

# JAVA programavimo kalba

Git

# Kas yra GIT

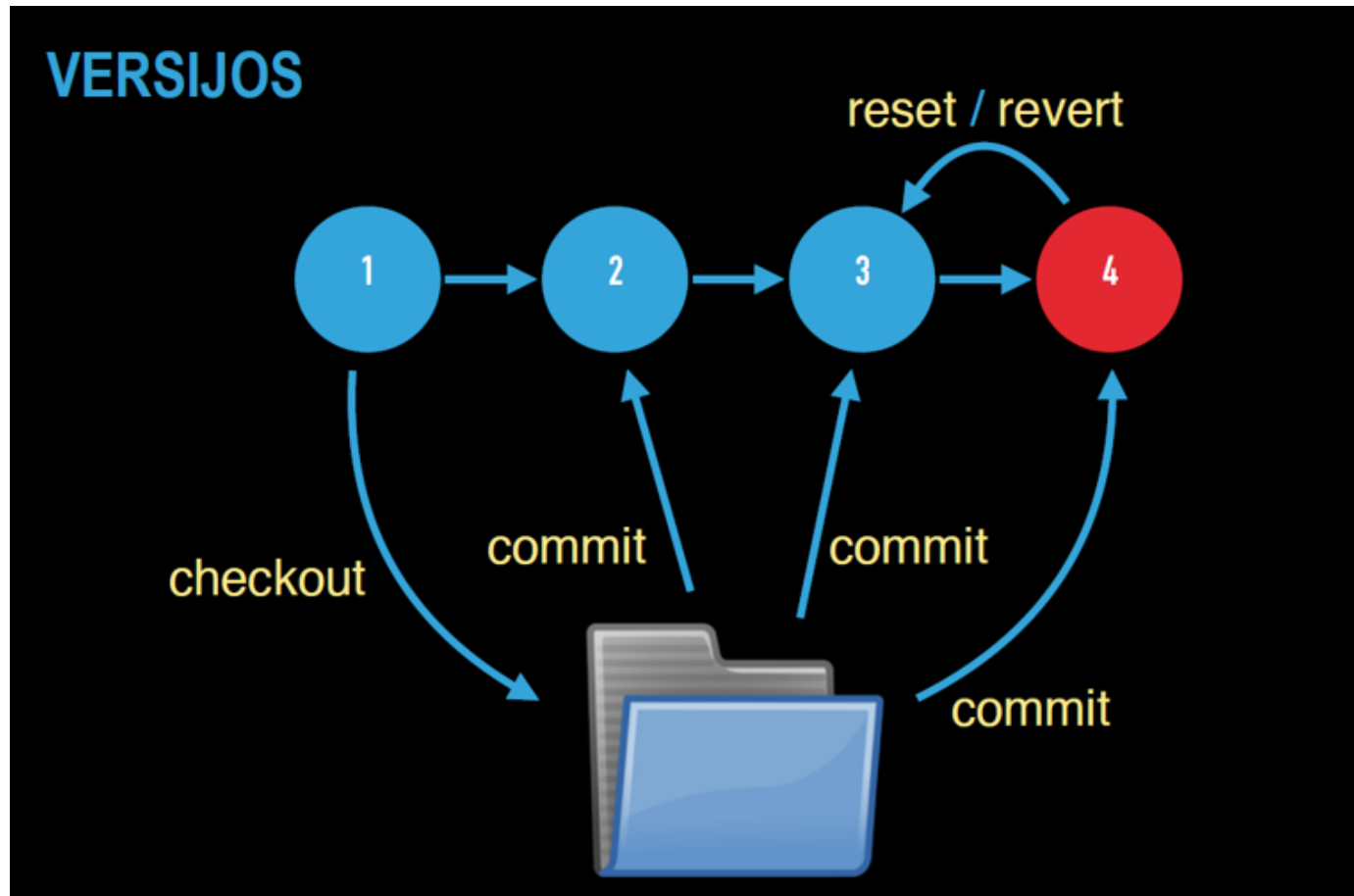
Tai programų arba tiesiog bet kokių tekstų versijų kontrolės sistema (version control system - VCS). Sukurta 2005 Linus Torvalds, kad padėtų didelėms programuotojų grupėms dirbti kartu kuriant ir toliau tobulinant Linux operacinės sistemos branduolį.

