

# IntelliJ IDEA

CAPABLE AND  
ERGONOMIC JAVA IDE

## L0: Fejlesztői eszközök és kellékek



Ezen laborgyakorlat célja a fejlesztői eszközök telepítése és példaprogramok kipróbálása. A laborgyakorlatok megoldásához használhatunk Windows, Linux vagy Mac OS X operációs rendszert.

### Fejlesztői eszközök

A tantárgy laborfeladatainak sikeres megoldásához szükségünk lesz különböző szoftver eszközökre, amelyeket otthoni használat céljából telepítenünk kell a saját számítógépünkre. A legfontosabb eszközök listája a következő:

#### 1. [git verziókövető](#). Használati opciók:

- konzolból a velejáró *Git Bash* segítségével;
- a szerkesztő pluginja segítségével - ld. [git IDEA-ból](#), [git VS Code-ból](#);
- külső git kliensalkalmazás, pl. [SourceTree](#), [GitKraken](#), stb.
- tesztparancs (új konzolból): `git version`

#### 2. [Java fejlesztői csomag](#) (Java Development Kit, JDK)

- ajánlott a 8-as verzió legújabb változatát telepíteni
- állítsuk be a `JAVA_HOME` környezeti változót a JDK telepítési helyére
- adjuk hozzá az eszköz `bin` almappját a `PATH` környezeti változóhoz
- tesztparancs (új konzolból): `java -version`, `javac -version`

#### 3. [Gradle](#) build- és függőségkezelő rendszerek

- állítsuk be a `GRADLE_HOME` környezeti változót a Gradle telepítési helyére
- adjuk hozzá az eszközök `bin` almappjait a `PATH` környezeti változóhoz
- tesztparancs (új konzolból): `gradle -v`

#### 4. IntelliJ IDEA

- minimum *Community Edition*

#### 5. Tomcat Java-alapú webkonténer

- állítsuk be az `CATALINA_HOME` és `CATALINA_BASE` környezeti változókat a Tomcat telepítési helyére
- adjuk hozzá az eszköz `bin` almappját a `PATH` környezeti változóhoz
- tesztparancs: `catalina version` (Windows-on lehet `catalina.bat version`)

## Példaprogramok

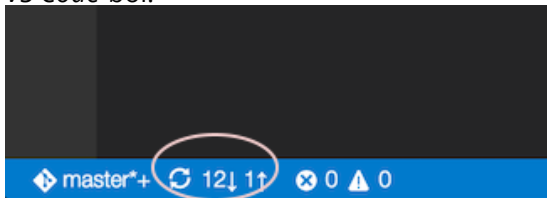
- Az előadás különböző tematikáihoz tartozó példaprogramok elérhetőek egy [publikus git tárolón](#).
- A példaprogramok tematika szerint vannak csoportosítva. Minden tematikának egy külön könyvtár felel meg, ezen belül megtalálhatóak az aktuális példaprogramok.

### **Első letöltés**

- git konzol/parancs:
  - navigáljunk azon mappába, ahova letöltenénk a példákat: `cd <celpont>`
  - letöltés: `git clone https://gitlab.com/ubb-idde/examples` (létrehoz egy `examples` almappát)
- VS Code-ból:
  - Command palette: `Ctrl+Shift+P`
  - „Git: Clone”
  - Repo URL: `https://gitlab.com/ubb-idde/examples`

### **A tartalom frissítése**

- git konzol/parancs:
  - navigáljunk azon mappába, ahova korábban letöltöttük a példákat: `cd <celpont>/examples`
  - frissítés: `git pull`
- VS Code-ból:



### **Hiba esetén**

Hiba léphet fel, hogyha a lokális tárolóban levő tartalom *megváltozott* (ha kézzel változtatunk a példaprogramok tartalmán). A könyvtár *tisztításáért* hívjuk a következőt ugyanazon `examples` mappából (**Vigyázat!**: Ezen parancsok meghívása a laborfeladatos tárolóban kitörölheti annak módosításait):

```
git reset --hard HEAD
git clean -xfd
```

## Saját laborfeladatos tároló

- A laborfeladatokat a GitLab verziókövető portálra kell feltölteni. Minden diáknak automatikusan készül egy-egy projekt, melynek URL-je:  
<https://gitlab.com/ubb-idde/labs2021/lab-<abcd1234>>  
ahol az **abcd1234** megfelel a felhasználói SSID-jának.
- A megadott tárolón minden házi feladatnak hozzunk létre **egy külön ágot**, s egy ehhez tartozó **merge requestet**. Ajánlott csak akkor létrehozni egy bizonyos ágot, amikor a korábbi feladatokhoz tartozó ágak már merge-elve vannak.
- Ahhoz hogy a projekthez elérést nyerjete, **készítsetek felhasználót a GitLabon**, majd adjátok le az itt használt **e-mail címeteket** bármely oktatónak.
- A feladatokat **inkrementálisan** kell megoldani, tehát pl. a 2-es feladat pontozására nem nyílik lehetőség, amíg az 1-es nincs elfogadva.
- Az értékelésnél kizárólag a GitLabra felpusholt állományokat vesszük figyelembe. További részletek a tantárgy syllabus-ában.

## Gyakorlat

1. Klónozzuk mind a saját projektet, mind a példaprogramokat tartalmazó git projektet. **Ezek ne ágyazódjanak egymásba - ne a példarepő belsejébe klónozzuk a sajátot, vagy fordítva!**
2. IntelliJ IDEA-t használva **importáljuk** a példaprogramok tárolójából a [01-build/gradle-application](#) projektet. Használjuk a Gradle modellt importkor. Próbáljuk futtatni a [HelloWorld](#) osztályt IDEA-ból.
3. Nyissunk meg az IDEA-ban egy konzolt (**Alt+F12**). Próbáljuk futtatni az alkalmazást Gradle segítségével: **gradle run**. Bizonyosodjunk meg, hogy mindkét futtatási módszer jelezte a megfelelő üzenetet a konzolra.

## Bónusz: SSH használata gittel

Annak érdekében, hogy ne kelljen minden pull/push operációkor *felhasználót és jelszót* megadni a gitnek, konfigurálható az SSH-s elérés. Ennek lépései:

1. Lokálisan a *Git Bash*-ből: **ssh-keygen** - minden kért argumentumot **üresen hagyva**.
2. **cat ~/.ssh/id\_rsa.pub** - ez kiírja a publikus SSH kulcsot.
3. A profilodon a *Settings* -> *SSH Keys* részbe másold be az SSH kulcsot *teljességében* (**ssh-rsa** kulcsszó az elején, a géped neve a végén).
4. A projekt oldalán másold le az SSH-s elérési linket - ennek formája **git@gitlab.com:ubb-idde/...**
5. A fenti HTTP-s **clone** parancs helyett: **git clone <link>** Ha már klónoztad a repőt korábban, akkor annak folderén belül: **git remote set-url origin <link>**