

Git, github használata Eclipse/Java projektekhez

A git verziókezelő rendszert előbb telepíteni kell a gépre (letöltés: <https://git-scm.com/download/win>).

Ha telepítésnél az alapértelmezett opciókat választjuk, a windows parancssorából már tudjuk is használni a git-et.

Ha már létező projektet szeretnénk letölteni, indítsuk el a parancssort és a cd parancs segítségével navigáljunk el abba a könyvtárba, ahova a projektet másolni szeretnénk. Jó ha ez már eleve az a könyvtár, ahova az Eclipse a projektjeit helyezi, de ez nem kötelező (az Eclipse is be tud olvasni projektet más útvonalról).

A projekt lemásolásához a git clone parancsot használjuk. Például a <https://github.com/webler-duka/barkoba.git> projekthez:

```
Microsoft Windows [Verzió] Copyright (c) 2017 Microsoft Corporation. Minden jog fenntartva.

C:\Users\Duka Árpád>cd workspace

C:\Users\Duka Árpád\workspace>git clone https://github.com/webler-duka/barkoba.git
Cloning into 'barkoba'...
remote: Counting objects: 22, done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 22 (delta 3), reused 22 (delta 3), pack-reused 0
Unpacking objects: 100% (22/22), done.

C:\Users\Duka Árpád\workspace>cd barkoba

C:\Users\Duka Árpád\workspace\barkoba>dir
Volume in drive C has no label.
Volume Serial Number is D4F0-75DC

Directory of C:\Users\Duka Árpád\workspace\barkoba

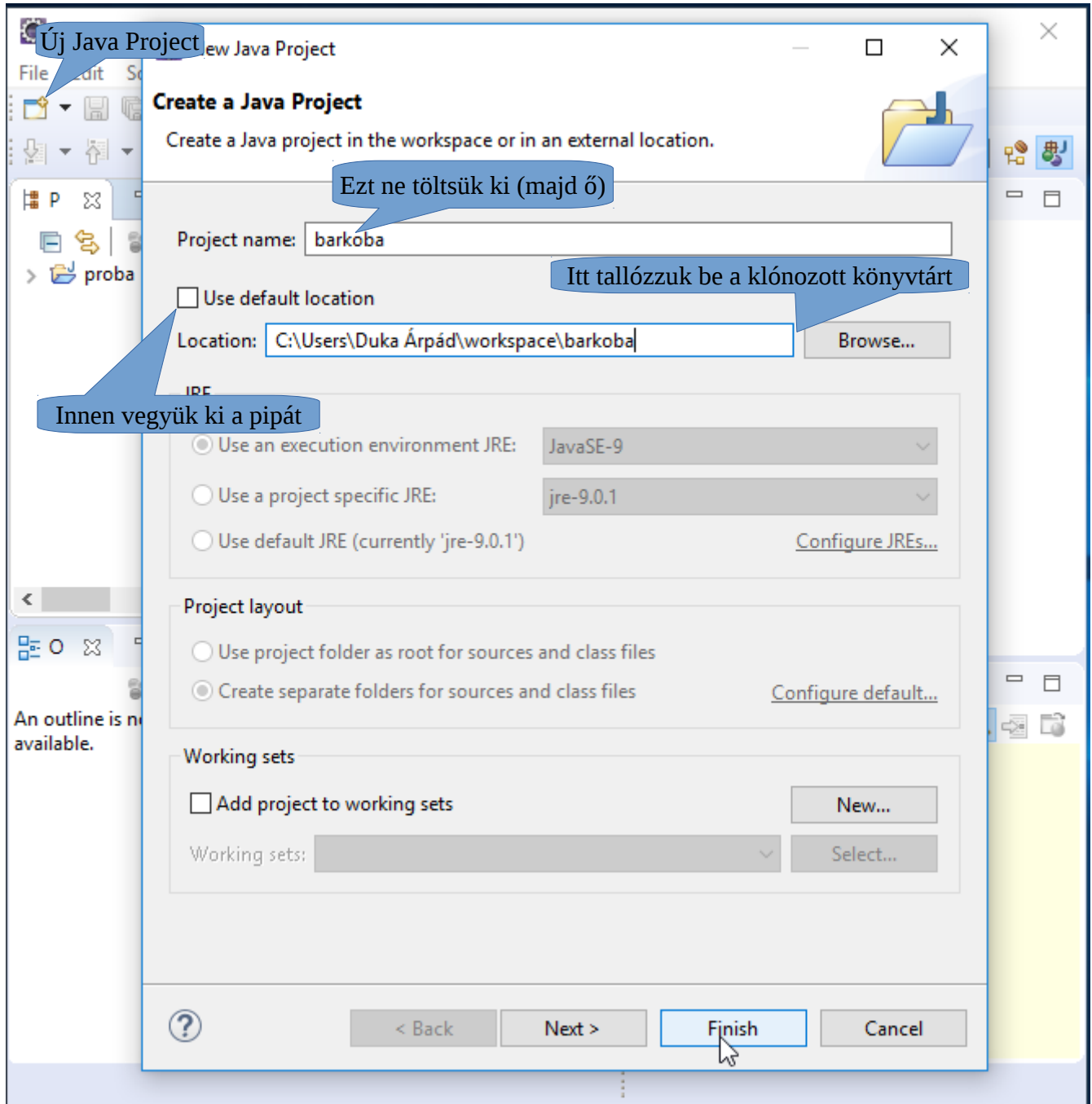
2017. 11. 18. 22:24 <DIR>      .
2017. 11. 18. 22:24 <DIR>      ..
2017. 11. 18. 22:24          422 .classpath
2017. 11. 18. 22:24          271 .gitignore
2017. 11. 18. 22:24          383 .project
2017. 11. 18. 22:24 <DIR>      src
2017. 11. 18. 22:24 <DIR>      test
                3 File(s)      1 076 bytes
                4 Dir(s)      3 818 680 320 bytes free

C:\Users\Duka Árpád\workspace\barkoba>
```

