wird .NET-Assembly genannt [5]. Allgemein bezeichnen Assemblies bereitgestellte bzw. übersetzte Programmklassen, vergleichbar mit JAR-Dateien in der Programmiersprache Java [10]. Welche Installationspfade dafür verwendet werden, kann unter anderem in [9] nachgelesen werden. Da der Code in ein Zwischenformat kompiliert wird, kann dieser unabhängig von Betriebssystemen (Windows, Linux etc.) auf unterschiedlicher Computerhardware (siehe oben) ausgeführt werden. Abbildung 12 soll diesen Zusammenhang und das zuvor Gesagte noch einmal verdeutlichen.



**Abb. 1: Prinzip der Erstellung und Ausführung eines .NET-Programms, vgl. [9].**

Wie Abbildung 12 zeigt, ist es Aufgabe der CLR, den IL-Code „Just In Time“ (JIT) in Maschinencode zu übersetzen. Dabei können mehr als 10.000 Zeilen pro Sekunde übersetzt werden. Der Code unter .NET wird in etwa genauso schnell ausgeführt wie ein C- oder C++-Programm. Insbesondere Systemabstürze werden reduziert, da die Runtime Applikationsfehler abfangen kann. Aus den vorangegangenen Ausführungen wird deutlich, dass ein Bruch mit den direktkompilierenden Programmiersprachen (insbesondere C++ und C) vorliegt. Damit wird einem Trend gefolgt, der durch den Erfolg von Java und anderen Skriptsprachen beeinflusst wurde [10].

Weiterhin sorgt die CLR dafür, dass das Programm in den Speicher geladen und verwaltet wird (Garbage Collector). Ebenso kontrolliert sie, welche Datentypen konvertiert werden müssen und ob dabei Ausnahmen zur Laufzeit auftreten [9]. Hierdurch wird deutlich, dass es sich bei .NET-Sprachen um ein statisches Typsystem handelt. Allerdings unterscheidet sich .Net von nativen Programmiersprachen dadurch, dass grundlegende Datentypen im Basissystem - dem sogenannten Common Type System (CTS) - und nicht in einer der Programmiersprachen verankert sind [9]. Standardmäßig werden von der Klasse System zum Beispiel System.Boolean, System.Char, System.Int16, System.Int32, System.Double usw. als Datentypen bereitgestellt. Dabei werden Werttypen (englisch: value type) und Referenztypen (englisch: reference type, deutsch auch als Verweistyp verwendet) unterschieden. Erstere haben eine feste Größe in Bytes und werden auf dem sogenannten Stack gespeichert, während letztere keine feste Größe in Bytes besitzen und auf dem sogenannten Heap gespeichert werden [9]. Allerdings soll im Folgenden nicht näher auf Datentypen eingegangen werden.

Da .NET plattformunabhängig ist, unterstützt Microsoft insgesamt die drei Programmiersprachen C#, F# und VB (Visual Basic). F# ist eine rein funktionsbasierte Sprache. VB ist eine der ältesten Sprachen aus dem Microsoft-Ökosystem. Die Syntax wird heute nicht mehr weiterentwickelt, aber Updates wirken sich dennoch positiv auf VB aus. C# ist eine plattformübergreifende und universelle Sprache. Sie ist sehr leistungsfähig, wird stetig weiterentwickelt und ist die beliebteste .NET-Sprache [5].

