

# Visual Basic

Eine Einführung in die Programmierung

Michael Lappenbusch

FACHINFORMATIKER ANWENDUNGSENTWICKLUNG

## Inhaltsverzeichnis

Einführung in die Programmierung mit Visual Basic.....	2
1.1 Was ist Visual Basic?.....	2
1.2 Die Entwicklungsumgebung von Visual Basic.....	3
1.3 Erste Schritte mit Visual Basic .....	4
Grundlagen der Programmierung .....	5
2.1 Variablen und Datentypen .....	5
2.2 Verzweigungen und Schleifen .....	6
2.3 Funktionen und Prozeduren.....	7
Fortgeschrittene Techniken in Visual Basic.....	9
3.1 Steuerelemente und Formulare .....	9
3.2 Datenbankzugriff mit Visual Basic.....	10
3.3 Erstellung von Benutzerdefinierten Steuerelementen .....	11
Anwendungsentwicklung mit Visual Basic .....	12
4.1 Erstellung eines einfachen Programms.....	12
4.2 Erstellung einer Datenbankanwendung.....	13
4.3 Erstellung einer Web-Anwendung mit Visual Basic .....	14
Fehlerbehandlung und Debugging .....	16
5.1 Fehlerbehandlung mit Try-Catch-Finally .....	16
5.2 Debugging-Tools in Visual Basic .....	17
Erweiterungen und Add-Ons.....	18
6.1 Erweiterungen für die Entwicklungsumgebung .....	18
6.2 Add-Ons für Visual Basic.....	19
Impressum.....	21

# Einführung in die Programmierung mit Visual Basic

## 1.1 Was ist Visual Basic?

Visual Basic (VB) ist eine objektorientierte Programmiersprache und Entwicklungsumgebung, die von Microsoft entwickelt wurde. Es wurde erstmals 1991 veröffentlicht und ist seitdem eine der am häufigsten verwendeten Programmiersprachen für die Entwicklung von Windows-Anwendungen.

Visual Basic ermöglicht es Entwicklern, grafische Benutzeroberflächen (GUIs) für ihre Anwendungen zu erstellen, indem sie Steuerelemente wie Schaltflächen, Textfelder und Listboxen auf ein Formular ziehen und diese dann mit Code verknüpfen. Dieser Ansatz wird als visuelle Programmierung bezeichnet und erleichtert es Entwicklern, benutzerfreundliche Anwendungen zu erstellen, ohne dass tiefgreifende Kenntnisse der Programmierung erforderlich sind.

Visual Basic unterstützt die Entwicklung von Anwendungen auf der .NET Framework-Plattform von Microsoft und kann sowohl für Desktop-Anwendungen als auch für Web-Anwendungen verwendet werden. Es bietet auch eine breite Palette von Bibliotheken und Klassen, die Entwicklern das Erstellen von Anwendungen mit erweiterten Funktionen wie Datenbankzugriff und Netzwerkkommunikation erleichtern.

Im Laufe der Jahre hat Visual Basic mehrere Versionen erfahren, die neue Funktionen und Verbesserungen hinzufügten. Die aktuelle Version ist Visual Basic.NET (VB.NET) und wurde im Jahr 2002 veröffentlicht. Es unterscheidet sich von früheren Versionen von Visual Basic durch die Verwendung des .NET Frameworks, die Unterstützung von objektorientierten Konzepten und die Verwendung einer modularen, ereignisgesteuerten Programmierung.

## 1.2 Die Entwicklungsumgebung von Visual Basic

Die Entwicklungsumgebung von Visual Basic, auch als Visual Studio bekannt, ist eine integrierte Entwicklungsumgebung (IDE) von Microsoft, die Entwicklern das Erstellen, Debuggen und Bereitstellen von Anwendungen in Visual Basic erleichtert. Es bietet eine Vielzahl von Werkzeugen und Funktionen, die es Entwicklern ermöglichen, effizienter und produktiver zu arbeiten.

Einige der wichtigsten Bestandteile der Entwicklungsumgebung von Visual Basic sind:

**Projektmappen-Explorer:** Ein Werkzeug, mit dem Entwickler ihre Projekte und Dateien verwalten und organisieren können.

**Formulareditor:** Ein Werkzeug, mit dem Entwickler grafische Benutzeroberflächen für ihre Anwendungen erstellen können, indem sie Steuerelemente wie Schaltflächen, Textfelder und Listboxen auf ein Formular ziehen und diese dann mit Code verknüpfen.

**Code-Editor:** Ein Werkzeug, mit dem Entwickler ihren Code schreiben, bearbeiten und debuggen können. Es bietet Funktionen wie Syntaxhervorhebung, Code-Vervollständigung und automatisches Einrücken.

**Debugger:** Ein Werkzeug, mit dem Entwickler ihren Code ausführen und Fehler finden und beheben können. Es bietet Funktionen wie die Möglichkeit, die Ausführung des Codes zu unterbrechen, die Werte von Variablen anzuzeigen und die Ausführungsschritte zu verfolgen.

**Server-Explorer:** Ein Werkzeug, mit dem Entwickler auf Datenbanken, Webservices und andere Serverressourcen zugreifen können, um ihre Anwendungen mit Daten zu verbinden.

**Toolbox:** Eine Sammlung von Steuerelementen und Komponenten, die Entwicklern beim Erstellen ihrer Anwendungen zur Verfügung stehen.

**Lösungs-Explorer:** Eine Übersicht über alle Projekte und Dateien, die Teil einer Lösung sind.

Visual Studio gibt es in mehreren Editionen, darunter Community, Professional und Enterprise, die unterschiedliche Funktionen und Tools enthalten.

