

# PROGRAMIRANJE

ZBIRKA ZADATAKA

Ishod učenja 4

2023-2024.

## Zadatak 1

Definirajte klasu Osoba tako da je prilikom instanciranja obavezno navesti ime, prezime, OIB i godinu rođenja. Na klasu dodajte i metodu koja ispisuje sve podatke u formatu prema vašoj želji. U glavnom dijelu napravite jednu instancu tako da sve vrijednosti učitate od korisnika, a zatim sve vrijednosti iz prve instance prekopirajte u drugu instancu. Na kraju, pozivom metode ispišite sve vrijednosti iz druge instance.

*Moguće rješenje:*

---

```
class Osoba:
    def __init__(self, ime, prezime, OIB, godina_rodjenja):
        self.ime = ime
        self.prezime = prezime
        self.OIB = OIB
        self.godina_rodjenja = godina_rodjenja

    def ispisi(self):
        print(self.ime, self.prezime, self.OIB, self.godina_rodjenja)

o1 = Osoba(
    input("Upišite ime: "),
    input("Upišite prezime: "),
    input("Upišite OIB: "),
    int(input("Upišite godinu rođenja: "))
)

o2 = Osoba(
    o1.ime,
    o1.prezime,
    o1.OIB,
    o1.godina_rodjenja
)

o2.ispisi()
```

## Zadatak 2

Definirajte klasu koja čuva informacije o automobilima koji su koristili garažu (pogledajte donju tablicu). Kreirajte listu automobila sa svim automobilima iz tablice. Na kraju, omogućite korisniku da unese sat i minutu pa mu ispišite sve automobile koji su ušli u garažu nakon tog trenutka. Za svaki automobil ispišite i je li još u garaži ili ju je napustio. Primjerice, ako korisnik upiše 11 sati i 45 minuta, potrebno je ispisati:

ST 1112-AB (napustio garažu)

PU 632-IN (još je u garaži)

Automobil	Sat ulaska	Minuta ulaska	Napustio garažu
ZG 2566-PF	11	16	NE
RI 356-HF	11	37	DA
ST 1112-AB	11	55	DA
PU 632-IN	12	14	NE

*Moguće rješenje:*

```
class KorisnikGaraze:
    def __init__(self, automobil, sat_ulaska, minuta_ulaska, napustio_garazu):
        self.automobil = automobil
        self.sat_ulaska = sat_ulaska
        self.minuta_ulaska = minuta_ulaska
        self.napustio_garazu = napustio_garazu

    def ispisi(self):
        print(korisnik.automobil+ " (", end="")
        if korisnik.napustio_garazu:
            print("napustio garažu", end="")
        else:
            print("još je u garaži", end="")
        print(")")

garaza = [
    KorisnikGaraze("ZG 2566-PF", 11, 16, False),
    KorisnikGaraze("RI 356-HF", 11, 37, True),
    KorisnikGaraze("ST 1112-AB", 11, 55, True),
    KorisnikGaraze("PU 632-IN", 12, 14, False)
]

min_sat = int(input("Upišite sat ulaska: "))
min_minuta = int(input("Upišite minutu ulaska: "))

for korisnik in garaza:
    if korisnik.sat_ulaska > min_sat or (korisnik.sat_ulaska == min_sat and
    korisnik.minuta_ulaska > min_minuta):
        korisnik.ispisi()
```

### Zadatak 3

Definirajte klasu Drzava koja čuva podatke o nazivu države, broju stanovnika u državi (u milijunima) te popis proizvoljnog broja gradova u državi (0 ili više). Prilikom izrade objekta, neka korisnik zada naziv i broj stanovnika. Neka se dodavanje jednog po jednog grada može napraviti pozivom posebne metode. Nakon toga, napravite listu od tri države te svakoj izmislite broj stanovnika i dodajte proizvoljni broj gradova. Nakon toga, ispišite sve države, a gradove svake države odvojite zarezom, uz točku na kraju popisa. Primjer ispisa:

Belgija ima 11.5 milijuna stanovnika i gradove: Antwerpen, Bruxelles, Brugge.  
Finska ima 5.5 milijuna stanovnika i gradove: Helsinki, Turku, Tampere, Rovaniemi.  
Juzna Koreja ima 51 milijun stanovnika i gradove: Seul, Incheon.

