

# Programiranje za umetnike 2

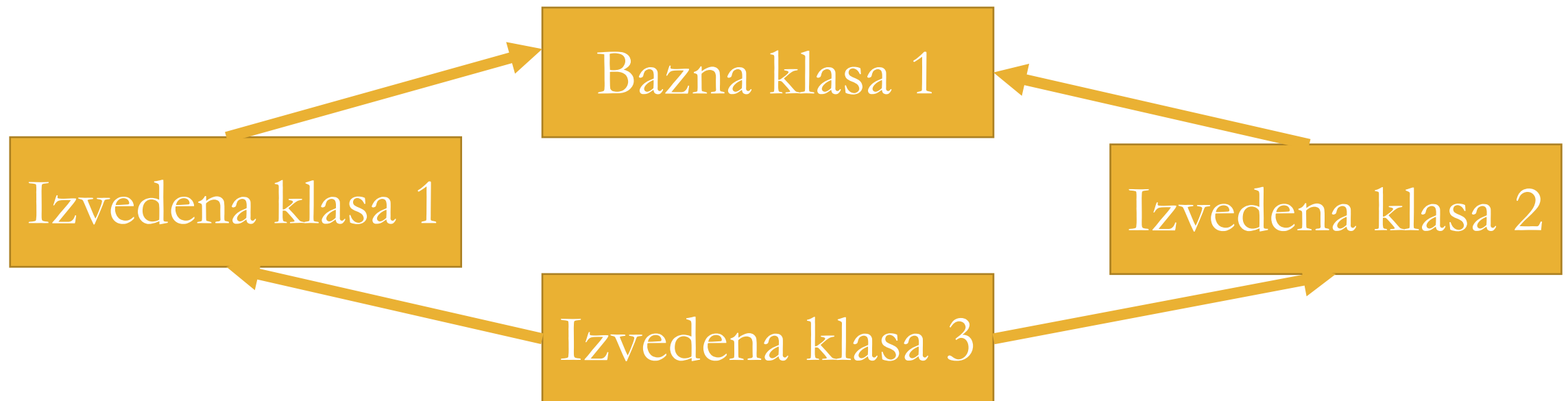
~ 4 ~

---

Staša Vujičić Stanković

# Hibridno nasleđivanje

- Hibridno nasleđivanje obuhvata sve vrste nasleđivanja o kojima smo do sada govorili – direktno, u više nivoa, hijerarhijsko, višestruko.



# Skrivanje (enkapsulacija) podataka

---

- Enkapsulacija (skrivanje podataka) označava ograničeni pristup metodama i atributima. Ovim se između ostalog sprečava da vrednosti podataka budu greškom izmenjene.
- Članica izvedene klase (atribut ili metod) može da ima isto ime kao i članica bazne klase, čime članica bazne klase biva skrivena.
- Svaka upotreba tog imena odnosi se na članicu izvedene klase.
- Članici bazne klase se pristupa pomoću `super().imeČlanice`.

# Skrivanje (enkapsulacija) podataka u Python-u

---

- I. Članice koje nemaju prefiks sačinjen iz donjih crta (primer: `a`, `b`) se smatraju javnim i može im se pristupiti iz bilo kog dela koda (i iz same klase i iz spoljnih klasa).
- II. Članice čiji identifikatori počinju jednom donjom crtom su takozvane zaštićene članice i njima je moguće pristupati unutar bazne klase i nasleđenih klasa, ali im nije moguće pristupati iz spoljnih delova koda (primer: `_a`, `_b`).
- III. Članice čiji identifikatori počinju dvema donjim crtama su takozvane privatne članice i njima je moguće pristupati samo unutar klase i nije im moguće pristupati iz spoljnih delova koda (primer: `__a`, `__b`).

# Pristup privatnim atributima preko javnih metoda

---

```
# ...
```

```
def vratiA():  
    return self.a
```

```
def pisiA(a):
```

```
# proverava ispravnosti  
vrednosti a koju zelimo da  
dodelimo
```

```
    if a < 0:
```

```
        return
```

```
    self.a = a
```

```
# ...
```

Pogledajmo na primeru kako se za pristup, odnosno ažuriranje vrednosti privatih atributa koriste tzv. getteri i setteri.

# Polimorfizam

---

- Polimorfizam znači sposobnost uzimanja različitih oblika.
- To je važan koncept kod nasleđivanja, odnosno kod roditeljskih i izvedenih klasa.

Polimorfizam se u Python-u primenjuje preko:

- redefinisanja metoda (engl. overriding) i
- preopterećenja metoda (engl. overloading).

# Redefinisanja metoda (engl. overriding)

---

- Metod jedne klase može da se ponaša različito od istoimenog metoda druge klase.

# Preopterećenje metoda (engl. overloading)

---

- Omogućava definisanje „fleksibilnih“ metoda tako da metod može da se pozove samo sa nekim od argumenata i nema potrebe da se navode drugi argumenti (a po potrebi može da se pozove i sa svim argumentima).
- Postoje i mehanizmi da se izvrši „overloading“ operatora, tj. da se u samoj klasi definiše kako će se ponašati objekti te klase kada se nad njima primene Python operatori.



# Specijalne metode

---

- `__init__` - konstruktor metoda
- `__str__` - metoda koja definiše na koji će način objekat biti štampan
- Za sve Python operatore postoje specijalne metode.

Primeru radi, postoji metoda `__add__` koja omogućava da se nad objektima naših klasa koristi operator `+`.

[Pogledati primere](#)

# Kopiranje objekata

---

kopija = objekat

NEISPRAVNO! Nego...

-----  
`from copy import copy`

...

`kopija = copy(objekat)`

# Literatura

- Lutz, Mark. Learning Python: Powerful Object-Oriented Programming. O'Reilly Media, Inc., 2013.
- [Wentworth, Peter, Elkner, Jeffrey, Downey, Allen B. and Meyers, Chris. How to Think Like a Computer Scientist: Learning with Python 3. free online book](#)
- Bunn, Tristan. Learn Python Visually: Creative Coding with Processing.py. No Starch Press, Inc., 2021.
- [Data model](#)
- ...

# Hvala



Staša Vujičić Stanković



[stasa@math.rs](mailto:stasa@math.rs)



[www.matf.bg.ac.rs/~stasa](http://www.matf.bg.ac.rs/~stasa)