### 1.3 Erste Schritte mit Visual Basic

Erste Schritte mit Visual Basic können für Anfänger einschüchternd sein, aber mit ein wenig Anleitung und Übung kann man schnell ein grundlegendes Verständnis für die Sprache und die Entwicklungsumgebung erlangen.

Einer der ersten Schritte beim Erlernen von Visual Basic ist die Installation der Entwicklungsumgebung, Visual Studio. Es kann von der Microsoft-Website heruntergeladen und installiert werden. Nachdem die Installation abgeschlossen ist, kann man Visual Studio starten und ein neues Projekt erstellen.

Ein einfacher Weg, um die ersten Schritte in Visual Basic zu machen, ist das Erstellen einer einfachen Anwendung mit einer Schaltfläche und einem Textfeld. Dazu kann man ein neues Formular erstellen, indem man auf die Schaltfläche "Projekt hinzufügen" -> "Windows Form" klickt. Dann kann man die Toolbox öffnen und eine Schaltfläche und ein Textfeld auf das Formular ziehen.

Nachdem die Steuerelemente platziert sind, kann man mit dem Schreiben des Codes beginnen. Um eine Aktion auszuführen, wenn die Schaltfläche gedrückt wird, kann man den Code-Editor öffnen und das Ereignis "Click" der Schaltfläche auswählen. Dann kann man den Code schreiben, der ausgeführt werden soll, wenn die Schaltfläche gedrückt wird, z.B. das Textfeld mit einem bestimmten Text zu füllen.

Das ist ein sehr einfaches Beispiel, aber es zeigt, wie einfach es ist, eine Anwendung mit Visual Basic zu erstellen. Mit etwas Übung und weiterer Erforschung der Sprache und der Entwicklungsumgebung kann man schnell fortgeschrittene Techniken und Konzepte erlernen und komplexere Anwendungen erstellen.

# Grundlagen der Programmierung

## 2.1 Variablen und Datentypen

Variablen sind ein wichtiger Bestandteil jeder Programmiersprache, und Visual Basic ist keine Ausnahme. Eine Variable ist ein Behälter, in dem man einen Wert speichern kann. Der Wert einer Variablen kann im Laufe der Ausführung eines Programms geändert werden.

In Visual Basic gibt es verschiedene Datentypen, die für verschiedene Arten von Werten verwendet werden können. Einige der gängigsten Datentypen sind:

Integer (Integer): Ein ganzzahliger Wert ohne Nachkommastellen, z.B. -5, 0, 10.

Single (Single): Ein Gleitkommawert mit einer Genauigkeit von 7 Nachkommastellen, z.B. 3.14, -0.01.

Double (Double): Ein Gleitkommawert mit einer Genauigkeit von 15 Nachkommastellen, z.B. 3.14159265358979323846.

String (String): Eine Zeichenfolge, z.B. "Hallo Welt!".

Boolean (Boolean): Ein Wahrheitswert, entweder True oder False.

Eine Variable in Visual Basic wird deklariert, indem man den Datentyp und den Namen der Variable angibt, z.B:

```
Dim myInteger As Integer
```

```
Dim myString As String
```

Nachdem eine Variable deklariert wurde, kann man einen Wert zuweisen, indem man den Variablennamen und den Wert in einer Zuweisungsoperation verwendet, z.B:

```
myInteger = 5
```

```
myString = "Hallo Welt!"
```

Es ist auch möglich, Variablen zu deklarieren und gleichzeitig einen Wert zuzuweisen, z.B:

```
Dim myInteger As Integer = 5
```

```
Dim myString As String = "Hallo Welt!"
```

Es ist wichtig zu beachten, dass der Datentyp der Variablen bei der Deklaration festgelegt wird und sich nicht ändern kann, und dass man die Variablentypen entsprechend dem zugewiesenen Wert verwenden sollte um Fehler zu vermeiden.

## 2.2 Verzweigungen und Schleifen

Verzweigungen und Schleifen sind wichtige Konzepte in der Programmierung, die es ermöglichen, das Verhalten eines Programms an bestimmte Bedingungen anzupassen oder bestimmte Anweisungen wiederholt auszuführen. Visual Basic unterstützt sowohl Verzweigungen als auch Schleifen durch die Verwendung von Anweisungen wie If...Then...Else und For...Next, While...End While und Do...Loop.

Verzweigungen ermöglichen es, das Verhalten eines Programms an bestimmte Bedingungen anzupassen. In Visual Basic wird dies durch die Verwendung der Anweisung If...Then...Else erreicht. Ein Beispiel für die Verwendung von If...Then...Else ist:

```
If x > 0 Then
```

```
    'Anweisungen, die ausgeführt werden, wenn x größer als 0 ist
```

```
Else
```

```
    'Anweisungen, die ausgeführt werden, wenn x kleiner oder gleich 0 ist
```

```
End If
```

Es ist auch möglich, mehrere Bedingungen zu überprüfen, indem man die Anweisung ElseIf verwendet.

Schleifen ermöglichen es, bestimmte Anweisungen wiederholt auszuführen. In Visual Basic gibt es mehrere Arten von Schleifen, darunter For...Next, While...End While und Do...Loop. Ein Beispiel für die Verwendung von For...Next ist:

```
For i = 1 to 10
```

```
    'Anweisungen, die wiederholt ausgeführt werden
```

```
Next
```

Die Schleife wird solange ausgeführt, bis der Wert von i größer als 10 ist. Es ist auch möglich eine Schleife mit While-Bedingung auszuführen, die solange ausgeführt wird, bis eine bestimmte Bedingung erfüllt ist, z.B:

```
While x < 10
```

```
    'Anweisungen die ausgeführt werden
```

```
    x = x + 1
```

```
End While
```

Es ist wichtig, dass die Bedingungen und Zähler korrekt gesetzt und verändert werden, um unendliche Schleifen oder Fehler zu vermeiden.