*Moguće rješenje:*

```
class Drzava:
    def __init__(self, naziv, broj_stanovnika):
        self.naziv = naziv
        self.broj_stanovnika = broj_stanovnika
        self.gradovi = []

    def dodaj_grad(self, naziv_grada):
        self.gradovi.append(naziv_grada)

    def ispisi_se(self):
        print("{} ima {} milijuna stanovnika i gradove:".format(self.naziv,
self.broj_stanovnika), end=" ")

        for i in range(len(self.gradovi)):
            print(self.gradovi[i], end="")
            if i < len(self.gradovi) - 1:
                print(", ", end="")
            else:
                print(".", end="")
        print()

drzava1 = Drzava("Belgija", 11.5)
drzava1.dodaj_grad("Antwerpen")
drzava1.dodaj_grad("Bruxelles")
drzava1.dodaj_grad("Brugge")

drzava2 = Drzava("Finska", 5.5)
drzava2.dodaj_grad("Helsinki")
drzava2.dodaj_grad("Turku")
drzava2.dodaj_grad("Tampere")
drzava2.dodaj_grad("Rovaniemi")

drzava3 = Drzava("Juzna Koreja", 51)
drzava3.dodaj_grad("Seul")
drzava3.dodaj_grad("Incheon")

drzave = [ drzava1, drzava2, drzava3 ]

for d in drzave:
    d.ispisi_se()
```

## Zadatak 4

Potrebno je napisati program koji ispisuje sljedeće:

358/1/43 Visa

- 1 Kava s mlijekom 1.2
- 2 Mineralna voda 1.5

Sljedeći dio programa vam je zadan, a vaš je zadatak nadopuniti ga tako da dobijete traženi ispis:

```
racun = Racun(broj_racuna="358/1/43", nacin_placanja="Visa")
```

```
stavka1 = Stavka(redni_broj=1, proizvod="Kava s mlijekom", cijena=1.20)
stavka2 = Stavka(redni_broj=2, proizvod="Mineralna voda", cijena=1.50)
```

```
racun.dodaj_stavku(stavka1)
racun.dodaj_stavku(stavka2)
```

```
racun.ispisi()
```

*Moguće rješenje:*

```
class Stavka:
```

```
    def __init__(self, redni_broj, proizvod, cijena):
        self.redni_broj = redni_broj
        self.proizvod = proizvod
        self.cijena = cijena
```

```
    def ispisi(self):
        print("\t- {} {} {}".format(self.redni_broj, self.proizvod, self.cijena))
```

```
class Racun:
```

```
    def __init__(self, broj_racuna, nacin_placanja):
        self.broj_racuna = broj_racuna
        self.nacin_placanja = nacin_placanja
        self.stavke = []
```

```
    def dodaj_stavku(self, stavka):
        self.stavke.append(stavka)
```

```
    def ispisi(self):
        print("{} {}".format(self.broj_racuna, self.nacin_placanja))
        for s in self.stavke:
            s.ispisi()
```

```
racun = Racun(broj_racuna="358/1/43", nacin_placanja="Visa")
```

```
stavka1 = Stavka(redni_broj=1, proizvod="Kava s mlijekom", cijena=1.20)
stavka2 = Stavka(redni_broj=2, proizvod="Mineralna voda", cijena=1.50)
```

```
racun.dodaj_stavku(stavka1)
racun.dodaj_stavku(stavka2)
```

```
racun.ispisi()
```

## Zadatak 5

Napišite program koji definira klasu Datum. Neka prilikom izrade objekta te klase treba zadati godinu, mjesec i dan. Napišite i metodu koja prima drugi objekt tipa Datum i vraća manjeg. Pripremite dva datuma pa pronađite i ispišite manjeg u obliku „dan.mjesec.godina.“, npr. „8.12.2023.“