- Už GIT sistemos tolimesnį vystymą atsakingas Junio Hamano - <https://git-blame.blogspot.com/>
- GIT - a stupid content tracker

# Git saugykla (repository)

- Tarkime mes turime kažkokį projektą. Visi šitam projektui priklausantys failai (programos tekstai, moduliai, dokumentacija), jų pakeitimų istorija ir versijos sudaro visumą, kuri git sistemoje yra vadinama “saugykla (**repository**)”.
- Git susideda bent jau iš lokaliai (**local repository**) saugyklos esančios git naudotojo kompiuteryje, o taip pat ji gali ir turėti saugyklą kažkur kitame serveryje vadinama nutolusia saugykla (**remote repository**) - todėl apie git sakoma, kad tai paskirstyta (distributed) sistema.

# Versijos

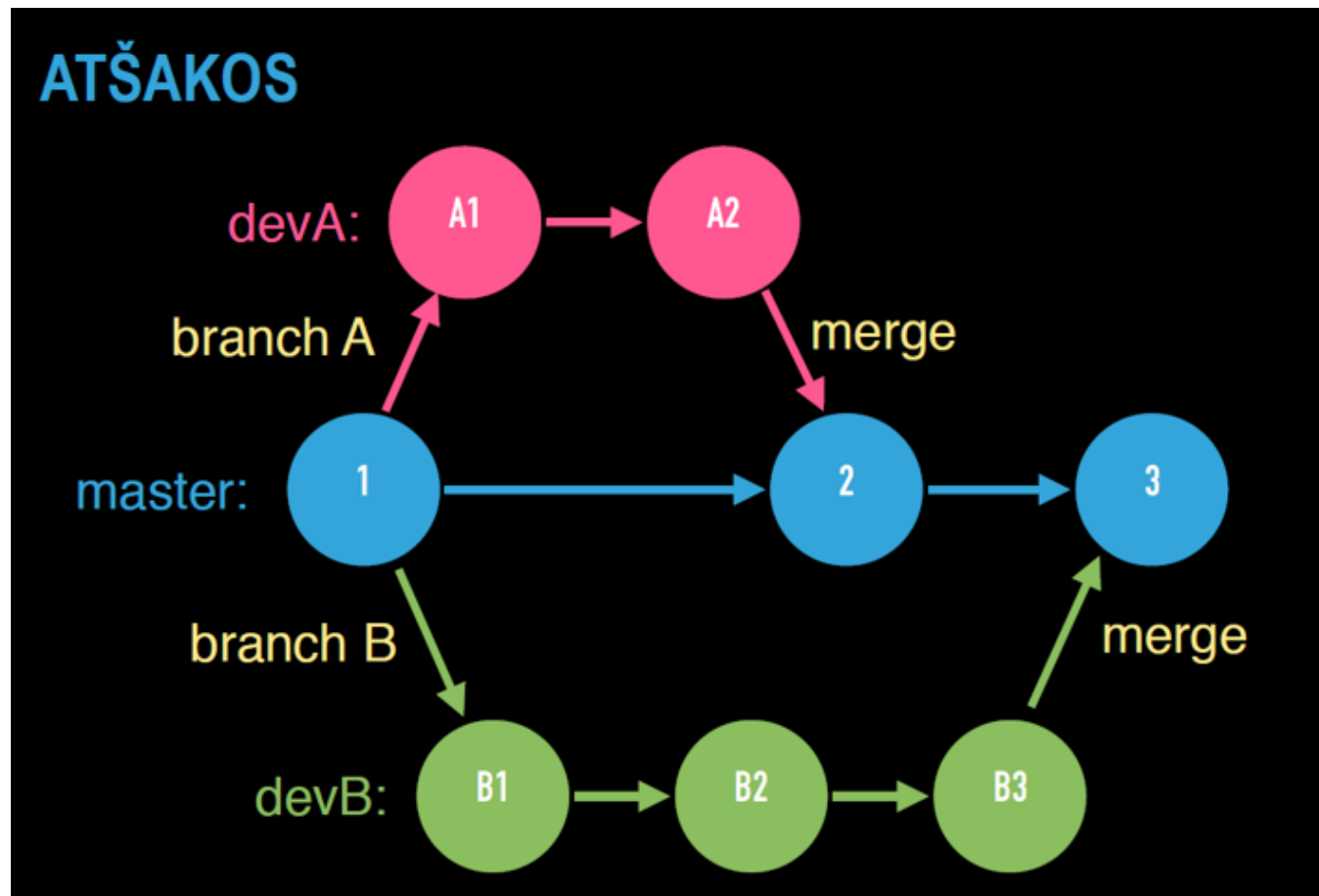


# Git atšakos ( branch )

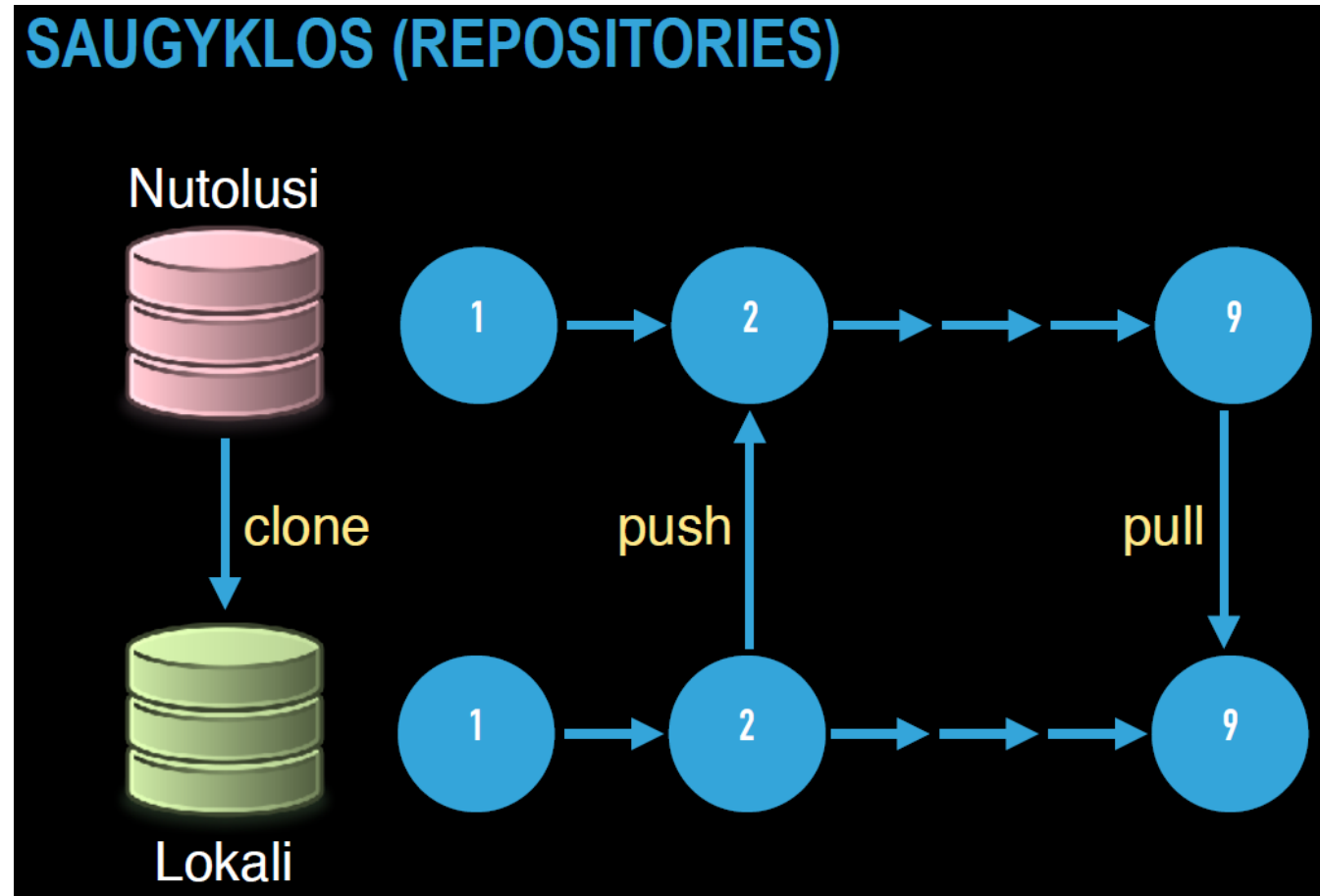
Atšakos (branches) - tai galimybė dirbti kartu keliems programuotojams vienu metu. Arba tam pačiam programuotojui dirbti vienu metu prie kelių skirtingų uždavinių.

