



HALADÓ PROGRAMOZÁS

Előadások/Gyakorlatok:	
Eseménykezelés alapjai, gyakorlati megvalósítása	2+2
Delegáltak készítése, a .NET beépített delegáltjainak használata	2+2
Fájlok és könyvtárak kezelése	2+2
XML és JSON fájlok létrehozása, feldolgozása	2+2
Nyelvbe ágyazott lekérdezések (LINQ) alapjai	2+2
LINQ lekérdezések objektumgyűjteményeken, XML és JSON fájlokon	2+2
DLL-ek készítése, felhasználása	2+2
Attribútum alapú programozás, reflexió alapjai	2+2
Adatbázis elérési lehetőségek Adatbázisok felhasználása	2+2
Adatbázisok készítése DbFirst módszerrel	2+2
Adatbázisok készítése CodeFirst módszerrel	2+2
ZH	2+2
Féléves feladat bemutatása	2+2
Féléves feladat pótbemutatása	2+2

Tematika

- 1 db elméleti ZH és a gyakorlat teljesítése

Követelmények

- Az aláírás megszerzésének feltétele: A **gépes ZH**, és az **elméleti ZH** megírása, a **házi feladat** beadása. A ZH-k és házifeladatból megszerezhető pontok 50%-nak teljesítése.
- A félévközi jegy a három részjegy átlaga, de egyik sem lehet elégtelen.

Osztályozás:

83% 5

75% 4

62% 3

50% 2

<50% 1

25% alatt letiltás

Könyv



Scott Chacon, Ben Straub Pro Git Version 2.1.448, 2025-07-25



Troelsen, Andrew, and Phil Japikse. *Pro C# 8 with .NET Core 3: Foundational Principles and Practices in Programming*. Apress, 2020.



Richter, Jeffrey. *CLR via C*. Pearson Education, 2012.

Bevezető (Miért tanuljuk ezt?)

A .Net platform és a C# nyelv eredete 2002-ben indult és azóta egy fontos eszköze a modern szoftverfejlesztésnek

Nagyszámú programnyelvet lehet vele használni (C#, VB, F#)

2016-ban indul a .NET Core – összekötni az MS és Linux világot

2021 C# 10 és .NET 6 (külön verzió és egymástól függetlenül telepíthető)

Felülről kompatibilitás

Verziókezelés - GIT

- **Mi ez?**

Rendszer, amely **időben rögzíti a fájlok változásait**, így bármikor visszahívhatók korábbi állapotok.

- Általánosan elterjedt valamilyen verziókezelés használata
- Szoftverfejlesztésben kötelező
- Egyénileg és csoportban is használható