## 2.3 Funktionen und Prozeduren

In Visual Basic, Funktionen und Prozeduren sind beides Arten von Code-Blöcken, die bestimmte Aufgaben ausführen. Der Unterschied besteht darin, dass Funktionen einen Wert zurückgeben, während Prozeduren keinen Wert zurückgeben.

### 2.3 Funktionen

Funktionen sind Code-Blöcke, die bestimmte Aufgaben ausführen und einen Wert zurückgeben. Sie können als Unterprogramme bezeichnet werden, die innerhalb eines größeren Programms aufgerufen werden können. Funktionen ermöglichen es, Code zu organisieren und wiederzuverwenden. Eine Funktion kann mit Argumenten aufgerufen werden, die als Eingabe verwendet werden, und kann einen Wert zurückgeben, der als Ausgabe verwendet werden kann.

Beispiel:

```
Function AddiereZahlen(ByVal zahl1 As Integer, ByVal zahl2 As Integer) As Integer
```

```
Return zahl1 + zahl2
```

```
End Function
```

In diesem Beispiel wird die Funktion "AddiereZahlen" definiert, die zwei Integer-Argumente nimmt und deren Summe zurückgibt.



## 2.3 Prozeduren

Prozeduren sind Code-Blöcke, die bestimmte Aufgaben ausführen, aber keinen Wert zurückgeben. Sie können ebenfalls als Unterprogramme bezeichnet werden und dienen dazu, Code zu organisieren und wiederzuverwenden. Prozeduren können mit Argumenten aufgerufen werden, die als Eingabe verwendet werden, und können keinen Wert zurückgeben.

Beispiel:

```
Sub GibEineNachrichtAus(ByVal nachricht As String)
```

```
MsgBox(nachricht)
```

```
End Sub
```

In diesem Beispiel wird die Prozedur "GibEineNachrichtAus" definiert, die ein String-Argument nimmt und eine Meldungsbbox mit der übergebenen Nachricht anzeigt.

# Fortgeschrittene Techniken in Visual Basic

## 3.1 Steuerelemente und Formulare

In Visual Basic stehen Ihnen verschiedene Steuerelemente zur Verfügung, die Sie auf einem Formular platzieren können, um bestimmte Funktionen bereitzustellen. Einige Beispiele für Steuerelemente sind Textfelder, Schaltflächen, Listenfelder und Kontrollkästchen. Jedes Steuerelement hat seine eigenen Eigenschaften, Methoden und Ereignisse, die Sie anpassen und verwenden können, um das Verhalten des Steuerelements zu steuern.

Formulare sind Fenster, die Sie zur Erstellung von Benutzeroberflächen verwenden können. Auf einem Formular können Sie Steuerelemente platzieren, um Funktionen bereitzustellen und Daten von oder an den Benutzer zu übertragen. Formulare haben auch ihre eigenen Eigenschaften, Methoden und Ereignisse, die Sie verwenden können, um das Verhalten des Formulars zu steuern.

Beispiel:

Sie können ein Textfeld (TextBox-Steuerelement) auf ein Formular platzieren und es so konfigurieren, dass es nur numerische Eingaben akzeptiert, indem Sie die Eigenschaft "AcceptsReturn" auf "False" und die Eigenschaft "AcceptsTab" auf "False" festlegen. Sie können auch eine Schaltfläche (Button-Steuerelement) hinzufügen, die eine bestimmte Aktion ausführt, wenn der Benutzer darauf klickt, indem Sie dem Ereignis "Click" des Steuerelements eine Prozedur zuordnen.

In diesem Beispiel wurde ein Formular verwendet, um eine Benutzeroberfläche zu erstellen, die ein Textfeld und eine Schaltfläche enthält. Das Textfeld wurde konfiguriert, um nur numerische Eingaben zu akzeptieren und die Schaltfläche wurde programmiert, um eine bestimmte Aktion auszuführen, wenn der Benutzer darauf klickt.

### 3.2 Datenbankzugriff mit Visual Basic

Visual Basic bietet verschiedene Möglichkeiten, um auf Datenbanken zuzugreifen und diese zu verwalten. Eine Möglichkeit ist die Verwendung von ADO (ActiveX Data Objects), einer Bibliothek von Komponenten, die es ermöglichen, auf verschiedene Arten von Datenbanken wie MS Access, SQL Server und Oracle zuzugreifen. Eine andere Möglichkeit ist die Verwendung von LINQ (Language Integrated Query), einer Technologie von Microsoft, die es ermöglicht, Datenbankabfragen in nativer Sprache zu schreiben.

Um auf eine Datenbank mit ADO zuzugreifen, müssen Sie zunächst eine Verbindung zur Datenbank herstellen. Dies kann mithilfe des ADO-Objekts "Connection" erfolgen. Sobald eine Verbindung hergestellt ist, können Sie mit dem ADO-Objekt "Command" Abfragen an die Datenbank senden und mit dem ADO-Objekt "Recordset" das Ergebnis empfangen. Sie können auch mit dem ADO-Objekt "DataAdapter" Daten zwischen einer Datenbank und einem Datencontainer wie einem DataSet oder einer DataTable übertragen.

