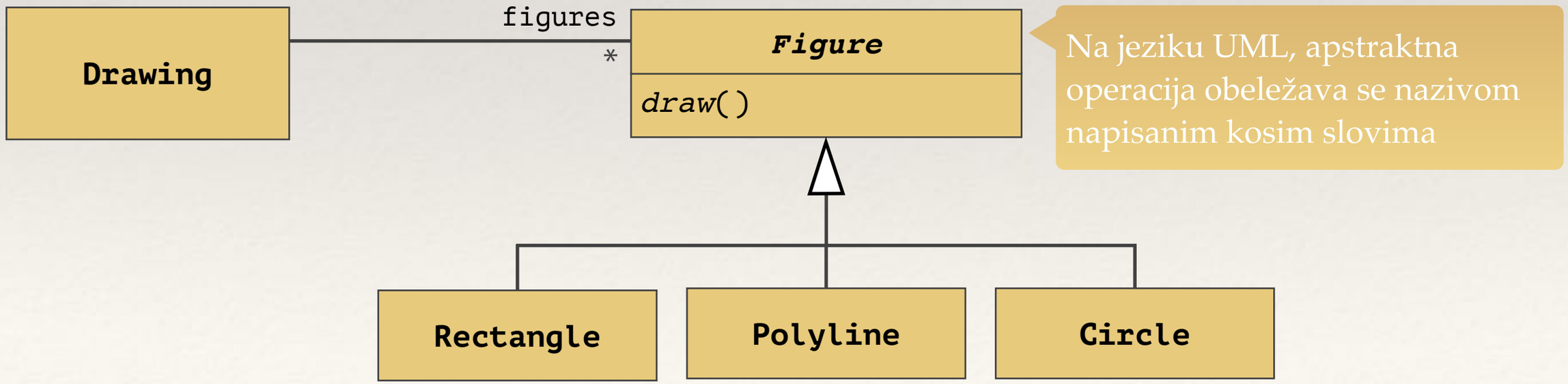
Hijerarhijska dekompozicija

- Kako treba da izgleda implementacija operacije (metoda) *draw* klase *Figure*? Može li se nacrtati figura koja "nije ništa posebno" (a takva i ne postoji, jer je ova klasa apstraktna)?
- U nekim slučajevima, polimorfna operacija apstraktne klase se može implementirati u toj klasi:
  - kada ova operacija ima neko logično podrazumevano ponašanje, makar i prazno
  - ona je pomoćna (*helper*) metoda koja se može, ali ne mora redefinisati; recimo, opciono proširenje podrazumevano praznog koraka neke složenije metode osnovne klase
- Ako to nije slučaj, operacija je *apstraktna* (*abstract operation*): predstavlja samo *specifikaciju usluge* koja se može tražiti od objekata, ali ne daje *implementaciju* te usluge (metodu); tu implementaciju daće izvedene klase



Hijerarhijska dekompozicija

- Na jeziku C++ apstraktne operacije nazivaju se *čisto virtuelnim funkcijama* (*pure virtual functions*) i označavaju specifikatorom = *0* u potpisu funkcije; takva funkcija neće imati definiciju:

```
class Figure {
public:
  virtual void draw (Viewport*) = 0;
  …
};

class Circle : public Figure {
public:
  virtual void draw (Viewport*) override;
  …
};
void Circle::draw (Viewport* vp) {
  …
}
```

- Na jeziku C++, ako klasa ima bar jednu čisto virtuelnu funkciju, onda je ona za prevodilac apstraktna - neće dozvoliti kreiranje direktnih instanci te klase (bez obzira na dostupnost konstruktora)
- Konceptualno, obrnuto ne važi: klasa može biti apstraktna i ako nema nijednu apstraktnu operaciju, mada prevodilac za jezik C++ ne tretitra posebno tu klasu kao apstraktnu (u užem smislu, prema pravilima jezika)
- Izvedena klasa može, a ne mora da definiše metodu za apstraktnu operaciju koju nasleđuje; ako je ne definiše, ta operacija i dalje ostaje apstraktna, a klasa je takođe apstraktna