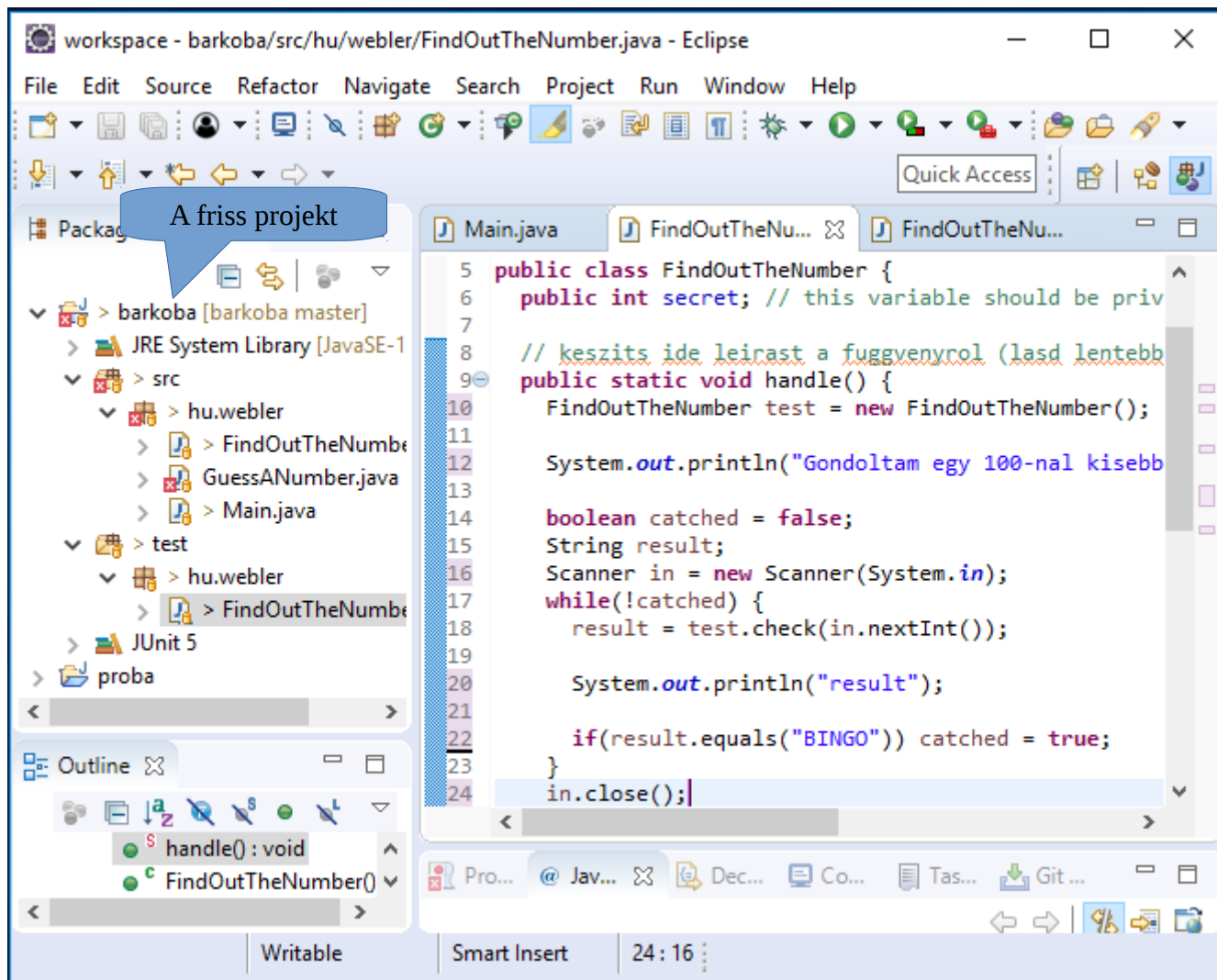
Projekt megnyitása Eclipse fejlesztőkörnyezetben

