

Projektovanje softvera

Prototip

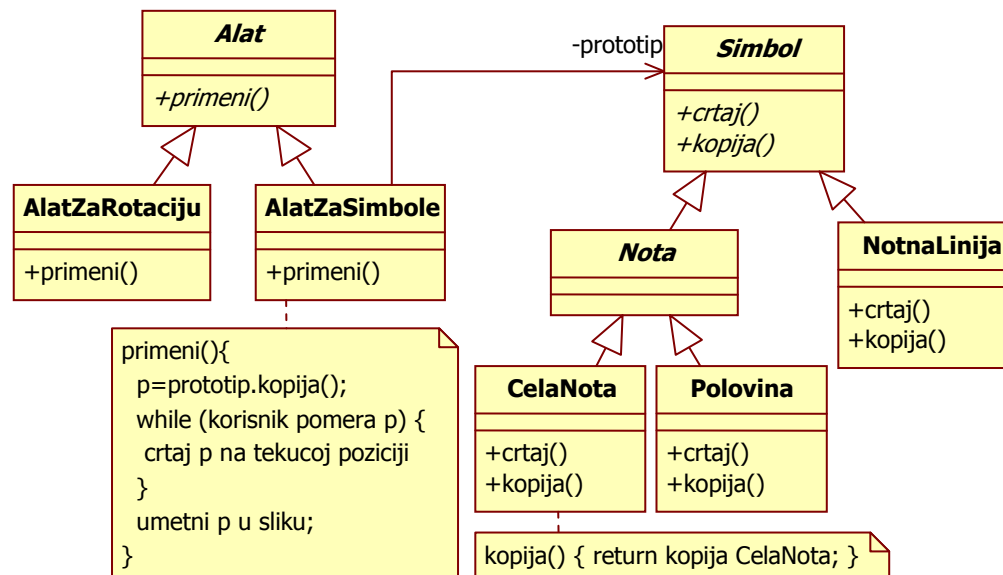


Prototip (1)

- Ime i klasifikacija:
 - Prototip (polimorfna kopija, engl. *Prototype*) – objektni uzorak kreiranja
- Namena:
 - specificira vrste objekata koje se kreiraju korišćenjem prototipske instance
 - kreira nove objekte kopiranjem prototipa
- Motivacija:
 - editor za muzičke partiture bi se mogao realizovati:
 - prilagođenjem opšteg radnog okvira za grafičke editore
 - dodavanjem novih objekata koji predstavljaju note, pauze i druge muzičke simbole
 - radni okvir editora može imati paletu alata za dodavanje muzičkih objekata partituri
 - paleta bi takođe uključila i alate za selektovanje, pomeranje i druge radnje nad objektima
 - korisnik bi kliknuo na alat "čtvrtinka" i dodao ovu u partituru
 - korisnik bi mogao da koristi alat za pomeranje da premesti notu na drugu notnu liniju
 - radni okvir nudi
 - apstraktnu klasu `Simbol` za grafičke komponente (npr. note, linije)
 - apstraktnu klasu `Alat` za definisanje alata kakvi su u paleti
 - klasu `AlatZaSimbole`, potklasu klase `Alat`, za kreiranje i dodavanje objekata u dokument

Prototip (2)

- Motivacija (nastavak):

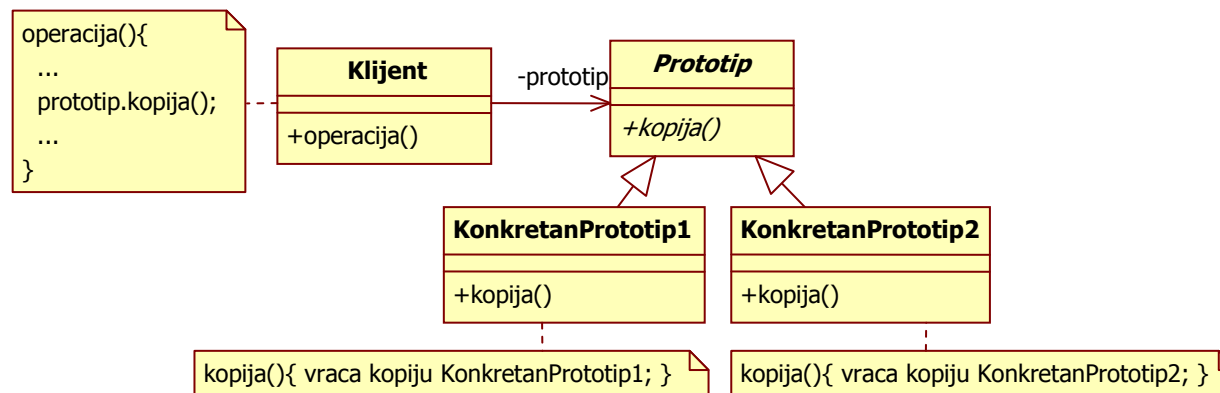


Prototip (3)