*Moguće rješenje:*

```
class Datum:
    def __init__(self, dan, mjesec, godina):
        self.dan = dan
        self.mjesec = mjesec
        self.godina = godina

    def vrati_manjeg(self, d2):
        if self.godina < d2.godina:
            return self
        elif d1.godina > d2.godina:
            return d2
        else: # godine su jednake
            if self.mjesec < d2.mjesec:
                return d1
            elif self.mjesec > d2.mjesec:
                return d2
            else: # godine i mjeseci su jednaki
                if self.dan < d2.dan:
                    return self
                else:
                    return d2

    def vrati_manjeg2(self, d2):
        ''' Ova funkcija radi istu stvar '''
        ticks_self = self.godina * 365 + self.mjesec * 12 + self.dan
        ticks_d2 = d2.godina * 365 + d2.mjesec * 12 + d2.dan;

        if ticks_self < ticks_d2:
            return self
        else:
            return d2

d1 = Datum(dan=8, mjesec=10, godina=2020)
d2 = Datum(dan=9, mjesec=11, godina=2021)

m = d1.vrati_manjeg2(d2)
print("{}.{}.{}.".format(m.dan, m.mjesec, m.godina))
```

## Zadatak 6

Napišite program koji priprema listu objekata iz donje tablice. Omogućite korisniku da upiše duljinu u kilometrima, a vi mu ispišite sve rijeke koje su dulje od toga. Primjerice, ako korisnik upiše 6500, onda trebate ispisati:

Nil (Afrika)

Amazona (Južna Amerika)

Rijeka	Kontinent	Duljina [km]
Nil	Afrika	6695
Amazona	Južna Amerika	6570
Jangce	Azija	6380

*Moguće rješenje:*

```
class Rijeka:
    def __init__(self, naziv, kontinent, duljina):
        self.naziv = naziv
        self.kontinent = kontinent
        self.duljina = duljina

    def ispisi(self):
        print("{} ({}).format(self.naziv, self.kontinent))

rijeke = [
    Rijeka("Nil", "Afrika", 6695),
    Rijeka("Amazona", "Južna Amerika", 6570),
    Rijeka("Jangce", "Azija", 6380)
]

min_duljina = int(input("Upisite minimalnu duljinu rijeke: "))

for i in range(len(rijeke)):
    if rijeke[i].duljina > min_duljina:
        rijeke[i].ispisi()
```

## Zadatak 7

Napišite program koji definira klasu IpAdresa koja čuva podatke o jednoj IP adresi (dobro razmislite kako ćete čuvati te podatke). Napišite i metodu koja vraća klasu adrese (A, B ili C), prema sljedećim kriterijima:

- Klasu A čine adrese od 0.0.0.0 do 127.255.255.255
- Klasu B čine adrese od 128.0.0.0 do 191.255.255.255
- Klasu C čine adrese od 192.0.0.0 do 223.255.255.255

Napravite objekt, pozovite metodu i ispišite kojoj klasi pripada ta IP adresa kao u primjeru:

IP adresa 13.107.246.60 pripada klasi A

*Moguće rješenje:*

```
class IpAdresa:
    def __init__(self, prvi_bajt, drugi_bajt, treci_bajt, cetvrti_bajt):
        self.prvi_bajt = prvi_bajt
        self.drugi_bajt = drugi_bajt
        self.treci_bajt = treci_bajt
        self.cetvrti_bajt = cetvrti_bajt

    def ispisi(self):
        print("IP adresa {}.{}.{}.{} pripada klasi {}".format(self.prvi_bajt,
self.drugi_bajt, self.treci_bajt, self.cetvrti_bajt, self.klasa()))

    def klasa(self):
        if self.prvi_bajt <= 127:
            return 'A'
        elif self.prvi_bajt <= 191:
            return 'B'
        else:
            return 'C'

# ping openai.com
# Pinging openai.com [13.107.246.60] with 32 bytes of data :
ip = IpAdresa(13, 107, 246, 60)
ip.ispisi()
```

## Zadatak 8

Napišite program koji definira klase Zvijezda, Planet i Mjesec. Za sva tri svemirska tijela omogućite čuvanje naziva. Dodatno, za zvijezde omogućite čuvanje proizvoljnog broja planeta, a za planete proizvoljnog broja mjesece. Primjerice, za zvijezdu Sunce možemo upisati planete Merkur, Veneru i Mars, a za planet Mars mjesece Deimos i Phobos. Pripremite listu s nekoliko zvijezda. Na kraju, ispišite sadržaj tako da zvijezde ispišete bez uvlačenja, planete jednom uvučeno, a mjesece dva puta, kao u prikazu:

Sunce

    Merkur

    Mars

        Deimos

        Phobos

    Jupiter

        Io

        Ganymede

        Callisto

Kosjenka

*Moguće rješenje:*

```
class Mjesec:
    def __init__(self, naziv):
        self.naziv = naziv

    def ispisi(self):
        print("\t\t", self.naziv)

class Planet:
    def __init__(self, naziv):
        self.naziv = naziv
        self.mjesece = []

    def dodaj_mjesec(self, naziv):
        self.mjesece.append(naziv)

    def ispisi(self):
        print("\t", self.naziv)
        for m in self.mjesece:
            m.ispisi()

class Zvijezda:
    def __init__(self, naziv):
        self.naziv = naziv
        self.planeti = []

    def dodaj_planet(self, naziv):
        self.planeti.append(naziv)

    def ispisi(self):
        print(self.naziv)
        for p in self.planeti:
            p.ispisi()

# planeti
merkur = Planet("Merkur")
```

```
mars = Planet("Mars")
mars.dodaj_mjesec(Mjesec("Deimos"))
mars.dodaj_mjesec(Mjesec("Phobos"))

jupiter = Planet("Jupiter")
jupiter.dodaj_mjesec(Mjesec("Io"))
jupiter.dodaj_mjesec(Mjesec("Ganymede"));
jupiter.dodaj_mjesec(Mjesec("Callisto"))

# zvijezde
sunce = Zvijezda("Sunce")
sunce.dodaj_planet(merkur);
sunce.dodaj_planet(mars);
sunce.dodaj_planet(jupiter);

kosjenka = Zvijezda("Kosjenka")

zvijezde = [ sunce, kosjenka ]

for z in zvijezde:
    z.ispisi()
```

## Zadatak 9

Napišite program koji definira klasu Udžbenik prilikom čije izrade treba zadati naziv i broj stranica. U glavnom dijelu programa učitavajte od korisnika udžbenike dok to korisnik želi. Nakon učitavanja, ispišite naziv udžbenika koji ima najviše stranica (ispišite i broj stranica) te ispišite ukupan zbroj stranica svih udžbenika. Primjer rada:

```
Upišite naziv: Udžbenik 1
Upišite broj stranica: 42
Dalje (1=da, 0=ne): 1
Upišite naziv: Udžbenik 2
Upišite broj stranica: 195
Dalje (1=da, 0=ne): 1
Upišite naziv: Udžbenik 3
Upišite broj stranica: 75
Dalje (1=da, 0=ne): 0
Najviše stranica (195) ima Udžbenik 2.
Ukupno svi udžbenici imaju 312 stranica.
```

*Moguće rješenje:*

```
class Udžbenik:
    def __init__(self, naziv, broj_stranica):
        self.naziv = naziv
        self.broj_stranica = broj_stranica

udzbenici = []

while (True):
    novi = Udžbenik(
        input("Upišite naziv: "),
        int(input("Upišite broj stranica: "))
    )
    udzbenici.append(novi)

    if input("Dalje (1=da, 0=ne): ") == "0":
        break

najdeblji = udzbenici[0]
ukupno_stranica = 0

for udzbenik in udzbenici:
    if udzbenik.broj_stranica > najdeblji.broj_stranica:
        najdeblji = udzbenik

    ukupno_stranica += udzbenik.broj_stranica;

print("Najviše stranica ({} ) ima {}".format(najdeblji.broj_stranica,
    najdeblji.naziv))
print("Ukupno svi udžbenici imaju {} stranica.".format(ukupno_stranica))
```

## Zadatak 10

Napišite program koji definira klasu Prognoza prilikom čije izrade treba zadati datum, najveću temperaturu i opis vremena (primjerice, za datum „9.12.2023.“ je najveća temperatura 2 stupnja, a opis „oblačno“). U glavnom dijelu programa omogućite korisniku da unese tri prognoze. Nakon unosa svih prognoza, ispišite ih u sljedećem formatu:

Upišite datum: 9.12.2023.