A File alatti plusszos ablakos gombra kattintva válasszuk a Java Projekt opciót.

A felugró ablakban ne írjunk semmit a „Project name” mezőbe, hanem vegyük ki a pipát alatt a „Use default location” checkboxból és az alatta levő „Location” sornál tallózzuk ki a frissen klónozott projektünk könyvtárát. Ez után „Finish”.

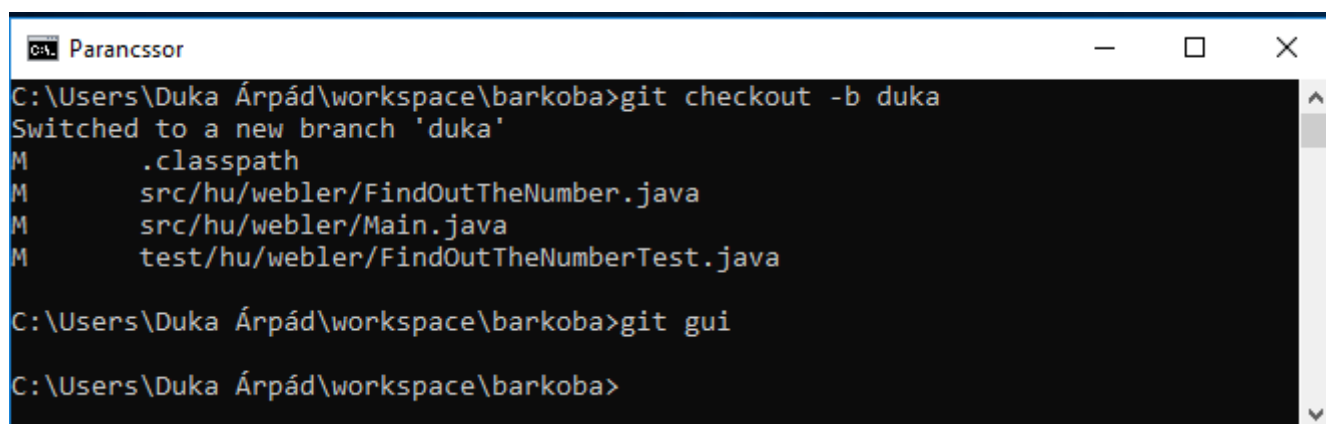


A Finish gomb megnyomása után létrejön a projektünk, amit már szerkeszthetünk és futtathatunk is.



