

**OOP**

Nasljeđivanje

# U ovom poglavlju naučit ćete

---

Nasljeđivanje

# Uvod

---

- *specijalizacija* - u C#-u implementira se pomoću *nasljeđivanja*.



# Nasljeđivanje

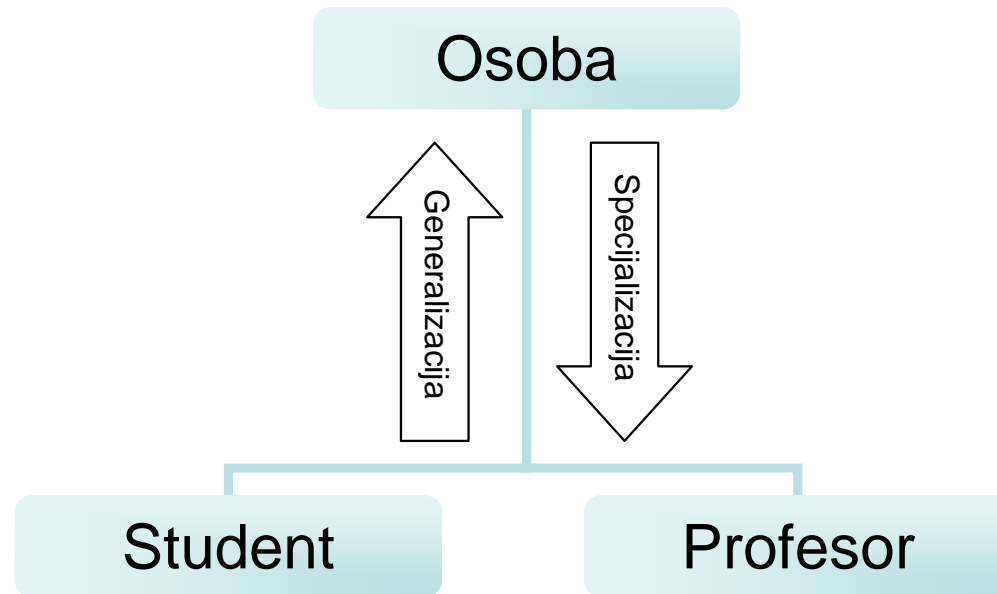
- Klase i njihove instance (objekti) nalaze se u raznim manje ili više složenim međusobnim odnosima i ovisnostima.
- Relacija je = izvor nasljeđivanja

# Nasljeđivanje...

- Student i Profesor su osobe
- Student i profesor imaju neke zajedničke osobine (opće osobine osobe) ali se razlikuju po svojim specijaliziranim osobinama.
- Kažemo da su i student i profesor **specijalizacije** osobe.

