

**OOP**

Apstraktne klase i  
Polimorfizam

# U ovom poglavlju naučit ćete

---

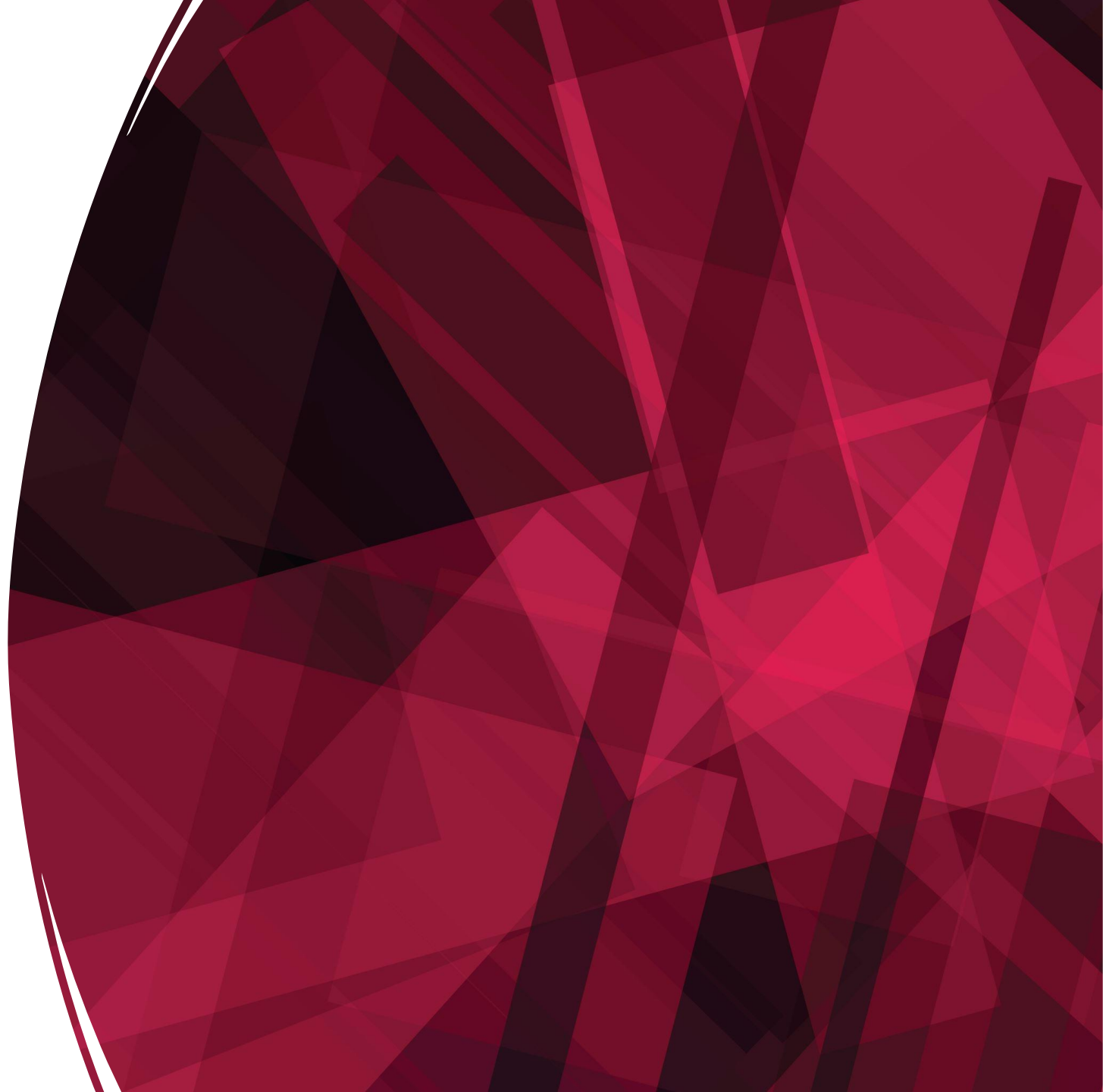
- Apstraktne klase i metode
- Virtualne metode
- Polimorfizam



