Varna sočasnost [SMAP:4]

- *angl.* concurrency-safe, thread safe
- vse programske strukture in metode ter s tem tudi paketi niso prilagojeni za sočasno izvajanje
- lahko prihaja do tveganih stanj
- če so programske strukture in metode deklarirane kot varne za sočasnost, ne bo prišlo do tveganih stanj
- slovar-1.go
  - uporabimo strukturo slovar (`map`), ki ni varna za sočasnost
  - če vse sočasne gorutine samo berejo, je vse v redu
  - če ena gorutina piše in druga bere, pride do tveganega stanja
  - če dve gorutini pišeta, pride do tveganega stanja
- slovar-2.go
  - lahko uporabimo bralno-pisalno ključavnico
- slovar-3.go
  - uporabimo posebno strukturo `sync.Map`
- paket `math/rand`
  - dostop do strukture `rand` iz paketa `math/rand` je v jeziku go zaščiten s ključavnicami, ne moremo pa določiti semena (pi-11.go)
  - nastavljanje semena - neučinkovita rešitev pi-12.go
    - strukturi tipa `*rand.Rand` , ki jo ustvarimo sami, lahko določimo seme, pri hkratni uporabi v več gorutinah pa moramo dostope varovati s ključavnicami
    - pri uporabi globalne spremenljivke rnd ne moremo zagotavljati ponovljivosti rezultatov pri danem semenu in številu gorutin
  - dobra praksa pri delu s psevdonaključnimi števili pi-13.go
    - če želimo učinkovito generirati psevdonaključna števila, v vsaki gorutini naredimo lokalno strukturo tipa `*rand.Rand`
    - lokalni strukturi lahko nastavimo seme
    - rezultat je pri enakem številu gorutin ponovljiv