

**Задатак 042:** Креирати класу Pas па креирати три објекта за псе, сваки са различитим бројем година. Написати функцију daje\_najveci\_broj() која узима све бројеве година паса као \*args и враћа број година најстаријег пса.

```
class Pas:
    def __init__(self, ime, godine):
        self.ime = ime
        self.godine = godine

def daje_najveci_broj(*args):
    return max(args)

reks = Pas("Reks", 7)
vule = Pas("Vule", 4)
aska = Pas("Aska", 5)

print("Najstariji pas ima {} godina.".format(
    daje_najveci_broj(reks.godine, vule.godine, aska.godine)))
```

Аргумент \*args омогућава да се неограничени број вредности као позиционих аргумената смести у објекат args. При томе се користи оператор распаковања \* (unpacking operator) који може да користи само торке.

**Задатак 043:** Креирати класу Pas са иницијализатором са аргументима име и године старости пса и са две методе инстанце opis и pricaj (аргумент звук). За објекат пса дати вредности за све потребне атрибуте и методе.

```
class Pas:
    vrsta = 'sisar'

    def __init__(self, ime, godine):
        self.ime = ime
        self.godine = godine

    def opis(self):
        #metoda instance
        return "{} je {} godina star.".format(self.ime, self.godine)

    def pricaj(self, zvuk):
        #metoda instance
        return "{} kaze {}".format(self.ime, zvuk)

reks = Pas("Reks", 7)
print(reks.opis())
print(reks.pricaj("Av Av"))
```

**Задатак 044:** Креирати класу Emejл са иницијализатором и методом инстанце poslati\_emejl. Иницијлаизатор и poslati\_emejl постављају атрибут poslat на False тј на True. Приказати модификацију атрибута објекта.

```
class Emejл:
    def __init__(self):
        self.poslat = False

    def poslati_emejl(self):
        self.poslat = True

moj_emejl = Emejл()
print(moj_emejl.poslat)
moj_emejl.poslati_emejl()
print(moj_emejl.poslat)
```

**Задатак 045:** Класа Новчић има иницијализатор, методе бацање и узми\_страну. Иницијализатор има атрибут страна постављен на "Глава". Метода бацање генерише случајан број у опсегу од 0 до 1 и ако се добије 0, страна се сетује на Глава, иначе је Писмо. Метода узми\_страну враћа вредност добијену за атрибут страна. Потребно је пре и после сваког бацања приказати страну новчића.

```
import random
class Novcic:
    def __init__(self):
        self.strana = 'Glava'

    def bacanje(self):
        if random.randint(0, 1) == 0:
            self.strana = 'Glava'
        else:
            self.strana = 'Pismo'

    def uzmi_stranu(self):
        return self.strana

def main():
    moj_novcic = Novcic()
    print('Pre bacanja ova strana novcica je gore:', moj_novcic.uzmi_stranu())
    print('Bacam novcic . . .')
    moj_novcic.bacanje()
    print('Posle bacanja ova strana novcica je gore:', moj_novcic.uzmi_stranu())

main()
```

Задаци за самосталан рад:

**39.** Класа Zaposleni има методу истанце prikazi која исписује име запосленог у фирми и иницијализатор са аргументом име који се односи на име запосленог у фирми.

**40.** Класа Zaposleni има иницијализатор са аргументом име (име запосленог) и атрибут radi који је постављен на True (запослени јесте у радном односу у фирми), методу истанце dao\_otkaz са атрибутом radi постављеном на False, методу истанце radni\_odnos која исписује поруку о имену запосленог и његовом тренутном радном односу са фирмом.

**41.** Класа Компјутер има иницијализатор са сетованом вредности атрибута \_\_maxцена, методу продаја за приказ максималне цене за лаптоп и сетер методу seter\_max\_cene са атрибутом cena за постављање нове макс цене.

**42.** Класа Новчић је идентична као у задатку 045. Учинити да је атрибут страна приватан чиме се спречава приступ промени резултата бацања изван класе. Потребно је пре првог бацања приказати страну новчића а затим 10 пута бацити новчић. Приказати резултате бацања процентуално.