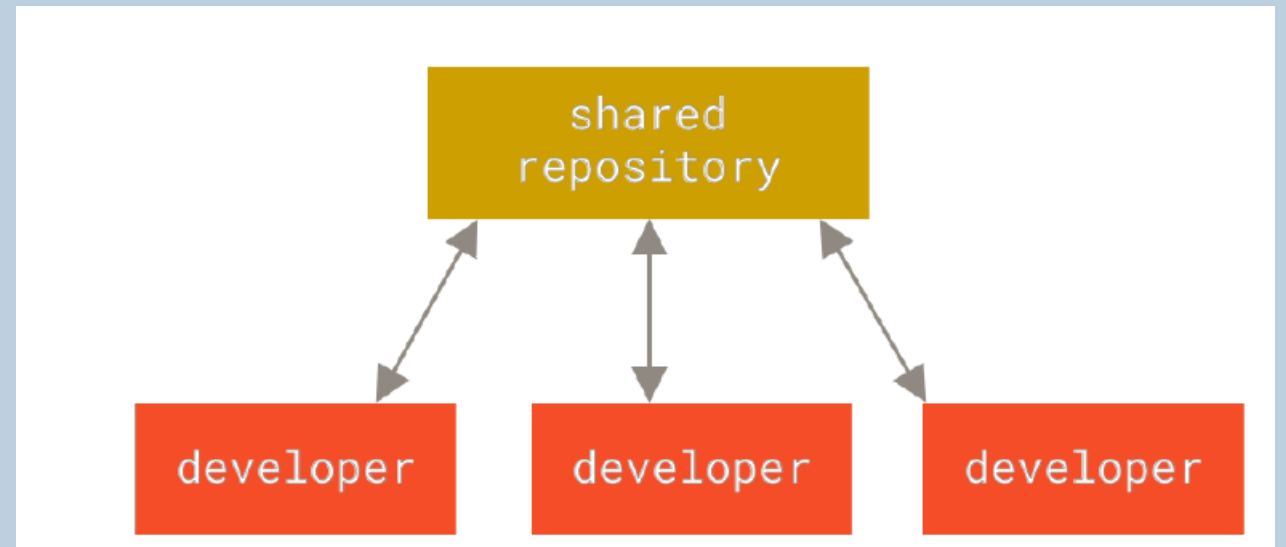
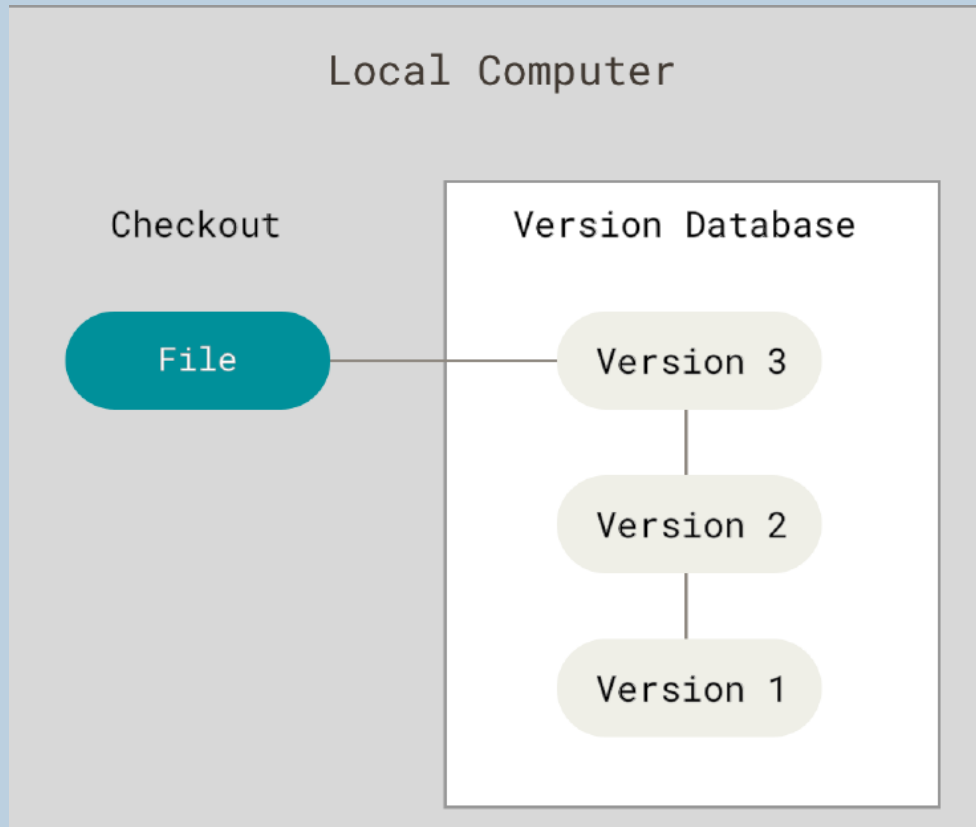
Miért fontos?

- **Visszaállítás:** egy fájl vagy az egész projekt visszatekerhető korábbi verzióra.
- **Összehasonlítás:** látható, mi változott és mikor.
- **Nyomon követés:** ki módosított mit; hibát okozó változtatás beazonosítható.
- **Biztonság:** elrontott vagy elveszett fájlok könnyen helyreállíthatók.
- **Alacsony overhead:** sok előny, kevés többletmunka mellett. Nem csak programkódra: képek, dizájnok, dokumentumok is kezelhetők.

Rossz gyakorlat vs. helyi VCS

- **Kézi másolgatás** (időbélyeges mappák): egyszerű, de erősen hibalehetőséges (rossz mappa, felülírás, keveredés).
- **Helyi adatbázis**: egyszerű adatbázis tartja nyilván a fájlok minden módosítását.
- megszünteti a kézi másolásból fakadó hibákat.