Beispiel:

```
Dim cnn As New ADODB.Connection
cnn.Open("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\db.mdb")
Dim cmd As New ADODB.Command
cmd.ActiveConnection = cnn
cmd.CommandText = "SELECT * FROM Customers"
Dim rs As New ADODB.Recordset
rs.Open(cmd)
```

In diesem Beispiel wird eine Verbindung zu einer Access-Datenbank hergestellt und eine Abfrage an die Tabelle "Customers" gesendet. Das Ergebnis wird in einem Recordset gespeichert.

LINQ ermöglicht es, Abfragen in nativer Sprache zu schreiben und bietet eine intuitiv verständliche Syntax. Es nutzt die Technologie von Language-Integrated Query (LINQ), um Daten aus verschiedenen Quellen abzufragen, einschließlich Datenbanken, XML und Objekten. LINQ unterstützt sowohl die Abfrage von Daten als auch die Aktualisierung von Daten in einer Datenbank.

Beispiel:

```
Dim db As New DataContext("C:\db.mdb")
Dim query = From c In db.Customers Select c.Name, c.Address
```

In diesem Beispiel wird eine Verbindung zu einer Access-Datenbank hergestellt und eine Abfrage an die Tabelle "Customers" gesendet. Die Ergebnisse sind die Namen und Adressen der Kunden.

Es gibt auch andere Möglichkeiten, auf Datenbanken mit Visual Basic zuzugreifen, wie z.B. über die Verwendung von ODBC (Open Database Connectivity) oder OLEDB (Object Linking and Embedding, Database), die auch verfügbar sind.

Eine wichtige Sache zu beachten, wenn man mit Visual Basic und Datenbanken arbeitet, ist die Verwendung von geeigneten Sicherheitsmaßnahmen, um unbefugten Zugriff auf Daten zu verhindern. Dazu gehören die Verwendung von sicheren Anmeldeinformationen, das Verwenden von Verschlüsselungstechnologien und die Begrenzung der Berechtigungen für Benutzer, die auf die Datenbank zugreifen.

Insgesamt ermöglicht Visual Basic durch verschiedene Möglichkeiten von Datenbankzugriff, Daten zu speichern, zu abfragen, zu bearbeiten und zu manipulieren, um eine Vielzahl von Anwendungen zu erstellen, die Datenbanken nutzen.

### 3.3 Erstellung von Benutzerdefinierten Steuerelementen

In Visual Basic können Sie auch benutzerdefinierte Steuerelemente erstellen, die speziell auf Ihre Anforderungen zugeschnitten sind. Benutzerdefinierte Steuerelemente ermöglichen es Ihnen, eigene Funktionalitäten und Aussehen zu definieren und wiederverwendbar zu machen. Es gibt zwei Arten von benutzerdefinierten Steuerelementen, die Sie erstellen können: benutzerdefinierte Steuerelemente, die auf bestehenden Steuerelementen basieren, und benutzerdefinierte Steuerelemente, die von Grund auf neu erstellt werden.

Ein Beispiel für ein benutzerdefiniertes Steuerelement, das auf einem bestehenden Steuerelement basiert, ist eine erweiterte TextBox, die zusätzliche Funktionalitäten wie die Überprüfung der Eingabe auf numerischen Wert oder die Anzeige von Fehlermeldungen enthält. Um dieses benutzerdefinierte Steuerelement zu erstellen, können Sie eine neue Klasse erstellen, die von der TextBox-Klasse erbt, und die erforderlichen Methoden und Eigenschaften hinzufügen oder überschreiben.

Ein Beispiel für ein benutzerdefiniertes Steuerelement, das von Grund auf neu erstellt wird, ist ein eigenständiger Thermometer-Steuerelement, das die aktuelle Temperatur anzeigt. Um dieses benutzerdefinierte Steuerelement zu erstellen, müssen Sie eine neue Klasse erstellen und die erforderlichen Methoden und Eigenschaften hinzufügen, um das Verhalten des Steuerelements und seine Darstellung zu definieren.

Es ist wichtig zu beachten, dass die Erstellung von benutzerdefinierten Steuerelementen in der Regel etwas komplexer als die Verwendung von vorhandenen Steuerelementen ist und erfordert ein gewisses Verständnis der Programmierung und der verwendeten Technologie. Es ist jedoch eine leistungsstarke Möglichkeit, um Anwendungen anzupassen und zu erweitern und ermöglicht es Ihnen, Zeit und Ressourcen zu sparen, indem Sie wiederverwendbare Codeblöcke erstellen.

## Anwendungsentwicklung mit Visual Basic

### 4.1 Erstellung eines einfachen Programms

Die Erstellung eines einfachen Programms in Visual Basic erfordert die Verwendung der Entwicklungsumgebung von Visual Studio, die Tools und Funktionen bereitstellt, die Sie zum Schreiben, Testen und Bereitstellen von Anwendungen benötigen.

Um ein einfaches Programm zu erstellen, können Sie die folgenden Schritte ausführen:

Öffnen Sie Visual Studio und wählen Sie "Neues Projekt" aus dem Datei-Menü aus.

Wählen Sie im Dialogfeld "Neues Projekt" einen Projekttyp aus, z.B. "Windows Forms Application" für eine Anwendung mit einer grafischen Benutzeroberfläche. Geben Sie dem Projekt einen Namen und wählen Sie einen Speicherort aus.

Nachdem das Projekt erstellt wurde, wird das Formular "Form1" automatisch geöffnet. Sie können es umbenennen, indem Sie auf den Namen "Form1" im Projektextplorer klicken und ihn bearbeiten.

Fügen Sie Steuerelemente wie Textfelder, Schaltflächen, Listenfelder usw. auf das Formular hinzu, indem Sie sie aus der Toolbox auf das Formular ziehen.