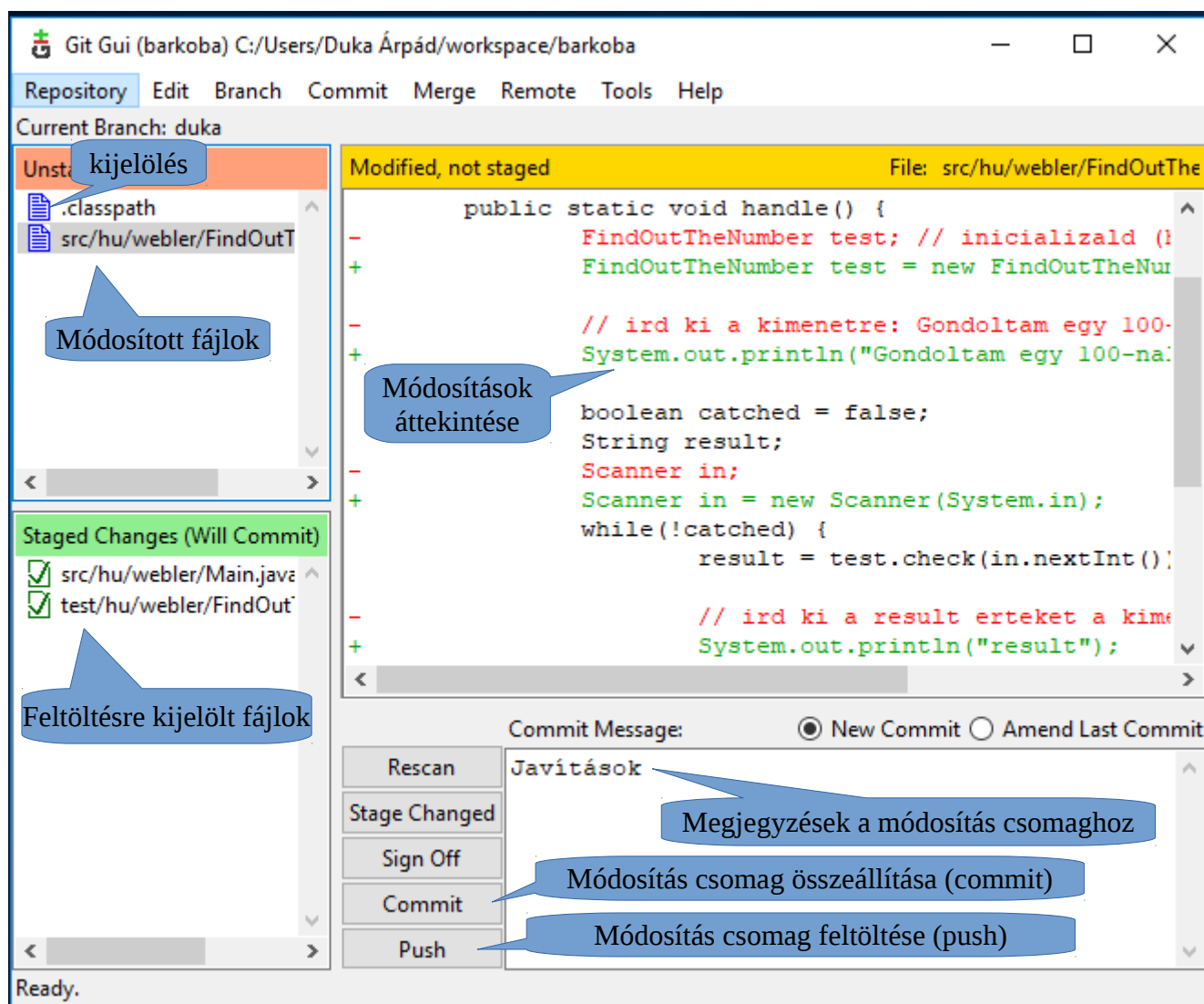
Módosítások visszatöltése a github-ra

Miután elvégeztük a módosításokat, nyissuk meg újra a windows parancssort a projekt könyvtárban. Hozzunk létre egy új munka branch-et, ez lehet például a nevünk. Új branch létrehozása a **git checkout -b branchnev** paranccsal lehetséges. Ez a parancs át is vált az új branch-re.



Itt indítsuk el a **git gui**-t. Ez a gitnek egy grafikus ablaka, ami lehetővé teszi a feltöltendő módosítások kijelölését és feltöltését (commitolás, pusholás).

A felugró ablak a bal felső dobozban listázza azokat a fájlokat, amelyeket módosítottunk, vagy újonnan létrehoztunk. Ha a fájlok nevére kattintunk, a jobb felső részben megjelenik a fájl tartalma jelölve azt, hogy a fájlban milyen módosításokat végeztünk. A módosításokat úgy jelöli (soronként), hogy az eredeti változatot töröltként jeleníti meg (pirossal, a sor elején „-” jellel), az új verziót pedig új sorként tünteti fel (zöldel, a sor elején „+” jellel).

