

MAUI dokumentatsioon

1. Kuidas C# kasutatakse MAUI'is

C# keelt kasutatakse .NET MAUI (Multi-platform App UI) platvormil mitmeplatvormiliste mobiilirakenduste loomiseks. .NET MAUI kasutab C# programmeerimiskeelt, kuna see võimaldab arendajatel luua turvalisi ja tõhusaid rakendusi.

C# keelt kasutatakse .NET MAUI platvormil ka selleks, et luua ühendusi erinevate teenustega, nagu andmebaasid, võrguteenused ja pilveteenused. C# keelel on tugev objektorienteeritud programmeerimine, mis võimaldab arendajatel luua modulaarseid ja skaleeritavaid rakendusi.

.NET MAUI platvorm kasutab XAML-i (Extensible Application Markup Language) kasutajaliideste loomiseks, mis võimaldab kiiret ja lihtsat kasutajaliidese kujundamist. C# keel võimaldab arendajatel luua rakendusi, mis töötavad nii Androidis, iOS-is kui ka Windowsis.

Kokkuvõttes kasutatakse .NET MAUI platvormil C# keelt, et luua kiireid, turvalisi ja mitmeplatvormilisi mobiilirakendusi. C# keele tugevad funktsioonid ja võimalused võimaldavad arendajatel luua kõrgkvaliteetseid rakendusi, mis töötavad sujuvalt mitmel platvormil.

2. Teekid (libraries)

C# teek on eelnevalt kirjutatud korduvkasutatavate koodimoodulite või komponentide kogum, mida saab kasutada C# programmeerimisülesannete lihtsustamiseks.

See on sisuliselt koodipakett, millele C# projekt saab viidata, et pakkuda tavapäraselt kasutatavaid funktsioone või täita konkreetseid ülesandeid.

Teeke saab kasutada arendusaja säästmiseks, koodi kvaliteedi parandamiseks ja projekti jaoks vajaliku kodeerimise hulga vähendamiseks. Konkreetse funktsiooni täitmiseks uue koodi kirjutamise asemel saab arendaja lihtsalt viidata raamatukogule, mis pakub vajalikke funktsioone.

See võib aidata vähendada vigu ja parandada koodi üldist kvaliteeti, kuna pärast koodi kirjutamist on vähem vigu, mis võivad kahe silma vahele jääda.

Erinevalt programmist pole teegid iseseisvalt töötavad, vaid on programmidele erinevaid teenuseid pakkuvad abifailid.

Teekidel on tavaliselt .dll failitüüp, aga võib olla ka .lib kuigi nendel kahel on vahe (.dll on Dynamic Link Library ehk dünaamiline aga .lib on static library ehk staatiline).

3. keerukamad concept'id

Geneerilised tüüp-parameetrid on parameetrid, mis töötavad igat sorti andmetüüpidega. Kui kompilaator satub geneerilise tüübi peale programmi kompileerides, siis genereerib ta koodi, mis on võimeline ette antud andmetüübiga töötama.

Sünkroonsuse saavutamiseks asünkroonses koodis on C#'is võimalik kasutada "lock" lauset. Sellega saame kindlustada, et mingi kindla koodi osaga tegeleb ainult üks lõim. Teised lõimed peavad ootama kuni lukk "vabastatakse." lukud sarnanevad Javas esinevate "synchronous" funktsiooni tüüpidega.

Evendid lubavad klassidel/objektidel omavahel suhelda/teada anda kui midagi juhtub. Kõige tihedamini kasutatakse UI elementide puhul, näiteks nupp, millele vajutades anname mingile teisele objektile/klassile teada, et seda on vajutatud.

4. XAML

XAML on XML-põhine deklaratiivne keel, mida kasutatakse enamasti .NET tehnoloogiates. MAUI kasutab XAML'i kasutajaliidese elementide defineerimiseks ning sidumiseks sündmuste ja andmetega.

XAML'i suur eelis tavalise koodi kirjutamisest on selle lihtne loetavus, mis teeb kasutajaliidese kirjutamise palju lihtsamaks. Miinusteks on see, et koodi ei saa XAML'i sisse kirjutada, peab eraldi koodi failis defineerima; XAML'is ei saa tsükleid kasutada; tingimuslikke lauseid pole võimalik sisestada ja mõned veel. Enamasti samad piirangud nagu ikka deklaratiivsetes keeltes nagu HTML.

MAUI kasutab kahte tüüpi faile, kus XAML'i kasutatakse. Failis .xaml on puhtalt ainult xaml'i kirjutatud ja .xaml.cs failides kirjutatakse koodi mis on seotud xaml'i failidega.

XAML meenutab tugevalt HTML'i koodi, kui olete varem veebiprogrammeerisega tegelenud, siis peaks olema lihtne XAML ära õppida. XAML'il võib vahel puududa sarnased kujundamise võimalused nagu CSS'il, selle asemel saab C# koodi kirjutada keerulisematel olukordadel.

5. Sarnasused ja erinevused

Sarnasused:

1. C# on C-põhine keel, mis tähendab, et selle süntaks on sarnane C ja C++-ga. See teeb selle õppimise lihtsamaks, kui on olemas eelnev kogemus nende keeltega.
2. Nagu enamus programmeerimiskeeli, on C# võimeline looma funktsioone, klassid, muutujaid ja muid programmeerimise objekte.
3. C# kasutab ka Managed Code süsteemi, mis tähendab, et keel kontrollib koodi kasutamist. See muudab koodi turvalisemaks ja aitab vältida vigu. Väga lihtsalt öeldes on hallatud kood just see: kood, mille täitmist haldab käituskeskkond (runtime).

Erinevused:

1. C# on mõeldud peamiselt Windowsi platvormile, samas kui paljud teised programmeerimiskeeled on saadaval mitmel platvormil.
2. C# kasutab .NET Frameworki, mis sisaldab palju valmis komponente ja raamistikke, mida saate oma koodis kasutada. See võib muuta arendusprotsessi kiiremaks ja lihtsamaks.
3. C# on tugevalt orienteeritud objektidele. See tähendab, et kõik programmeerimise objektid on klassid ja C# kasutab objektorienteeritud disaini põhimõtteid nagu pärilikkus ja polümorfism.
4. C# on objektorienteeritud programmeerimiskeel. Objektorienteeritud programmeerimise neli põhiprintsiipi on järgmised: (Google translate-iga tõlgitud)
 - 4.1. Abstraksioon Olemite asjakohaste atribuutide ja interaktsioonide modelleerimine klassidena, et määratleda süsteemi abstraktne esitus.
 - 4.2. Kapseldamine Objekti sisemise oleku ja funktsionaalsuse peitmine ning juurdepääsu võimaldamine ainult avaliku funktsioonide komplekti kaudu.
 - 4.3. Pärimine Võime luua uusi abstraktsioone olemasolevate abstraktsioonide põhjal.
 - 4.4. Polümorfism Võime rakendada päritud omadusi või meetodeid erinevatel viisidel mitme abstraktsiooni ulatuses.
5. C# sisaldab ka mõningaid funktsioone, mida teistes keeltes ei leidu, nagu näiteks sündmuste käsitlemine ja LINQ (Language Integrated Query), mis võimaldab andmetega töötamist lihtsamaks muuta.

Kokkuvõttes on C# sarnane paljudele teistele programmeerimiskeeltele, kuid sellel on ka mitmeid erinevusi, mis muudavad selle sobivaks erinevatele kasutusjuhtumitele.