- Motivacija (nastavak):
 - problem za projektanta radnog okvira je klasa `AlatZaSimbole`
 - klase za note i notne linije su specifične za aplikaciju, nepoznate radnom okviru
 - mogla bi se praviti potklase `AlatZaSimbole` za svaki muzički objekat, ali ima ih mnogo
 - potklase bi se razlikovale samo po vrsti muzičkog objekta koje stvaraju
 - kompozicija objekata je česta fleksibilna alternativa izvođenju i nasleđivanju
 - radni okvir parametrizuje primerke `AlatZaSimbole` objektima klase `Simbol`
 - rešenje:
 - `AlatZaSimbole` kreira novi `Simbol` kopiranjem objekta potklase `Simbol`
 - dotični objekat (primerak potklase `Simbol`) se naziva prototipom
 - objekat `AlatZaSimbole` je parametrizovan prototipom koji klonira i dodaje u dokument
 - potrebno da sve potklase `Simbol` imaju operaciju `kopija()`
 - u muzičkom editoru, alati za kreiranje muzičkih obj. su objekti `AlatZaSimbole`
 - dotičan objekat `AlatZaSimbole` se inicijalizuje različitim prototipom
 - svaki objekat `AlatZaSimbole` proizvodi muzički objekat kloniranjem njegovog prototipa

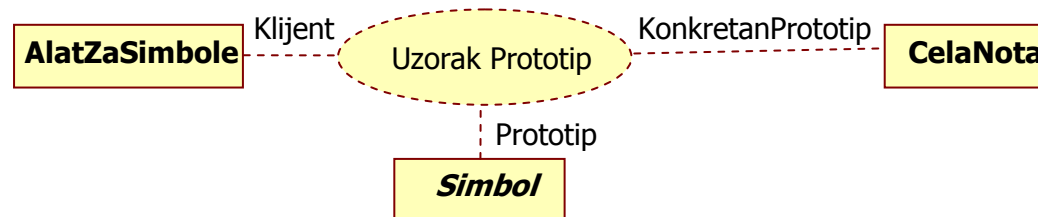
Prototip (4)

- **Primenljivost: uzorak treba koristiti**
 - kada sistem treba da bude nezavisan od toga kako se njegovi proizvodi kreiraju i predstavljaju
 - kada se klase specificiraju u vreme izvršenja, npr. dinamičkim učitavanjem
 - kada treba izbeći izgradnju hijerarhije fabrika paralelno sa hijerarhijom proizvoda
- **Struktura:**



Prototip (5)

- Učesnici:
 - Prototip (klasa Simbol)
 - deklarirše interfejs za sopstveno kloniranje
 - KonkretanPrototip (klase CelaNota, Polovina, NotnaLinija)
 - implementira operaciju za sopstveno kloniranje
 - Klijent (klasa AlatZaSimbole)
 - kreira novi objekat zahtevom prototipu da se klonira
- Saradnja:
 - klijent zahteva od prototipa da se klonira
- UML notacija:



Prototip (6)

- Posledice:
 - *Prototip* ima sledeće dobre strane:
 - smanjivanje potrebe izvođenja
 - dodavanje i uklanjanje proizvoda u vreme izvršenja (registracija prototipske instance)
 - dinamičko konfigurisanje aplikacije klasama
 - redukovanje broja imena koje klijent treba da poznaje
 - loša strana uzorka *Prototip*:
 - svaka potklasa mora implementirati operaciju `kopija()` što je komplikovano
 - ako klasa već postoji
 - ako objekat klase sadrži (pod)objekte koji ne podržavaju kopiranje
- Povezani uzorci:
 - *Apstraktna Fabrika* je alternativan uzorak, ali se može i dopuniti sa *Prototipom*:
 - fabrika može sadržati skup prototipova na osnovu kojih klonira i vraća proizvode
 - često zajedno sa uzorcima *Kompozicija (Sastav, Sklop)* i *Dekorater (Dopuna)*