

Alkfejl 2

A .NET egy API gyűjtemény, ez alapján adja a .NET implementációkat.

architektúra:

- átfogja a használható programozási nyelveket
- objektum orientált és konzisztens
- fejlesztőkörnyezetként használhatunk Visual Studio-t

MSIL:

- köztes nyelv
- minden támogatott fordító erre fordít

.NET Framework:

- platformfüggő elsősorban Windows
- asztali alkalmazások fejlesztésére optimalizált (Windows Forms és WPF)

NET Core:

- platformfüggetlen
- open source
- támogatja a UWP-t WinForms-t és WPF-et is
- ASP.NET Core segítségével webes alkalmazások készíthetők
- githubon megtalálható

NuGet csomagkezelő rendszer:

- .NET csomagkezelő rendszere
- újrafelhasználható kódok csomagokba szervezése és megosztása
- galériában lehet keresni a csomagok közül
- a csomag a telepítés után elérhető a kódban
- a csomagok közti függőségeket a csomagkezelő rendszer menedzseli

GUI programozás

WinForms:

- .NET Core 3.0 bekerült csak windowson
- alapegysége a Form
- minden Form a Form osztályból származik
- a vizuális elemek a Control osztályból származnak
- konfigurációs file: App.config (adatbázis elérést, logolást és kulcs-érték párokat helyezhetünk el)
- drag and Drop módszerrel lehet felületeket létrehozni
- eseménykezelés eventek segítségével történik
- partial class: hogy a designer által létrehozott kód ne keveredjen az általunk létrehozott kóddal

WPF:

- .Net Framework 3.0-ban jelent meg
- megjelenik a XAML
 - XAML alapú deklaratív felhasználói felület megadás
 - a felület egy hierarchikus fát alkotó elemek alkotják
 - ablakot, dialógust tudunk megadni
 - az egyes xml elemeknek lehetnek attribútumai
- 2Ds és 3Ds grafikai elemek, animációk támogatása
- az ablakok a Windows osztályból származnak
- elsődleges cél: felhasználói interakció lekezelése(pl.: eseménykezelés)

UWP:

- Sotref-os alkalmazások létrehozása
- Windows 10 szükséges
- XAML használat
- asztali alkalmazások
- xbox alkalmazások
- hololens alkalmazások
- IoT alkalmazások
- SurfaceHub alkalmazások
- Windows Runtime-et használnak nem .NET-et

Adatbáziskezelés

Adatforrások elérése Provider-eken keresztül lehetséges:

- Strukturálatlan formában
- Strukturált de nem hierarchikus formában (CSV, Excel)
- Hierarchikus formában (XML dokumentumok)
- Relációs adatbázisok (MS SQL Server, Oracle, ODBC)

Adatbáziskapcsolat:

- DbConnection
- rendelkezik Close és Open metódusokkal
- minden esetben manuálisan kell megnyitni a kapcsolatot
- connectionStringet vár ami tartalmazza az elérési útvonalat
- CreateCommand() segítségével tudunk SQL lekérdezéseket létrehozni

Lekérdezés:

- commandtext
- paraméterezhetünk
- ExecuteScalar: első találat első celláját adja vissza
- ExecuteNonQuery: érintett sorok számát adja vissza
- ExecuteReader: readert ad vissza SELECT esetén hasznos

DataReader:

- tároló, lekérés után itt tudjuk tárolni az adatokat
- Read() segítségével tudunk rajta végigiterálni
- Close(), Read(), NextResult() metódusok
- az egyes értékeket GetType(), GetOrdinal() metódusokkal tudjuk elérni

Webprogramozás

ASP.NET Core

- platformfüggetlen
- open source
- felhő alapú alkalmazások
- web servicek
- IoT alkalmazások
- mobil alkalmazások backendje
- futtatható .NET Core-on és .NET Framework-ön is
- ASP.NET Core modernebb mint az ASP.NET Frameworknek
 - tesztelhetőbb
 - használható Angulárral vagy React-tal is
 - hostolható
 - használhatunk C# kódot a böngészőben a JavaScript mellett