Upišite najveću temperaturu: 2

Upišite opis: oblačno

Upišite datum: 10.12.2023.

Upišite najveću temperaturu: 7

Upišite opis: kiša

Upišite datum: 11.12.2023.

Upišite najveću temperaturu: 8

Upišite opis: oblačno

Na datum 9.12.2023. će najveća temperatura biti 2.0 (oblačno)

Na datum 10.12.2023. će najveća temperatura biti 7.0 (kiša)

Na datum 11.12.2023. će najveća temperatura biti 8.0 (oblačno)

*Moguće rješenje:*

```
class Prognoza:
    def __init__(self, datum, max_temperatura, opis):
        self.datum = datum
        self.max_temperatura = max_temperatura
        self.opis = opis

    def ispisi(self):
        print("Na datum {} će najveća temperatura biti {} ({}).format(self.datum,
self.max_temperatura, self.opis))

prognoze = []

for i in range(3):
    nova = Prognoza(
        input("Upišite datum: "),
        float(input("Upišite najveću temperaturu: ")),
        input("Upišite opis: ")
    )
    prognoze.append(nova)

for p in prognoze:
    p.ispisi()
```

## Zadatak 11

Napišite program koji definira klasu TekuciRacun prilikom čije izrade je potrebno zadati IBAN broj. U beskonačnoj petlji korisniku omogućite:

- Otvaranje novog tekućeg računa.
- Dodavanje sredstava na tekući račun. Pretpostavite da će korisnik uvijek unijeti postojeći broj računa.
- Ispis svih tekućih računa s iznosima.

Primjer rada programa:

```
1 = otvaranje novog računa
2 = dodavanje sredstava na račun
3 = ispis svega
> 1
Upišite broj računaHR1723600001101234565
> 2
Upišite broj računaHR1723600001101234565
Koliko EUR želite dodati: 1520
> 1
Upišite broj računaAT483200000012345864
> 2
Upišite broj računaAT483200000012345864
Koliko EUR želite dodati: 4000
> 3
HR1723600001101234565: 1520.0 EUR
AT483200000012345864: 4000.0 EUR
> 2
Upišite broj računaHR1723600001101234565
Koliko EUR želite dodati: 485
> 3
HR1723600001101234565: 2005.0 EUR
AT483200000012345864: 4000.0 EUR
...
```

*Moguće rješenje:*

```
class TekuciRacun:
    def __init__(self, broj):
        self.broj = broj
        self.iznos = 0

    def ispisi(self):
        print("{}: {} EUR".format(self.broj, self.iznos))

racuni = []

print("1 = otvaranje novog računa")
print("2 = dodavanje sredstava na račun")
print("3 = ispis svega")

while (True):
    op = input("> ")

    if op == "1":
        racuni.append(
```

```
        TekuciRacun(  
            input("Upišite broj računa")  
        )  
    )  
elif op == "2":  
    broj = input("Upišite broj računa")  
    iznos = float(input("Koliko EUR želite dodati: "))  
  
    for r in racuni:  
        if r.broj == broj:  
            r.iznos += iznos  
            break  
elif op == "3":  
    for r in racuni:  
        r.ispisi()  
else:  
    print("Nepoznata operacija")
```

## Zadatak 12

Napišite program koji priprema listu objekata iz donje tablice. Nakon toga, omogućite korisniku da upiše s koliko novca raspolaže, a vi mu ispišite do kojih sve odredišta može stići. Primjer rada programa:

Unesite iznos kojim raspolažete: 8.00

Možete posjetiti neko od sljedećih odredišta:

Ivanić Grad

Kutina

Novska

Odredište	Iznos cestarine
Ivanić Grad	1.50 EUR
Kutina	4.20 EUR
Novska	5.50 EUR
Sl. Brod zapad	10.80 EUR

