

# Projektovanje softvera

Fasada



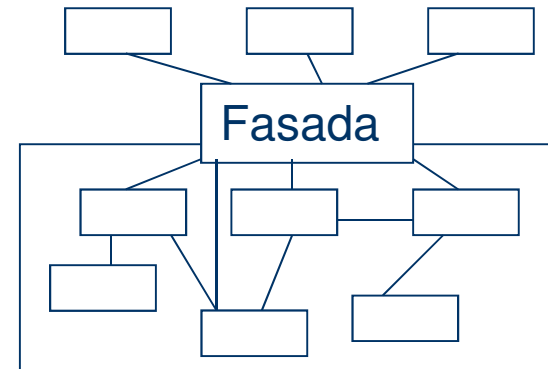
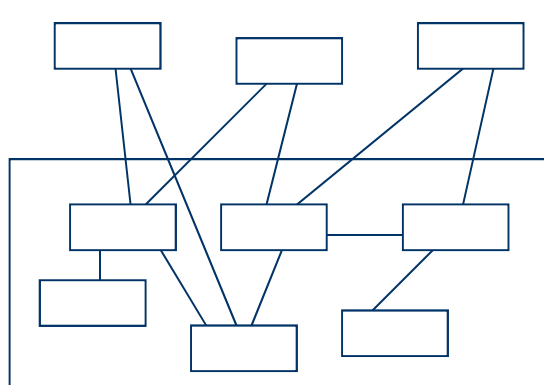
# Fasada (1)

- Ime i klasifikacija:
  - Fasada (engl. *Facade*) – objektni uzorak strukture
- Namena:
  - pružanje jedinstvenog interfejsa podsistema
  - definisanje interfejsa višeg nivoa da bi se podsistem lakše koristio

# Fasada (2)

- Motivacija:

- strukturiranje sistema u podsisteme
  - pojednostavljuje proces projektovanja (i održavanja)
- cilj je da se međuzavisnosti podsistema svedu na minimum
- uvođenje objekta “fasada” ide u pravcu ovog cilja
  - fasada nudi uprošćeni interfejs za opšte upotrebe podsistema



Fasada

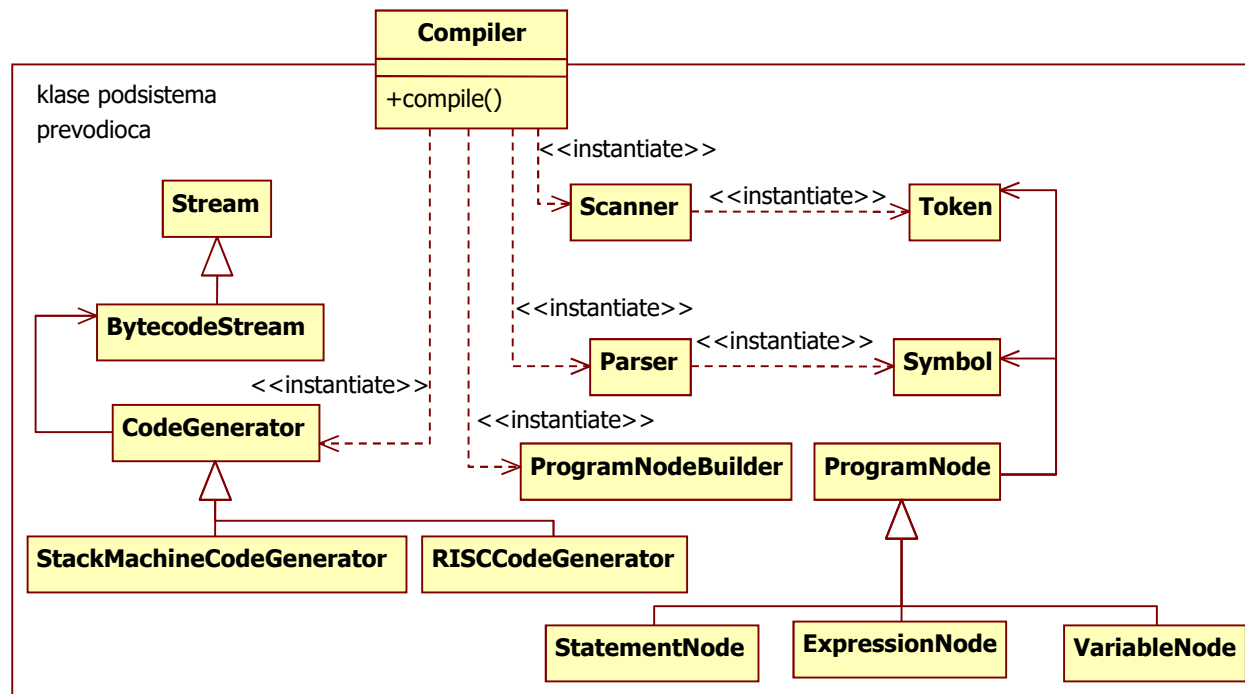
22.11.2023.

