GIT drugi dvočas (kod se može videti na linku https://github.com/bojantomic/biblioteka2017)
PREDUSLOVI: Napraviti za svakog studenta po jednu granu u repozitorijumu pre časa

Objasniti zašto I kako se radi clone nekog udaljenog repozitorijuma (kloniranje sa GitHub-a), sa prethodnog časa. (File → Import → Git → Projects from git → Next), pa se izabere Clone URI → Next pa se ubaci https://github.com/bojantomic/biblioteka2017.git pa se ide na Next. Izaberu se sve grane, pa Next. Objasniti šta je origin, izabrati direktorijum unutar workspace foldera. Ići na Next par puta. Objasniti da je skoro sve sad podešeno osim user-a I email-a.

Napraviti klasu Biblioteka u paketu biblioteka koja implementira interfejs BibliotekaInterface koja ima Atribut knjige koji je lista obj. klase Knjiga i implementaciju njegovih metoda. Implementaciju uraditi bez logičke kontrole.

Dodati u indeks, (Team → Add to index). Probati commit, I videti obaveštenje koje izlazi.
Podesiti user I email Window → Preferences → Team → Git → Configuration, podesiti samo za repozitorijum. Uraditi commit. Pokazati strelicu nagore koja se pojavljuje pored naziva projekta (lokalni repo je jedan commit ispred udaljenog).

Napraviti novu granu "logickakontrola" na kojoj će se refaktorisati klasa Biblioteka (Team → switch to → new branch → logicka kontrola). Uvodi se logička kontrola u sve metode, ali se I refaktoriše metoda pronadjiKnjigu tako da u slučaju da nije ubačen nijedan kriterijum, metoda baca izuzetak.

```
if (autor==null && ISBN==0 && naslov==null && izdavac==null)
                    throw new RuntimeException("Morate uneti bar neki kriterijum za
pretragu");
```

Uraditi commit na ovoj grani.

Prebaciti se na master granu (Team → switch to → master). Refaktorisati metodu pronadjiKnjigu tako da se u slučaju da nije ubačen nijedan kriterijum, metoda vraća listu sa svim knjigama. Ubaciti kod na isto mesto kao I malopre na grani logickakontrola.

```
if (autor==null && ISBN==0 && naslov==null && izdavac==null)
                    return knjige;
```

Probati sad merge grane logicka kontrola u master granu (Team → merge → logickakontrola)
Objasniti na prozoru opcije "commit, no commit, squash". Objasniti šta se dešava, šta je konflikt, otvoriti u editoru I izmeniti kod. Posle izmena, uraditi (Team → add to index), pa commit.

Objasniti stvari koje se koriste da se stvari vrate unazad :)

Ako je POSLEDNJI commit bio pogrešan i treba ga zameniti – opcija ammend commit tj. ammend commit dugme u commit dijalogu. I da se to radi iskljucivo pre push-a tog commita.

Ako je neki od prethodnih commitova bio pogrešan i treba poništiti njegove efekte – iz history taba desni klik na željeni commit pa opcija revert.

Ako ne želite da commitujete izmene iz radnog direktorijuma od poslednjeg commita već da se direktno vratite na HEAD reviziju Team → reset (HARD opcija)

Uraditi push svega na GitHub.

Pokazati kako više ljudi radi na nekom projektu istovremeno
Otići na GitHub i za repo probagit postaviti kolaboratore.
Tražiti studentima da kloniraju bojantomic/probagit repo i da postave config za user i email.

Slučaj 1 – svi rade na master grani

Nastavnik promeni klasu Biblioteka, metodu pronadjiKnjigu – doda pretragu i po autoru. Uradi i commit za to, a i push. Studenti onda urade pull (Team → pull) i vidi se kako dobijaju nov kod, ako ništa nije još promenjeno kod njih.

Onda nastavnik uradi izmenu klase Biblioteka, metode pronadjiKnjigu (pretraga po autoru i ostalim kriterijumima), i uradim commit i push.

Studenti - svako od njih napravi klasu čiji je naziv njihovo ime i prezime, i uradi lokalno neki commit. Onda da probaju svi u isto vreme push. Objasniti da onda mora da se radi prvo pull pa onda auto merge, a push, i da to urade jedan po jedan. Na kraju svi da urade pull da imaju poslednju verziju koda. Pogledati onda history view u repozitorijumu. Objasniti da mogu da simuliraju ovu situaciju ako otvore dva github naloga i dve instance eclipse-a nad različitim workspace-ovima.

Slučaj 2 – svi rade na odvojenim granama

Svako napravi svoju lokalnu granu i prebaci se na nju (checkout as new local branch)

Nastavnik Uradi na master grani dodavanje logičke kontrole za klasu Knjiga i commit sa push-om. Kod studenata na grani nemaju ove promene, ali recimo da hoće da ih imaju. Onda urade rebase (Team → rebase (origin master)) i sad imaju promene koje je nastavnik commitovao.

Studenti nešto sad izmene kod sebe i commituju u lokalnu granu. Onda urade i push na remote granu.

Onda npr. nastavnik uradi merge jedne grane u master.