*Moguće rješenje:*

```
class Odrediste:
    def __init__(self, naziv, iznos):
        self.naziv = naziv
        self.iznos = iznos

    def ispisi(self):
        print(self.naziv)

odredista = [
    Odrediste("Ivanić Grad", 1.50),
    Odrediste("Kutina", 4.20),
    Odrediste("Novska", 5.50),
    Odrediste("Sl. Brod zapad", 10.80)
]

iznos_na_raspolaganju = float(input("Unesite iznos kojim raspolažete: "))

print("Možete posjetiti neko od sljedećih odredišta:")
for o in odredista:
    if o.iznos <= iznos_na_raspolaganju:
        o.ispisi()
```

## Zadatak 13

Napišite program koji priprema listu objekata iz donje tablice. Nakon toga, učitajte od korisnika polazni i odredišni aerodrom pa mu ispišite sve raspoložive letove. Ako nema niti jednog raspoloživog leta, ispišite to. Primjer rada programa:

Upišite oznaku polazišnog aerodroma: ZAG

Upišite oznaku odredišnog aerodroma: CDG

Raspoloživi letovi:

OU220

AF5112

Polazište	Odredište	Broj leta
ZAG	CDG	OU220
CDG	VIE	LH1951
MUC	FRA	LH9562
ZAG	CDG	AF5112
ZAG	DBV	OU440

*Moguće rješenje:*

```
class Let:
    def __init__(self, polaziste, odrediste, broj):
        self.polaziste = polaziste
        self.odrediste = odrediste
        self.broj = broj

    def ispisi(self):
        print(self.broj)

letovi = [
    Let("ZAG", "CDG", "OU220"),
    Let("CDG", "VIE", "LH1951"),
    Let("MUC", "FRA", "LH9562"),
    Let("ZAG", "CDG", "AF5112"),
    Let("ZAG", "DBV", "OU440")
]

polaziste = input("Upišite oznaku polazišnog aerodroma: ")
odrediste = input("Upišite oznaku odredišnog aerodroma: ")

pronadjen_ijedan_let = False

print("Raspoloživi letovi:")
for let in letovi:
    if let.polaziste == polaziste and let.odrediste == odrediste:
        let.ispisi()
        pronadjen_ijedan_let = True

if pronadjen_ijedan_let == False:
    print("Na žalost, nema niti jednog raspoloživog leta")
```

## Zadatak 14

Napišite program koji definira klasu Korisnik prilikom čije izrade je potrebno zadati ime, prezime, korisničko ime, lozinku te je li korisnik administrator ili nije. Napišite funkciju koja od korisnika učitava jedan objekt tipa Korisnik i vraća ga pozivatelju. U glavnom dijelu programa, iskoristite funkciju da učitate jednog korisnika pa ga onda ispišite na ekran u formatu:

```
Korisnik: Mirić, Žarko
Podaci za prijavu: korisničko ime=zmiric, lozinka=78ocaA5ERaMeSU1UqPrAWdU6
Korisnik ima administrativne ovlasti
```

*Moguće rješenje:*

```
class Korisnik:
    def __init__(self, ime, prezime, korisnicko_ime, lozinka, is_admin):
        self.ime = ime
        self.prezime = prezime
        self.korisnicko_ime = korisnicko_ime
        self.lozinka = lozinka
        self.is_admin = is_admin

    def ispisi(self):
        print("Korisnik: {}, {}".format(self.prezime, self.ime))
        print("Podaci za prijavu: korisničko ime={},
lozinka={}".format(self.korisnicko_ime, self.lozinka))
        if self.is_admin == "DA":
            print("Korisnik ima administrativne ovlasti")
        else:
            print("Obični korisnik")

def ucitaj():
    k = Korisnik(
        input("ime: "),
        input("prezime: "),
        input("korisnicko ime: "),
        input("lozinka: "),
        input("Je li korisnik administrator (DA, NE): ")
    )
    return k

k = ucitaj()
k.ispisi()
```

## Zadatak 15

Napišite program koji definira klasu ZnanstveniRad koja može čuvati podatke o jednom ili više autora, naslovu te godini izdanja. Prilikom izrade objekta neka korisnik treba zadati naslov i godinu izdanja. Omogućite naknadno dodavanje autora. U glavnom dijelu programa, učitajte od korisnika podatke o jednom znanstvenom radu. Nakon završenog učitavanja, metodom ispišite objekt u sljedećem formatu:

- Autori su međusobno odvojeni zarezima, iza zadnjeg autora nema zareza.
- Godina se piše u okruglim zagradama
- Naslov se navodi zadnji s točkom na kraju.