Angesichts des stetig wachsenden Kostendrucks und der intensiven Konkurrenz auf dem Markt suchen Unternehmen kontinuierlich nach effizienteren Lösungen, um Applikationen und Software schneller und kostengünstiger zu entwickeln. Für das Jahr 2024 zeichnen sich bestimmte Trends ab. Ein Beispiel hierfür ist die Möglichkeit, mithilfe von .NET MAUI - einem weiteren Framework von .NET - plattformübergreifend native mobile Apps für die Betriebssysteme Android und iOS mit nur einer Sprache, nämlich C#, zu entwickeln. Die Stärke von .NET MAUI liegt in der plattformübergreifenden Bereitstellung von UI-Komponenten, die wiederverwendet werden können, was zu erheblichen Zeitersparnissen in der Entwicklung führt. Darüber hinaus ist geplant, maschinelles Lernen nahtlos in das .NET-Framework durch das open-source Framework ML.NET zu integrieren. Diese Integration bietet Entwicklern die Möglichkeit, fundiertere datengetriebene Entscheidungen zu treffen [1]. Dennoch bleibt es abzuwarten, inwieweit sich .NET und C# in diesem Bereich etablieren werden, da Python und seine Bibliotheken für maschinelles Lernen derzeit unangefochten sind. Schließlich wird Microsoft auch zukünftig einen Fokus auf IT-Sicherheitsthemen legen, wie beispielsweise kryptografisches Hashing in Verbindung mit einem integrierten Identitätssystem, mit dem Ziel, die Sicherheit von Benutzerdaten zu verbessern [1]. Angesichts der aufgezeigten Entwicklungen steht die Softwareindustrie auch im Jahr 2024 vor neuen Herausforderungen.

## Literaturverzeichnis

- [1] Becker, C. (2024, 26. Februar). Den Blick in die Zukunft richten: .NET-Entwicklungstrends im Jahr 2024 | Infobest Systemhaus GmbH aus Leverkusen. Infobest Software Outsourcing Company. <https://www.infobest.de/den-blick-in-die-zukunft-richten-net-entwicklungstrends-im-jahr-2024/> [zuletzt abgerufen am: 02.04.2024].
- [2] Microsoft. „.NET Core-Einführung“. Microsoft Learn. URL: <https://learn.microsoft.com/de-de/dotnet/core/introduction> [zuletzt abgerufen am: 02.04.2024].
- [3] Microsoft. „Erstellen von Apps mit .NET“. URL: <https://learn.microsoft.com/de-de/dotnet/core/apps> [zuletzt abgerufen am: 02.04.2024].
- [4] Microsoft. „.NET Framework-Versionen und Abhängigkeiten“. URL: <https://learn.microsoft.com/de-de/dotnet/framework/migration-guide/versions-and-dependencies> [zuletzt abgerufen am: 02.04.2024].
- [5] Microsoft. „Was ist .NET?“. URL: <https://learn.microsoft.com/de-de/training/modules/dotnet-introduction/2-what-is-dotnet> [zuletzt abgerufen am: 02.04.2024].
- [6] Microsoft. „Übersicht über ASP.NET Core“ URL: <https://learn.microsoft.com/de-de/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-8.0> [zuletzt abgerufen am: 02.04.2024].
- [7] Price, M. J. (2023). C# 12 and .NET 8 - Modern Cross-Platform Development Fundamentals - Eighth Edition: Start Building Websites and Services with ASP.NET Core 8, Blazor, and EF Core 8.
- [8] Programmieren 2 Script.zip. (K. Pflug). Dropbox. URL: [https://www.dropbox.com/s/4z9se8u2iqgjo16/Programmieren%20%20Script.zip?dl=0&file\\_subpath=%2FProgrammieren+2+Script.pdf](https://www.dropbox.com/s/4z9se8u2iqgjo16/Programmieren%20%20Script.zip?dl=0&file_subpath=%2FProgrammieren+2+Script.pdf) [zuletzt abgerufen am: 02.04.2024].
- [9] Wenger, R. (2010). Handbuch der .NET 4.0-Programmierung: C# und .NET-Grundlagen. Microsoft Press Deutschland.

[10] Wikipedia. (2003, 15. April). „.Net-Framework“. URL:  
<https://de.wikipedia.org/wiki/.Net-Framework> [zuletzt abgerufen am:  
02.04.2024].

[11] Wikipedia (2002, 12. Juli). Virtuelle Maschine. URL:  
[https://de.wikipedia.org/wiki/Virtuelle\\_Maschine](https://de.wikipedia.org/wiki/Virtuelle_Maschine) [zuletzt abgerufen am:  
02.04.2024].