Schreiben Sie den Code, der auf bestimmte Ereignisse reagieren soll, indem Sie doppelklicken Sie auf ein Steuerelement im Formular-Designer und den Code im Code-Editor schreiben. Beispielsweise können Sie den Click-Event einer Schaltfläche verwenden, um eine bestimmte Aktion auszuführen, wenn der Benutzer auf die Schaltfläche klickt.

Testen Sie das Programm, indem Sie auf "Start" im Menü "Debuggen" klicken. Sie können auch Breakpoints im Code setzen, um den Programmablauf zu überwachen.

Sobald das Programm getestet und fehlerfrei ist, können Sie es bereitstellen, indem Sie es kompilieren und eine Setup-Datei erstellen oder es als Portable-Anwendung veröffentlichen.

Beachten Sie, dass dies nur ein einfaches Beispiel ist und je nach Anforderungen und Komplexität des Projekts, kann es erforderlich sein, weitere Schritte wie die Verwendung von Datenbanken, Threading, Web-Service-Integration usw. hinzuzufügen.

## 4.2 Erstellung einer Datenbankanwendung

Eine Datenbankanwendung in Visual Basic ermöglicht es Ihnen, Daten zu speichern, zu ändern, abzufragen und zu manipulieren, um eine Vielzahl von Anwendungen zu erstellen, die Datenbanken nutzen.

Um eine Datenbankanwendung zu erstellen, können Sie die folgenden Schritte ausführen:

Erstellen Sie ein neues Projekt in Visual Studio, wählen Sie dazu "Windows Forms Application" oder einen anderen gewünschten Projekttyp aus.

Erstellen Sie eine neue Datenbank oder verwenden Sie eine bestehende. Sie können eine Access-Datenbank oder eine SQL-Datenbank (wie SQL Server oder MySQL) verwenden.

Verbinden Sie sich mit der Datenbank über ADO (ActiveX Data Objects) oder LINQ (Language Integrated Query). Sie können die Verbindungszeichenfolge und die Verbindungsmethoden entsprechend dem verwendeten Datenbanktyps konfigurieren.

Erstellen Sie Tabellen, Indizes, Abfragen und andere Datenbankobjekte, die für Ihre Anwendung erforderlich sind, indem Sie die entsprechenden SQL-Anweisungen ausführen.

Fügen Sie Steuerelemente wie Textfelder, Schaltflächen, Listenfelder usw. auf das Formular hinzu, indem Sie sie aus der Toolbox auf das Formular ziehen.

Schreiben Sie den Code, der auf bestimmte Ereignisse reagieren soll, indem Sie doppelklicken Sie auf ein Steuerelement im Formular-Designer und den Code im Code-Editor schreiben. Beispielsweise können Sie den Click-Event einer Schaltfläche verwenden, um Daten aus der Datenbank zu lesen oder zu schreiben.

Testen Sie das Programm, indem Sie auf "Start" im Menü "Debuggen" klicken. Sie können auch Breakpoints im Code setzen, um den Programmablauf zu überwachen.

Sobald das Programm getestet und fehlerfrei ist, können Sie es bereitstellen, indem Sie es kompilieren und eine Setup-Datei erstellen oder es als Portable-Anwendung veröffentlichen.

Es ist wichtig zu beachten, dass die Sicherheit bei der Erstellung einer Datenbankanwendung eine entscheidende Rolle spielt. Es ist wichtig, geeignete Sicherheitsmaßnahmen zu implementieren, um unbefugten Zugriff auf Daten zu verhindern, wie z.B. das Verwenden von sicheren Anmeldeinformationen, Verschlüsselungstechnologien und die Begrenzung der Berechtigungen für Benutzer, die auf die Datenbank zugreifen.

Auch bei der Entwicklung von Datenbankanwendungen ist es wichtig, die Leistung der Anwendung zu berücksichtigen und Optimierungen vorzunehmen, um eine gute Performance zu gewährleisten, wenn die Anzahl der Benutzer oder die Menge an Daten in der Datenbank zunimmt.

Insgesamt ermöglicht die Erstellung einer Datenbankanwendung in Visual Basic, Daten zu speichern, zu abfragen, zu bearbeiten und zu manipulieren und ermöglicht es, eine Vielzahl von Anwendungen zu erstellen, die Datenbanken nutzen. Es ist jedoch wichtig, die Sicherheit und Leistung der Anwendung zu berücksichtigen und entsprechende Maßnahmen zu ergreifen.

#### 4.3 Erstellung einer Web-Anwendung mit Visual Basic

Die Erstellung einer Web-Anwendung mit Visual Basic erfordert einige Vorbereitungen und Kenntnisse in verschiedenen Bereichen. Im Folgenden werden die Schritte zur Erstellung einer einfachen Web-Anwendung mit Visual Basic beschrieben:

Wählen Sie das Projekttyp: Wählen Sie in Visual Studio "File" -> "New" -> "Project" aus, und wählen Sie dann "ASP.NET Web Application" aus der Liste der verfügbaren Projekttypen. Geben Sie dem Projekt einen Namen und wählen Sie einen Speicherort auf Ihrem Computer aus.

Wählen Sie das Projektvorlage: Wählen Sie "Empty" oder "Web Forms" als Projektvorlage aus. "Empty" ermöglicht es Ihnen, eine benutzerdefinierte Struktur für Ihre Anwendung zu erstellen, während "Web Forms" eine vorgefertigte Struktur mit Standardelementen wie Textfeldern und Schaltflächen bietet.

Erstellen Sie eine HTML-Seite: Erstellen Sie eine HTML-Seite für Ihre Anwendung, indem Sie in Visual Studio "File" -> "New" -> "HTML Page" auswählen. Hier können Sie die Struktur und das Design Ihrer Anwendung definieren, indem Sie HTML-Markup, CSS und JavaScript verwenden.