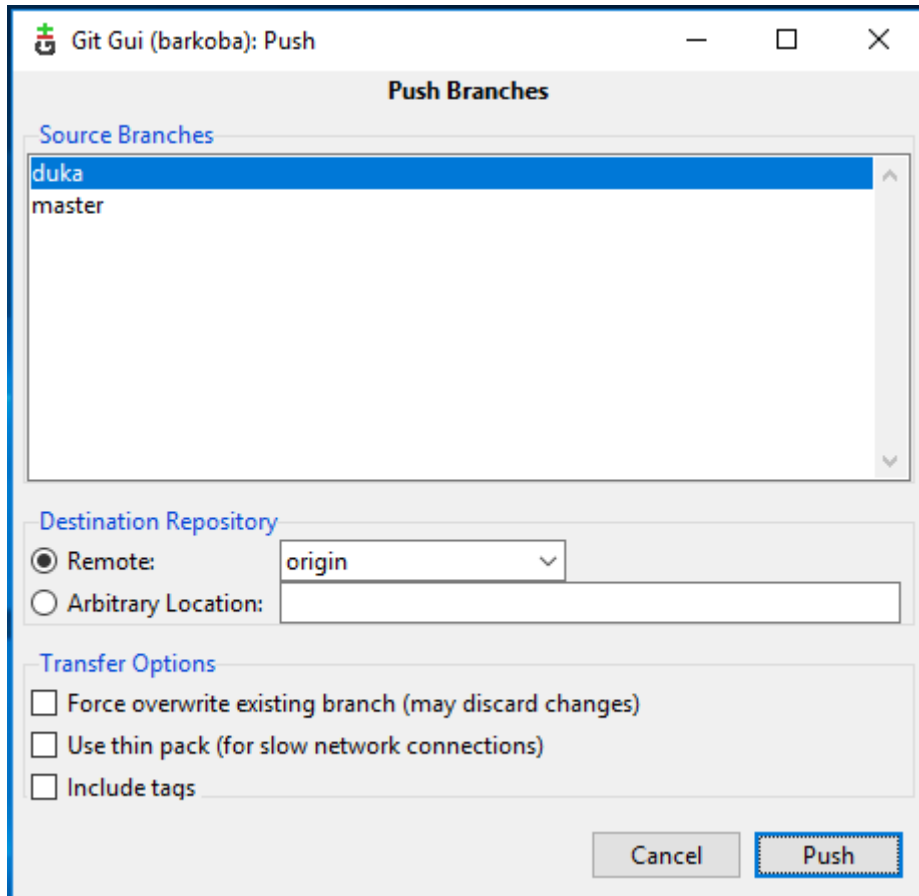


A módosított fájlok közül nem kötelező mindegyiket betenni a módosítási csomagba. Amelyiket be akarjuk tenni, azokat úgy jelölhetjük ki, hogy a bal felső listában a fájl neve előtti kis kék fájl ikonra kattintunk. Hasonlóképpen a feltöltésre kijelölt fájlok közül visszavonhatunk egyeseket ugyancsak a fájlnev előtti zöld pipás ikonra kattintva.