Verziókezelés- hagyományosan



A Git céljai (2005)

- **Sebesség**
- **Egyszerű dizájn**
- **Erős támogatás nemlineáris fejlesztéshez** (sok párhuzamos ág)
- **Teljesen elosztott** működés
- **Nagy projektek hatékony kezelése** (idő és tárhely)

Elv: Pillanatképek, nem különbségek

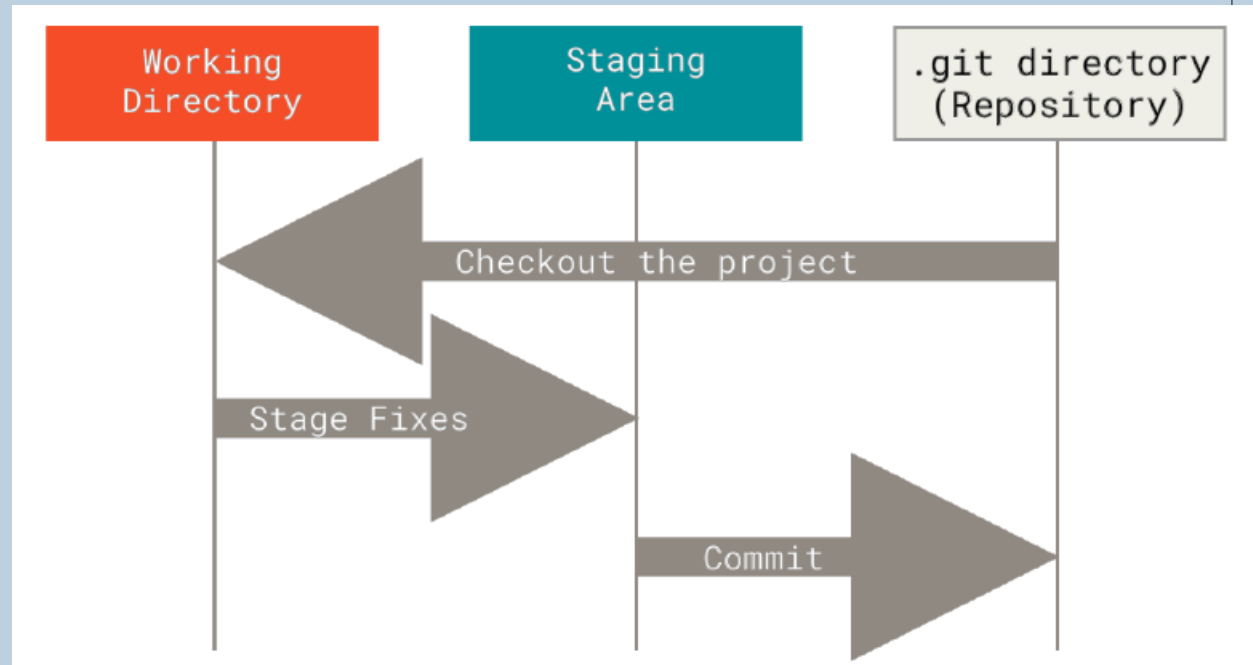
- A legtöbb VCS **delta-alapú** (fájlankénti változások listája).
- A Git **pillanatképekben (snapshots)** gondolkodik: minden commit a projekt **akkori állapotának képe**.
- Azonos fájlokat nem duplikál: csak **hivatkozást** tárol a már meglévő változatra.
- Következmények: gyors ágkezelés, erős integritás, hatékony tárhely-használat.

Kulcsfogalmak

- **Repository:** a projekt és a története.
- **Commit:** pillanatkép + metaadat.
- **Branch:** könnyű, mozgatható mutató commitokhoz.
- **Merge/Rebase:** ágelágazások egyesítése/újrarendezése.
- **Integritás:** Minden változást észrevesz, mert a file-hoz ellenőrző összeg tartozik.

Projekt állapotok

- A módosított (**modified**) azt jelenti, hogy módosítottad a fájlt, de még nem véglegesítetted az adatbázisodban.
- Az előkészített (**staged**) azt jelenti, hogy megjelölted a módosított fájlt az aktuális verziójában, hogy bekerüljön a következő készített pillanatképedbe.
- A véglegesített (**committed**) azt jelenti, hogy az adatok biztonságosan tárolódnak a helyi adatbázisodban.



Git használata

- Parancssoros
- GUI
- Hálózati
- Programfejlesztő
környezetbe épített

Sok Sikert!!!