# Fasada (3)

- Motivacija (nastavak):
  - fasada ne brani direktan pristup klasama unutar podsistema
  - primer: podsistem prevodioca:
    - klase: `Scanner`, `Parser`, `ProgramNode`, `BytecodeStream`, ...
    - klasa `Compiler` ostvaruje opštenamenski interfejs podsistemu
      - ona predstavlja fasadu podsistema prevodioca
      - ona većini klijenata pruža dovoljnu uslugu (kompletno prevođenje)
        - olakšava se posao većini programera
      - ona ne skriva ostale klase koje pružaju servise nižeg nivoa
        - ne sprečava se pristup pojedinim klasama unutar podsistema
        - pristup drugim klasama potreban je samo specijalnim aplikacijama

# Fasada (4)

- Motivacija (nastavak):

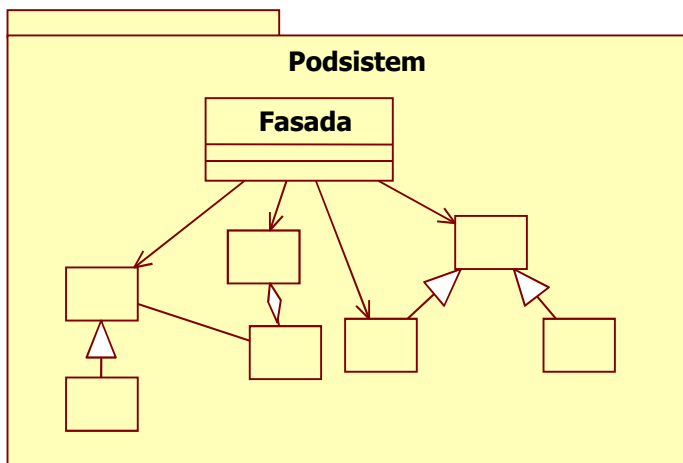


# Fasada (5)

- Primenljivost: uzorak treba koristiti kada
  - potrebno je dati jednostavan interfejs složenom podsistemu
    - složenost podsistema raste tokom razvoja
    - primena uzoraka pospešuje porast broja manjih klasa
    - podsistem postaje teži za upotrebu klijentima
    - većini klijenata je dovoljan interfejs podsistema preko fasade
  - postoje brojne zavisnosti između klijenata i klasa podsistema
    - fasada razdvaja podsistem od klijenata i drugih podsistema
  - postoji potreba da se rasloje podsistemi
    - fasada može da se koristi za ulaznu tačku svakog nivoa podsistema

# Fasada (6)

- Struktura:



- Učesnici:

- Fasada (klasa `Compiler`)
  - zna koje klase podsistema su odgovorne za koji zahtev
  - delegira zahteve klijenata odgovarajućim objektima podsistema
- klase podsistema
  - implementiraju funkcionalnosti
  - izvršavaju zahteve fasade
  - ne znaju ništa o fasadi

- Saradnja:

- klijenti šalju zahteve fasadi
- fasada prosleđuje zahteve objektima podsistema

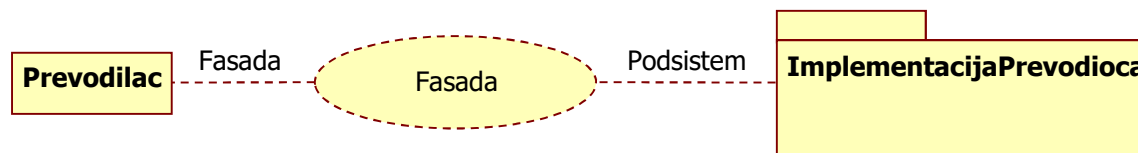
# Fasada (7)

- Posledice:
  - smanjivanje broja objekata od kojih klijenti zavise
    - olakšava korišćenje podsistema
  - slabo vezivanje između podsistema i klijenata
    - olakšava održavanje podsistema
      - promena klasa podsistema ne utiče na klijenta
      - promena u podsistemu ne zahteva ni ponovno prevođenje klijenta
    - povećava portabilnost (pod)sistema
      - izmene u podsistemu zbog prenosa ne utiču na druge podsisteme
  - uzorak ne sprečava korišćenje ostalih klasa podsistema



# Fasada (8)

- UML notacija:



- Povezani uzorci:

- *Apstraktna fabrika* predstavlja Fasadu za kreiranje objekata podsistema
- potreban je samo jedan objekat Fasade, pa se realizuje kao *Unikat*
- *Posrednik* je sličan *Fasadi* samo što on apstrahuje proizvoljne komunikacije između objekata (podsistema), a *Fasada* apstrahuje interfejs podsistema
  - objekat `Posrednik` može da dodaje funkcionalnost
  - objekti podsistema znaju za objekat `Posrednik`
  - objekti podsistema ne znaju za objekat `Fasada`