# Nasljeđivanje...

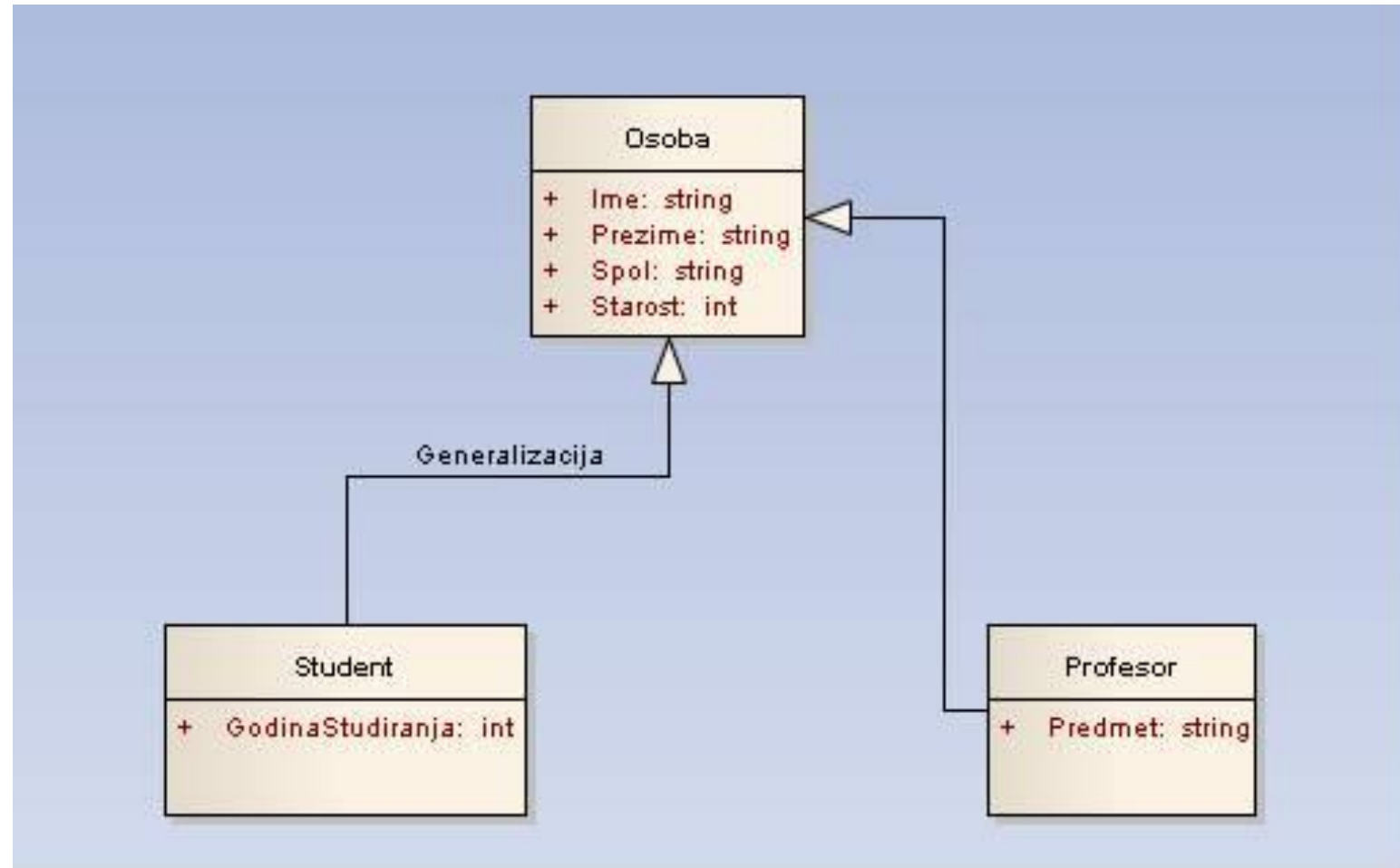
---



# UML

- Standardizirani jezik za opis sustava ili načina poslovanja je **UML** (engl. *Unified Modeling Language*).
- Dio UML-a sadrži dijagrame koji se koriste za dokumentiranje odnosa između klasa.

# UML...





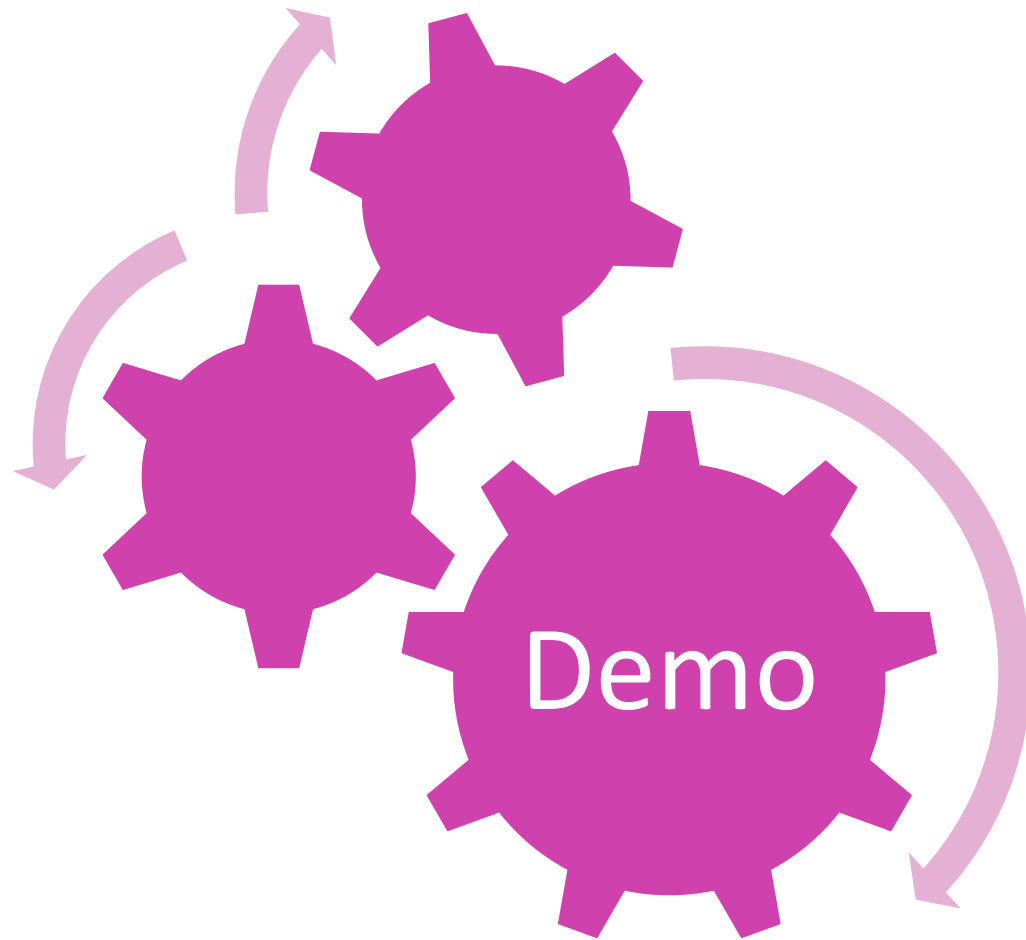
# Nasljeđivanje u C#-u

- Izjava da klasa Student **nasljeđuje iz (izvodi iz)** klase Osoba označava da je riječ o specijalizaciji klase Osoba.
- To zapravo znači da klasa Student nasljeđuje sve osobine i ponašanja klase Osoba i zatim se specijalizira prema vlastitim potrebama.
- U tom kontekstu klasa Osoba naziva se **osnovnom** ili **baznom** klasom, a klasa Student **izvedenom** ili **naslijeđenom** klasom.

# C# - primjer

```
class Student : Osoba
{
    // definicija ostalih
    (specijaliziranih) članova
}
```

# Demo – nasljeđivanje



**Hvala na pažnji!**