Fügen Sie Visual Basic-Code hinzu: Fügen Sie Visual Basic-Code hinzu, um die Funktionalität Ihrer Anwendung zu erstellen. Dies kann beispielsweise die Verarbeitung von Formularen, die Abfrage von Datenbanken oder die Verarbeitung von Benutzeraktionen sein. Sie können Visual Basic-Code in der Code-Behind-Datei Ihrer HTML-Seite hinzufügen, indem Sie auf die entsprechende Schaltfläche "Code" in Visual Studio klicken.

Testen und bereitstellen: Testen Sie Ihre Anwendung, indem Sie sie in einem Webbrowser aufrufen, und überprüfen Sie, ob sie wie erwartet funktioniert. Wenn alles in Ordnung ist, können Sie Ihre Anwendung auf einem Webserver bereitstellen, damit sie von Benutzern auf der ganzen Welt verwendet werden kann.

Es ist wichtig zu beachten, dass dies nur ein allgemeiner Überblick über die Schritte zur Erstellung einer Web-Anwendung mit Visual Basic ist und es weitere Schritte und Erwägungen abhängig von den Anforderungen der Anwendung geben kann. Einige wichtige Aspekte, die bei der Entwicklung einer Web-Anwendung mit Visual Basic berücksichtigt werden sollten, sind:

**Sicherheit:** Stellen Sie sicher, dass Ihre Anwendung gegen Angriffe wie SQL-Injection, Cross-Site Scripting (XSS) und Cross-Site Request Forgery (CSRF) geschützt ist.

**Datenbankzugriff:** Verwenden Sie ADO.NET oder ein anderes Datenbankzugriffstool, um auf Datenbanken zuzugreifen und Daten zu speichern und abzurufen.

**Validierung und Sanitierung von Eingaben:** Überprüfen und bereinigen Sie alle Benutzereingaben, um sicherzustellen, dass nur erwartete und zulässige Daten verarbeitet werden.

**Fehlerbehandlung:** Implementieren Sie Mechanismen zur Behandlung von Fehlern und Ausnahmen, um sicherzustellen, dass Ihre Anwendung stabil und robust ist.

**Responsive Design:** Stellen Sie sicher, dass Ihre Anwendung auf verschiedenen Geräten und Bildschirmgrößen gut aussieht und funktioniert.

Es gibt viele Ressourcen und Tutorials online verfügbar, die Ihnen helfen, mehr über die Erstellung von Web-Anwendungen mit Visual Basic zu erfahren und Ihnen bei der Umsetzung Ihrer Projekte helfen können.



# Fehlerbehandlung und Debugging

## 5.1 Fehlerbehandlung mit Try-Catch-Finally

Fehlerbehandlung ist ein wichtiger Aspekt der Softwareentwicklung, da es dazu beiträgt, dass eine Anwendung stabil und robust ist. Eine häufig verwendete Methode zur Fehlerbehandlung in Visual Basic ist die Verwendung von Try-Catch-Finally-Blöcken.

**Try-Block:** Der Try-Block enthält den Code, bei dem ein Fehler auftreten kann. Wenn ein Fehler im Try-Block auftritt, wird die Ausführung des Codes an den Catch-Block übergeben.

**Catch-Block:** Der Catch-Block enthält den Code, der ausgeführt wird, wenn ein Fehler im Try-Block auftritt. Hier kann der Entwickler spezifische Anweisungen hinterlegen, wie mit dem aufgetretenen Fehler umzugehen ist, z.B. das Ausgeben einer Fehlermeldung oder das Schreiben des Fehlers in ein Log-File. Es kann mehrere Catch-Blöcke geben, die auf unterschiedliche Fehlertypen reagieren.

**Finally-Block:** Der Finally-Block enthält den Code, der immer ausgeführt wird, unabhängig davon, ob im Try-Block ein Fehler aufgetreten ist oder nicht. Hier kann der Entwickler z.B. Ressourcen freigeben oder andere Aufräumarbeiten durchführen.

Beispiel:

Try

'Code, bei dem ein Fehler auftreten kann

Catch ex As Exception

'Code, der ausgeführt wird, wenn ein Fehler auftritt

Console.WriteLine("Fehler: " & ex.Message)

Finally

'Code, der immer ausgeführt wird

Console.WriteLine("Finally-Block wurde ausgeführt")

End Try

Es ist wichtig, dass der Entwickler die Art des Fehlers, der auftreten kann, kennt und entsprechende Catch-Blöcke bereitstellt, um darauf reagieren zu können. Es ist auch empfehlenswert, in den Catch-Blöcken nur die notwendigen Anweisungen zur Behandlung des Fehlers auszuführen und nicht zusätzlichen Code hinzuzufügen, der möglicherweise weitere Fehler verursachen kann.

Es ist auch wichtig zu beachten, dass Try-Catch-Finally-Blöcke nicht für die Behandlung von unerwarteten Fehlern verwendet werden sollten, die auf Probleme mit der Anwendungsarchitektur oder auf unerwartete Änderungen der Anforderungen zurückzuführen sind. Stattdessen sollten diese Probleme während der Entwicklungsphase identifiziert und behoben werden.

Es gibt auch andere Methoden zur Fehlerbehandlung in Visual Basic, wie z.B. die Verwendung von On Error Resume Next oder der Verarbeitung von unbehandelten Ausnahmen durch das Application-Objekt. Es ist jedoch in der Regel empfehlenswert, die Try-Catch-Finally-Methode zu verwenden, da sie eine explizite und strukturierte Art der Fehlerbehandlung bietet und es dem Entwickler ermöglicht, spezifisch auf Fehler zu reagieren.