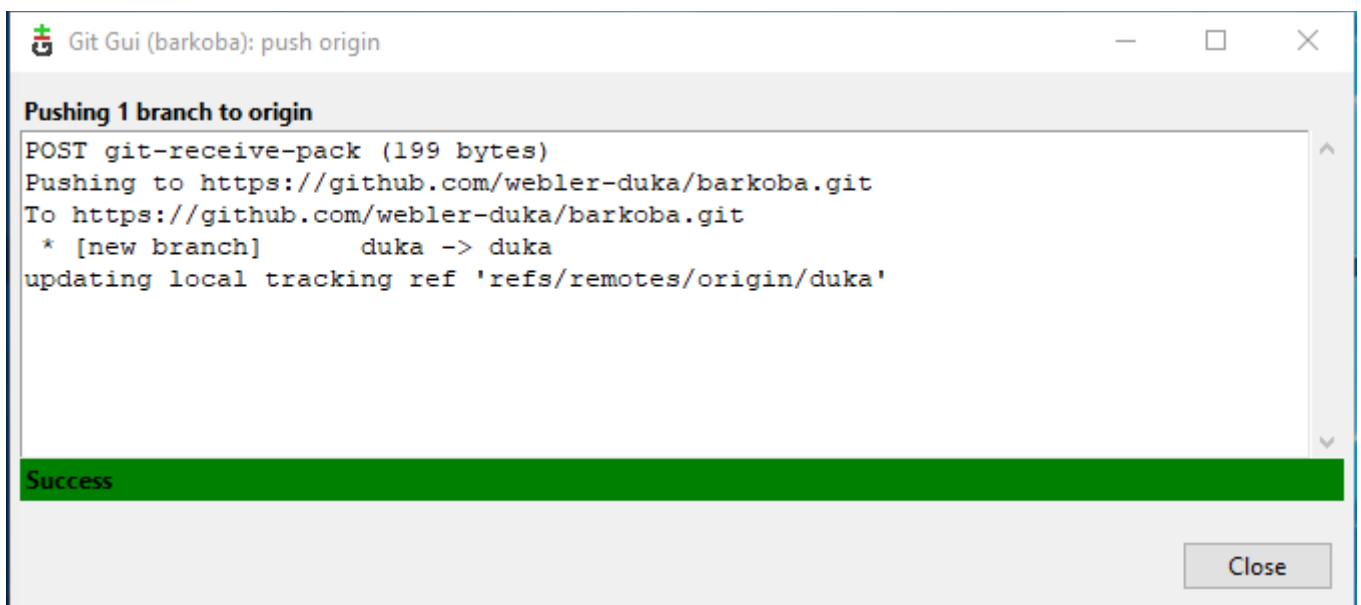
Miután átnéztük a módosításokat és kijelöltük azokat a fájlokat, amiket fel akarunk tölteni, létre tudjuk hozni a módosítási csomagot. Ezt a csomagot hívjuk commit-nak. Ehhez a jobb alsó szerkesztő mezőben adjunk meg egy megjegyzést. Ez lesz a commit címe, tájékoztatója. Ide általában azt szoktuk összefoglalni, hogy milyen módosításokat, javításokat hajtottunk végre, milyen feladatot kapcsán. A megjegyzés megírását követően kattintsunk a Commit feliratú gombra.

Ezt követően már elkészült a módosítási csomag (commit), de még nem került fel a github szerverére. Ezt egy külön tevékenység keretében tehetjük meg. Ezt a feltöltést nevezzük push eseménynek. Több commitot is lehet készíteni pusholás előtt, és több commitot is lehet egyben pusholni.

A Push gomb megnyomását követően megjelenik egy ablak, ami azt kérdezi, hogy a github melyik branch-re szeretnénk feltölteni.