- Saugykla pagal nutylėjimą visada turi bent vieną atšaką vardu **master**.
- **Master** atšaka sukurama git saugyklos sukūrimo metu.
- Paprastai **master** atšakoje laikomas ta kodo versija, kuri dabar realiai naudojama (ne testinė).
- Atšakos naudojamas tada kai norime dirbti prie kažkokių patobulinimų:
  - Kuriame naują atšaką nuo **master** (ar kitos) atšakos
  - Persijungiame į tą atšaką ir ten atliekame savo darbus
- Kai viskas atlikta ir viskas dirba be klaidų tai suliejame savo atšaką su **master** (ar kita) atšaka.

# Atšakos



# Saugyklos ( repositories )



# Failų būseną

Kiekvienas git lokalsios saugyklos failas gali būti vienoje iš trijų būsenų:

- užfiksuota (committed) - failo pakeitimai jau įrašyti į git saugyklą.
- modifikuota (modified) - failas modifikuotas bet dar neįrašytas į saugyklą. Jei failas nebus specialiai pažymėtas (staged), jis nebus įrašytas į saugyklą.
- ruošiamas (staged) - failas modifikuotas, dar neįrašytas į saugyklą, bet pažymėtas, kad bus įrašytas į saugyklą su sekančia commit komanda.
- p.s. jei mums reikia persijungti į kitą atšaką arba ištraukti atšakos failus iš nutolusios saugyklos tai reikia, kad visi mūsų lokalūs failai būtų committed.



# Git darbo principai

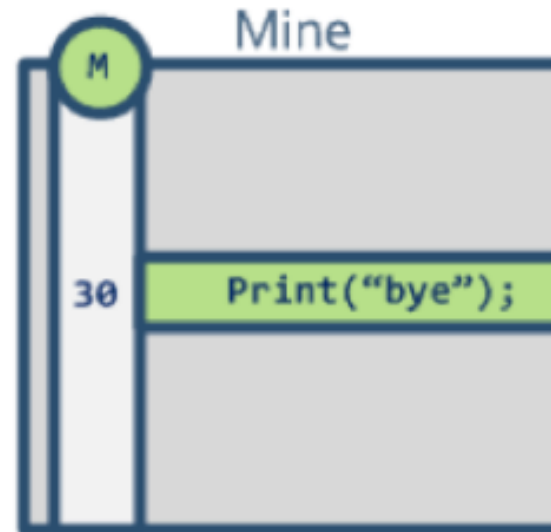
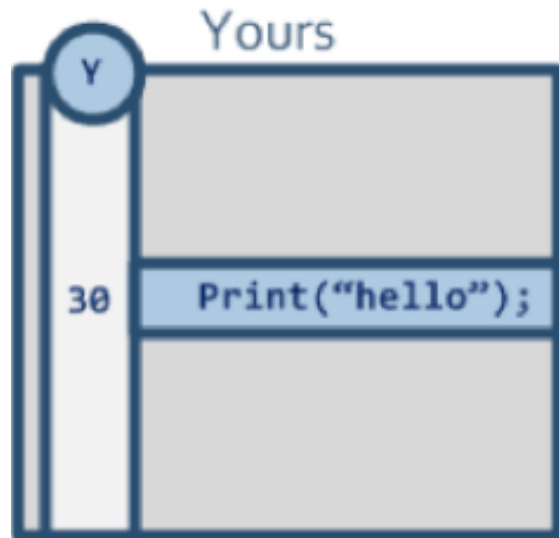
Kai persijungiame į reikiamą atšaką ir atkrauname jos failus, tai patenkame į taip vadinamą darbinę direktoriją ar medį (working directory/tree/area).