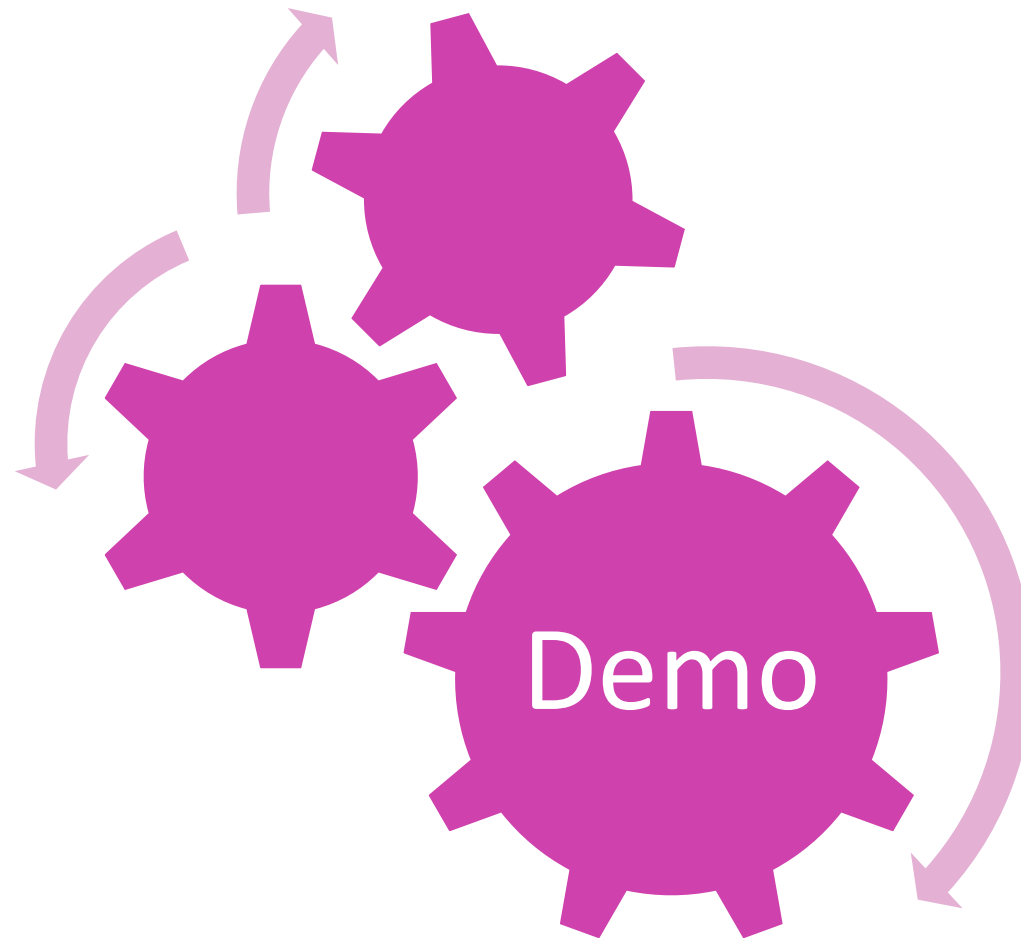
# Polimorfizam

---

- Vrlo korisna osobina nasljeđivanja je mogućnost ponovne upotrebe kôda
- Međutim, još jedan jako važan aspekt nasljeđivanja je **polimorfizam**. (grčki, *poli* = mnogo, *morf* = oblik).
- Polimorfizam predstavlja mogućnost korištenja više oblika tipa, bez obzira na njihove pojedinosti.
- Intenzivno korištenje virtualnih metoda (ključna riječ virtual)



# Polimorfizam



# Apstraktne klase

- Virtualne metode u osnovnoj klasi koristimo pod pretpostavkom da će izvedene klase implemetirati njihove vlastite inačice, no to nije obavezno
- Želimo li izvedenu klasu obvezati na vlastitu implementaciju neke metode osnovne klase, trebamo tu metodu označiti kao **apstraktnu**, navođenjem ključne riječi **abstract**:
- `abstract public void Detalji();`

# Apstraktne klase...

- Apstraktna metoda **nema implementaciju** pa se nakon deklaracije ne stavljaju vitičaste zagrade, već samo točka-zarez (;)
- Označavanje bilo koje metode u klasi apstraktnom, automatski i cijelu klasu čini **apstraktnom** pa se i deklaracija klase također mora označiti ključnom riječi **abstract**, kao u sljedećem primjeru:
  - `abstract public class SveucilisnaOsoba`

# Zapečaćene klase

- Suprotan pojam od apstraktnog je **zapečaćeno** (engl. *sealed*).
- Namjera apstraktne klase je biti naslijeđena, a zapečaćena klasa ne dopušta de se iz nje izvode druge klase.
- Ključna riječ **sealed**:
- `sealed public class SveucilisnaOsoba`

# Klasa Object

- Sve klase i strukture programskog jezika C# izvode iz klase System.Object
- Osnovna klasa je neposredan "roditelj" izvedenoj klasi
- Izvedena klasa može biti osnovna daljnjim izvedenim klasama stvarajući tako **stablo nasljeđivanja** ili *hijerarhiju*.
- Najviša klasa u hijerarhiji nasljeđivanja naziva se **korijenska klasa**, a u C#-u to je klasa Object.



**Hvala na pažnji!**