## 5.2 Debugging-Tools in Visual Basic

Debugging-Tools sind wichtig für die Entwicklung von Software, da sie es dem Entwickler ermöglichen, Fehler in ihrem Code schnell und effektiv zu finden und zu beheben. Visual Studio, die Entwicklungsumgebung für Visual Basic, bietet eine Vielzahl von Werkzeugen zum Debuggen, die es dem Entwickler ermöglichen, seinen Code Schritt für Schritt auszuführen, Variablenwerte anzuzeigen und Fehler zu identifizieren.

**Breakpoints:** Ein Breakpoint ermöglicht es dem Entwickler, die Ausführung des Codes an einer bestimmten Stelle anzuhalten und den Zustand der Anwendung zu untersuchen. Der Entwickler kann Breakpoints setzen, indem er einfach auf die linke Seite der Codezeile klickt, an der er den Breakpoint setzen möchte.

**Debug-Fenster:** Das Debug-Fenster ermöglicht es dem Entwickler, Variablenwerte und andere Informationen während der Laufzeit der Anwendung anzuzeigen. Der Entwickler kann das Debug-Fenster öffnen, indem er im Menü "Debug" auf "Windows" -> "Debug" klickt.

**Debug-Steuerelemente:** Mit den Debug-Steuerelementen kann der Entwickler den Code Schritt für Schritt ausführen. Es gibt Steuerelemente wie "Schritt über", "Einzelschritt" und "Fortsetzen", die es dem Entwickler ermöglichen, die Ausführung des Codes zu kontrollieren und an bestimmten Stellen anzuhalten, um den Zustand der Anwendung zu untersuchen.

**Debug-Assistenten:** Visual Studio bietet auch eine Reihe von Debug-Assistenten wie "Call Stack", "Watch" und "Immediate", die dem Entwickler ermöglichen, tiefer in den Code einzutauchen und spezifische Informationen zu erhalten, um Fehler zu identifizieren.

**Debug-Ausgabe:** Die Debug-Ausgabe ist ein Fenster, in dem Nachrichten ausgegeben werden können, die während der Laufzeit der Anwendung generiert werden. Dies kann sehr hilfreich sein, um zu sehen, was in der Anwendung passiert und welche Fehler aufgetreten sind.

Es ist wichtig zu beachten, dass Debugging-Tools nur ein Teil des Entwicklungsprozesses sind. Eine gründliche Planung und eine saubere Architektur können dazu beitragen, die Anzahl der Fehler zu reduzieren, die während der Entwicklung auftreten. Auch regelmäßiges Testen des Codes und die Verwendung von Code-Reviews kann dazu beitragen, Fehler frühzeitig zu erkennen und zu beheben.

Es ist auch wichtig zu beachten, dass jeder Entwickler eigene Präferenzen bei der Verwendung von Debugging-Tools haben kann. Es ist daher ratsam, sich mit den verschiedenen Werkzeugen vertraut zu machen und diejenigen zu verwenden, die am besten für die individuelle Arbeitsweise und die Anforderungen des Projekts geeignet sind.

Einige der Werkzeuge wie Breakpoints und Debug-Steuer-elemente sind für die Verwendung während der Entwicklungsphase gedacht, während andere wie die Debug-Ausgabe und Debug-Assistenten auch in der Produktionsumgebung verwendet werden können, um Probleme zu identifizieren und zu lösen.

Es ist auch wichtig zu erwähnen, dass Visual Studio auch Unterstützung für Remote-Debugging und die Möglichkeit zur Verwendung von Debugging-Tools mit anderen Sprachen und Frameworks bietet.

## Erweiterungen und Add-Ons

### 6.1 Erweiterungen für die Entwicklungsumgebung

Erweiterungen sind Zusatzprogramme, die die Funktionalität einer Entwicklungsumgebung erhöhen und es dem Entwickler erleichtern, seine Arbeit zu erledigen. Visual Studio bietet eine große Auswahl an Erweiterungen, die von der Microsoft-Community und Drittanbietern entwickelt wurden, die es Entwicklern ermöglichen, ihre Produktivität zu steigern und ihre Arbeit zu vereinfachen.

**IntelliSense-Erweiterungen:** IntelliSense ist eine Funktion von Visual Studio, die es dem Entwickler erleichtert, Code zu schreiben, indem sie ihm Vorschläge für Code-Snippets, Methoden, Eigenschaften und mehr anbietet. Es gibt Erweiterungen, die die Funktionalität von IntelliSense erhöhen, indem sie zusätzliche Vorschläge und Funktionen hinzufügen.

**Code-Formatierungserweiterungen:** Diese Erweiterungen ermöglichen es dem Entwickler, seinen Code automatisch zu formatieren, um die Lesbarkeit zu verbessern. Sie können Regeln für die Formatierung von Zeilenlänge, Einrückung, Leerzeichen und mehr festlegen.

**Refactoring-Erweiterungen:** Refactoring bezieht sich auf die Umstrukturierung von Code, um ihn lesbarer, wartbarer und einfacher zu verstehen und zu warten. Es gibt Erweiterungen, die

verschiedene Refactoring-Methoden wie Umbenennen, Extrahieren von Methoden und Verschieben von Code unterstützen.

**Debugging- und Profiler-Erweiterungen:** Diese Erweiterungen erweitern die Debugging-Funktionalität von Visual Studio und ermöglichen es dem Entwickler, tiefer in den Code einzutauchen, um Fehler zu identifizieren und zu beheben. Sie bieten zusätzliche Funktionen wie die Möglichkeit, den Speicherverbrauch zu überwachen, die Leistung des Codes zu messen und die Ausführung von Code-Abschnitten zu visualisieren.