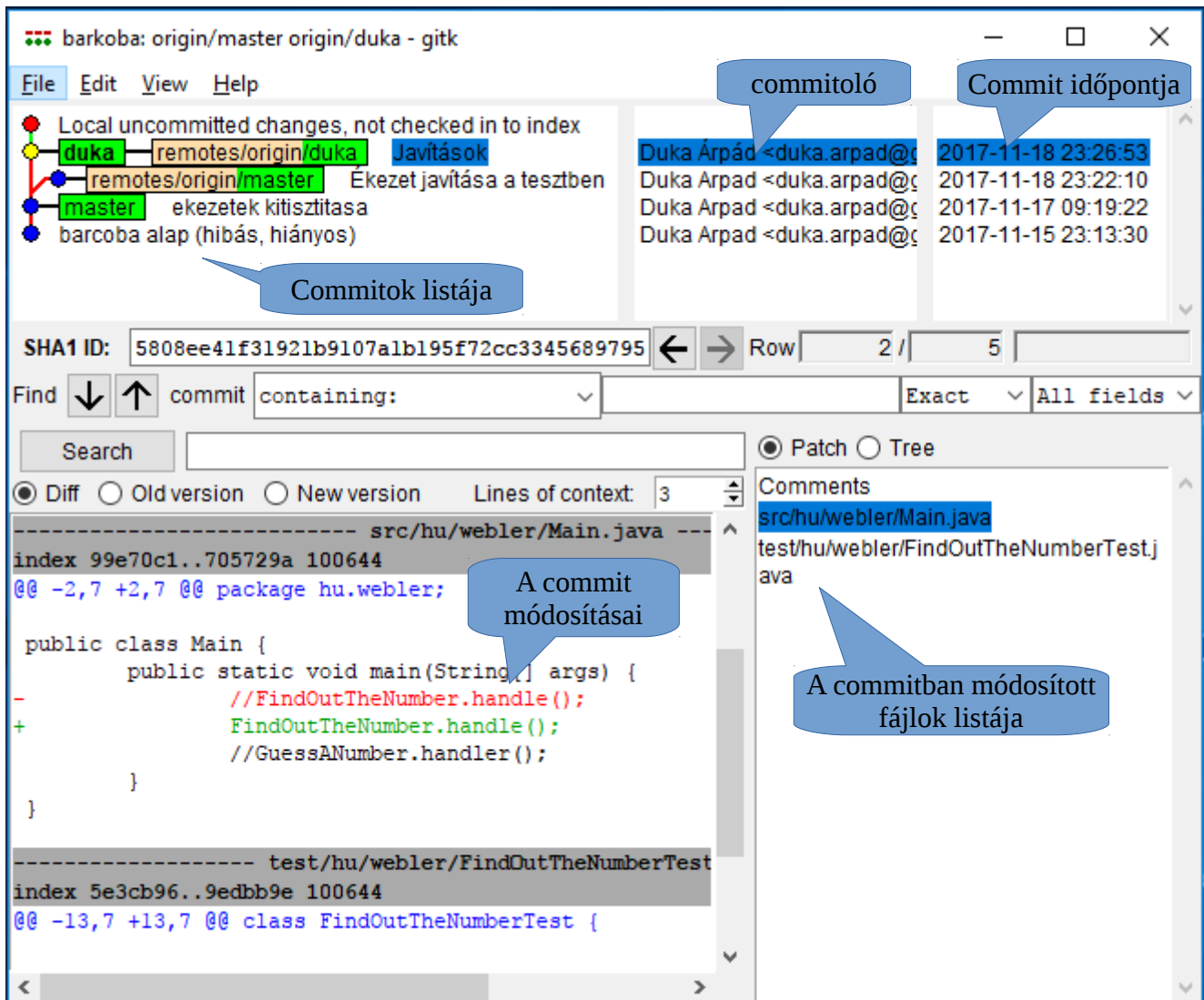
Itt már ki lesz jelölve az újonnan létrehozott branch-ünk neve, ezt nem is kell változtatni, nyomjuk meg ismét az ablak jobb alsó sarkában levő Push gombot. Ezt követően egy újabb felugró ablak a github-os bejelentkezési nevünket ill. jelszavunkat fogja kérni. Ha mindkettőt helyesen megadjuk és van is jogunk a kiválasztott repo-ba pusholni, a változtatásaink fel fognak tölteni a github szerverére.



Ha ez sikerült, a github-on is létrehoztunk egy új munka-branch-et. Ez az alap branch (master) módosításait érintetlenül hagyja.

Korábbi módosítások áttekintése

Amikor módosítási csomagokat állítunk össze, azok a saját projektünkben meg a github szerverén is egymás után sorban gyűlnek branch-enként. A korábbi commitok áttekintésére (mikor mi miről mire ki által változott) szintén rendelkezésünkre áll egy grafikus felület a **gitk**. Ezt szintén parancssorból indíthatjuk a projektünk könyvtárából.



A gitk alap esetben (ha nem adunk neki külön paraméterben utasítást), csak az aktuális branch commitjait fogja listázni. Ha más branch-eket is szeretnénk látni, akkor a megjelenítendő branch-eket soroljuk fel nek indításkor. Például a fent látható ablakot úgy indítottam, hogy a parancssorba **gitk duka master** parancsot írtam, így az aktuális (duka) branch mellett megjelenítette a master branch commitjait is.

Szintén gyakori és hasznos opció a gitknál az, amikor egy konkrét fájl vagy könyvtár módosításait szeretnénk listázni, vagyis csak azokat a commitokat, amik a érintet fájlt módosították. Ilyenkor a kérdéses fájl vagy könyvtár nevét kell megadnunk a gitk-nak indításkor paraméterként (pl.: **gitk src/hu/webler/GuessANumber.java**).