- Kai sukuriame naują failą, tai norint kad jis patektų į saugyklą, reikia jį pridėti į specialų sąrašą (staging area/index), t.y. į tokį sąrašą, iš kurio git sistema žino kokių failų pakeitimus reikia saugoti su patvirtinimo (commit) komanda.
- Visi jau egzistuojantys saugykloje failų taisymai automatiškai papuola į tą sąrašą.
- Mes galime pasirinkti kokių ir kokie failų pakeitimai papuls į lokalią saugyklą su sekančia saugojimo komanda.

# 2-way merge

Tarkime **aš** ir kažkas **kitas** modifikuojame tą patį failą (jo kopiją) vienu metu:

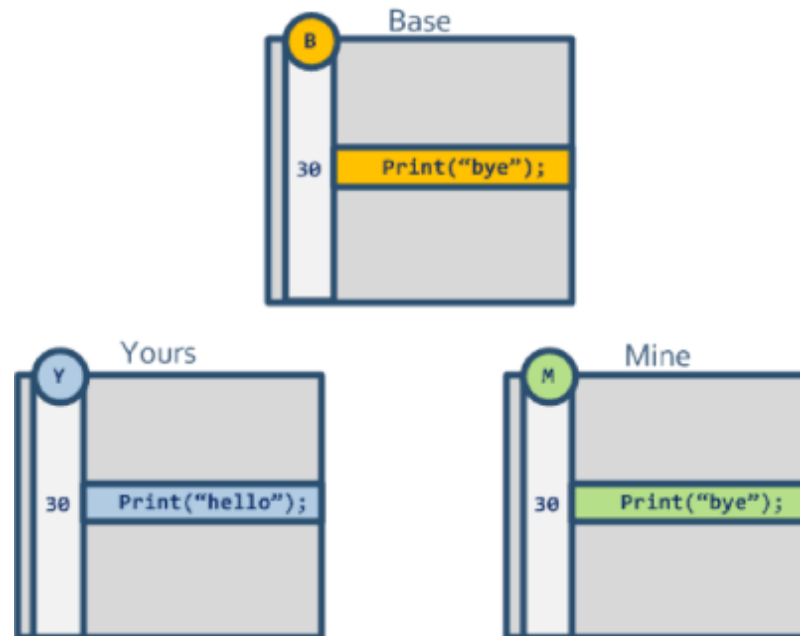
- Ir dabar kažkas trečias turi abi failo kopijas ir mato, kad failai skiriasi 30 eilutėje. Kaip žinoti kas pakeitė 30 eilutę - **aš** ar **kitas**? O gal mes abu keitėme tą eilutę?