**Projekt- und Dateiverwaltungserweiterungen:** Diese Erweiterungen erleichtern die Verwaltung von Projekten und Dateien innerhalb von Visual Studio. Sie können Funktionen wie das Durchsuchen von Projekten, das Verwalten von Dateiversionen, das Synchronisieren von Dateien mit einem entfernten Repository und vieles mehr bieten.

**Integrationserweiterungen:** Diese Erweiterungen ermöglichen die Integration von Visual Studio mit anderen Tools und Diensten wie Git, JIRA, Trello und Slack. Sie erleichtern die Zusammenarbeit und die Verwaltung von Projekten und ermöglichen es dem Entwickler, direkt aus der Entwicklungsumgebung auf diese Tools zugreifen zu können.

Es ist wichtig zu beachten, dass Erweiterungen die Leistung von Visual Studio beeinflussen können und dass es wichtig ist, sorgfältig auszuwählen, welche Erweiterungen installiert werden, um sicherzustellen, dass sie die Anforderungen des Projekts und die Arbeitsweise des Entwicklers unterstützen. Es ist auch empfehlenswert, regelmäßig überprüfen die installierten Erweiterungen und entfernen diejenigen, die nicht mehr verwendet werden, um die Leistung von Visual Studio zu verbessern.

## 6.2 Add-Ons für Visual Basic

Add-Ons sind Erweiterungen, die die Funktionalität von Visual Basic erhöhen und es dem Entwickler erleichtern, seine Arbeit zu erledigen. Es gibt eine Vielzahl von Add-Ons, die von der Microsoft-Community und Drittanbietern entwickelt wurden, die es Entwicklern ermöglichen, ihre Produktivität zu steigern und ihre Arbeit zu vereinfachen.

**UI-Steuerelemente:** Es gibt viele Drittanbieter-Add-Ons, die es dem Entwickler ermöglichen, benutzerdefinierte UI-Steuerelemente in seine Anwendungen einzubinden, wie z.B. Diagramme, Tabellen, Menüs und mehr. Diese Steuerelemente können die Funktionalität und die Optik der Anwendung erhöhen, ohne dass der Entwickler selbst viel Code schreiben muss.

**Datenbindung:** Es gibt Add-Ons, die es dem Entwickler ermöglichen, Daten aus verschiedenen Quellen wie Datenbanken, Web-Services und lokalen Dateien an seine Anwendungen zu binden. Diese Add-Ons können die Entwicklungszeit verkürzen und die Flexibilität der Anwendung erhöhen.

**Berichtsgeneratoren:** Es gibt Add-Ons, die es dem Entwickler ermöglichen, Berichte in seine Anwendungen zu integrieren, die es ihm ermöglichen, Daten aus verschiedenen Quellen anzuzeigen und zu analysieren. Sie können auch die Möglichkeit bieten, Berichte in verschiedenen Formaten wie PDF, Excel und HTML zu exportieren.

**Mailing- und SMS-Komponenten:** Es gibt Add-Ons, die es dem Entwickler ermöglichen, E-Mail- und SMS-Benachrichtigungen von seiner Anwendung aus zu senden. Dies kann sehr nützlich sein, um Benutzer über bestimmte Ereignisse in der Anwendung zu informieren oder um automatisierte Benachrichtigungen zu versenden.

**Sicherheitserweiterungen:** Es gibt Add-Ons, die es dem Entwickler ermöglichen, die Sicherheit seiner Anwendung zu erhöhen, indem sie verschiedene Sicherheitsfunktionen wie die Authentifizierung von Benutzern, die Verschlüsselung von Daten und die Verwaltung von Berechtigungen bereitstellen. Diese Erweiterungen können die Anwendung vor unbefugtem Zugriff und Datenverlust schützen.

**Cloud-Integration:** Es gibt Add-Ons, die es dem Entwickler ermöglichen, seine Anwendungen in die Cloud zu integrieren, indem sie Funktionen wie die Verbindung mit Cloud-Speicherdiensten, die Verwaltung von Cloud-Instanzen und die Verwendung von Cloud-basierten Diensten bereitstellen. Diese Erweiterungen ermöglichen es dem Entwickler, die Vorteile der Cloud-Technologie zu nutzen, um die Skalierbarkeit, Verfügbarkeit und Effizienz seiner Anwendungen zu erhöhen.

Es ist wichtig zu beachten, dass Add-Ons die Leistung von Visual Basic beeinflussen können und dass es wichtig ist, sorgfältig auszuwählen, welche Add-Ons installiert werden, um sicherzustellen, dass sie die Anforderungen des Projekts und die Arbeitsweise des Entwicklers unterstützen. Es ist auch empfehlenswert, regelmäßig überprüfen die installierten Add-Ons und entfernen diejenigen, die nicht mehr verwendet werden, um die Leistung von Visual Basic zu verbessern.

## Impressum

Dieses Buch wurde unter der  
**Creative Commons Attribution-NonCommercial-NoDerivatives (CC BY-NC-ND) Lizenz** veröffentlicht.



Diese Lizenz ermöglicht es anderen, das Buch kostenlos zu nutzen und zu teilen, solange sie den Autor und die Quelle des Buches nennen und es nicht für kommerzielle Zwecke verwenden.

Autor: **Michael Lappenbusch**

Email: [admin@perplex.click](mailto:admin@perplex.click)

Homepage: <https://www.perplex.click>

Erscheinungsjahr: 2023