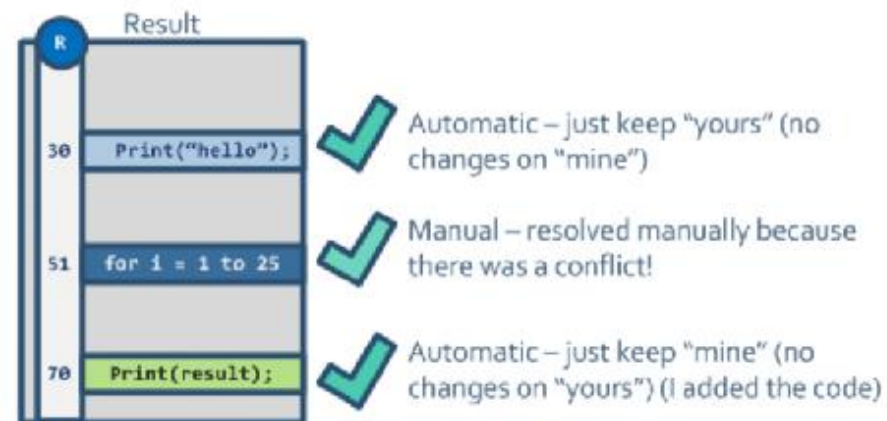
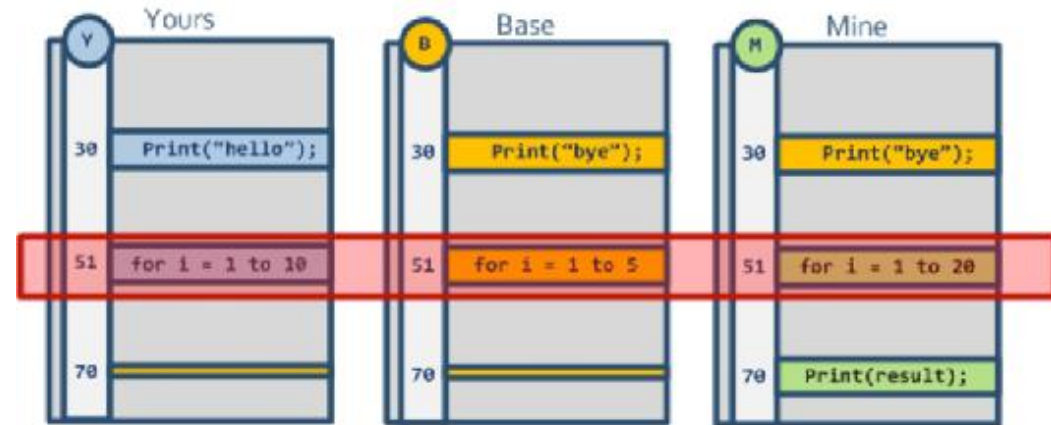
# 3-way merge

Viskas pasidaro žymiai paprasčiau, jei turime pradinį failą (base).

- Dabar trečiasis aiškiai mato, kad pakeitimą padarė **kitas** ir gali automatiškai sulieti.
- Tas bazinis (base) failas yra pats artimiausias bendras protėvis mano ir kito failui.



# Konfliktai



# Git tag

Kiekvienas git commit tur tam tikrą vardą ar id, bet pagal jį nėra patogiu orientuotis.

- Tam tikslui galima naudoti taip vadinamus git tag'us.
- Tokie užtaginti commit'ai žymiai lengviau randami ir ypač naudingi pažymint naujos versijos ar subversijos išleidimą.

# Git įdiegimas

- Visą git sistemą galima įdiegti iš čia: <https://git-scm.com/download/>
- Kompiuteriuose su Mac OS X ar Linux git greičiausia jau bus įdiegtas
- Tame pat puslapyje yra nuoroda į programas padedančias dirbti su git grafinėje aplinkoje

# Uždaviniai

1. Įdiekite, jei reikia, git ir SourceTree savo kompiuteryje
2. Užsiregistruokite github'e
3. Sukurkite github'e repositoryų
4. Klonuokite jį savo kompiuteryje
5. Sukurkite failą ir užkelkite jį į github'ą

# Nuorodos

1. <https://github.com>
2. <https://bitbucket.org>
3. <https://backlog.com/git-tutorial/en/>
4. <https://git-scm.com/book/en/v2>
5. <https://www.atlassian.com/git/tutorials>
6. <http://www.drdobbs.com/tools/three-way-merging-a-lookunder-the-hood/240164902>
7. <https://git-blame.blogspot.com>