Primjer rada programa:

Naslov: Toroidal Atmospheres around Extrasolar Planets

Godina izdanja: 2006

Autor: R. E. Johnson

Unijeti još jednog autora (da, ne): da

Autor: P. J. Huggins

Unijeti još jednog autora (da, ne): ne

R. E. Johnson, P. J. Huggins (2006) Toroidal Atmospheres around Extrasolar Planets.

*Moguće rješenje:*

```
class ZnanstveniRad:
    def __init__(self, naslov, godina_izdanja):
        self.naslov = naslov
        self.godina_izdanja = godina_izdanja
        self.autori = []

    def dodaj_autora(self, a):
        self.autori.append(a)

    def ispisi(self):
        for i in range(len(self.autori)):
            print(self.autori[i], end="")
            if i < len(self.autori) - 1:
                print(", ", end="")
        print(" ({} {})." .format(self.godina_izdanja, self.naslov))

zr = ZnanstveniRad(
    input("Naslov: "),
    int(input("Godina izdanja: "))
)

while True:
    a = input("Autor: ")
    zr.dodaj_autora(a)

    if input("Unijeti još jednog autora (da, ne): ") == "ne":
        break

zr.ispisi()
```

## Zadatak 16

Napišite program koji simulira rad DNS poslužitelja. Pripremite listu objekata prema donjoj tablici i u beskonačnoj petlji omogućite korisniku dvije opcije:

- Ako unese domensko ime, vi mu ispišite IP adresu.
- Ako unese IP adresu, vi mu ispišite sva domenska imena na toj IP adresi.

Primjer rada programa:

```
1 = domensko ime => ip adresa
2 = ip adresa => domenska imena
> 1
Upit: mail.google.com
142.250.180.133
> 2
Upit: 142.251.209.3
www.google.hr
> 2
Upit: 178.218.162.131
algebra.hr
www.algebra.hr
> ...
```

Domensko ime	IP adresa
www.google.com	142.250.180.132
www.google.hr	142.251.209.3
mail.google.com	142.250.180.133
algebra.hr	178.218.162.131
www.algebra.hr	178.218.162.131

*Moguće rješenje:*

```
class DnsZapis:
    def __init__(self, domensko_ime, ip_adresa):
        self.domensko_ime = domensko_ime
        self.ip_adresa = ip_adresa

dns = [
    DnsZapis("www.google.com", "142.250.180.132"),
    DnsZapis("www.google.hr", "142.251.209.3"),
    DnsZapis("mail.google.com", "142.250.180.133"),
    DnsZapis("algebra.hr", "178.218.162.131"),
    DnsZapis("www.algebra.hr", "178.218.162.131")
]

print("1 = domensko ime => ip adresa")
print("2 = ip adresa => domenska imena")

while True:
    opcija = input("> ")
    upit = input("Upit: ")

    if opcija == "1":
        for entry in dns:
            if entry.domensko_ime == upit:
```

```
        print(entry.ip_adresa)
        break
elif opcija == "2":
    for entry in dns:
        if entry.ip_adresa == upit:
            print(entry.domensko_ime)
else:
    print("Nepoznata opcija")
```

## Zadatak 17

Napišite program koji definira klasu Pločica koja čuva x i y koordinate lijevog gornjeg kuta. Neka program od korisnika učitava širinu i visinu ekrana u pikselima (pretpostavite da će korisnik uvijek unijeti širinu i visinu djeljivu s brojem 10). Neka program zatim generira potrebni broj objekata tipa Pločica širine i visine 10 piksela tako da se pokrije cijelu širina i visina ekrana, ako pretpostavimo da je prva generirana pločica na koordinatama (0, 0). Ispišite sve generirane pločice. Primjer rada programa:

Upišite širinu ekrana: 80

Upišite visinu ekrana: 50

```
(0,0) (10,0) (20,0) (30,0) (40,0) (50,0) (60,0) (70,0)
(0,10) (10,10) (20,10) (30,10) (40,10) (50,10) (60,10) (70,10)
(0,20) (10,20) (20,20) (30,20) (40,20) (50,20) (60,20) (70,20)
(0,30) (10,30) (20,30) (30,30) (40,30) (50,30) (60,30) (70,30)
(0,40) (10,40) (20,40) (30,40) (40,40) (50,40) (60,40) (70,40)
```

*Moguće rješenje:*

```
class Plocica:
    def __init__(self, x, y):
        self.x = x
        self.y = y

sirina = int(input("Upišite širinu ekrana: "))
visina = int(input("Upišite visinu ekrana: "))

plocice = []

for i in range(0, visina, 10):
    for j in range(0, sirina, 10):
        pl = Plocica(j, i)
        plocice.append(pl)

for pl in plocice:
    if pl.x == 0:
        print()
    print("{}\t".format(pl.x, pl.y), end="")
```

## Zadatak 18

Napišite program koji definira klasu `KolicinaKise` prilikom čije izrade treba zadati količinu pale kiše i redni broj mjeseca (primjerice, u 6. mjesecu je palo 47 mm kiše). Napravite listu sa svih 12 mjeseci i za svaki mjesec generirajte slučajnu količinu pale kiše između 25 i 100 mm. Nakon toga, pronađite u kojem mjesecu je palo najviše kiše i ispišite njegov broj i količinu kiše.

*Moguće rješenje:*

---

```
import random

class KolicinaKise:
    def __init__(self, mjesec, kolicina):
        self.mjesec = mjesec
        self.kolicina = kolicina

padaline = []

for i in range(1, 13):
    padaline.append(
        KolicinaKise(
            i,
            random.randint(25, 100)
        )
    )

najvise = padaline[0]
for p in padaline:
    if p.kolicina > najvise.kolicina:
        najvise = p

print("Najvise kiše je palo u mjesecu broj {} ({} mm)".format(najvise.mjesec,
    najvise.kolicina))
```

## Zadatak 19

Napišite program koji definira klasu `GeneratorLozinke`. Omogućite korisniku da prilikom instanciranja klase zada želi li da lozinke sadržavaju i specijalne znakove. Napišite metodu koja prima željenu duljinu lozinke i generira lozinku prema sljedećim pravilima:

- Prva kategorija znakova su mala slova: `abcdefghijklmnopqrstuvwxy`
- Druga kategorija znakova su velika slova: `ABCDEFGHIJKLMN`
- Treća kategorija znakova su brojke: `0123456789`
- Četvrta kategorija znakova su specijalni znakovi: `!#$%()*+,-./`
- Za svaki znak lozinke treba prvo slučajno odabrati jednu od tri ili četiri prethodne kategorije, a zatim iz te kategorije slučajno odabrati jedan znak.

U glavnom dijelu generirajte pet lozinki bez i pet lozinki sa specijalnim znakovima, primjerice:

Lozinke bez specijalnih znakova:

```
Vfk3f241uM8Gr1iA  
k4fwlvs10hh5932Y  
mN4MpAA3L0sNoigd  
aQ7hh7NETjR63Zw2  
2WAItae7Jd4f08Pu
```

Lozinke sa specijalnim znakovima:

```
R*X1088Ku6%I5Y-4  
8)781GP/B8(2(0u+  
#-0aH2*(8+5mtVo*  
j(52(/dk8Z9136h%  
1$JpL#59DS)Aa27)
```

*Moguće rješenje:*

```
import random  
  
class GeneratorLozinke:  
    def __init__(self, ukljuci_specijalne_znakove):  
        self.kategorije = [  
            "abcdefghijklmnopqrstuvwxy",  
            "ABCDEFGHIJKLMN",  
            "0123456789",  
        ]  
        if ukljuci_specijalne_znakove:  
            self.kategorije.append("!#$%()*+,-./")  
  
    def generiraj(self, n):  
        lozinka = ""  
        while len(lozinka) < n:  
            kategorija = self.slucajna_kategorija()  
            lozinka += self.slucajni_znak(kategorija)  
        return lozinka  
  
    def slucajna_kategorija(self):  
        ri = random.randint(0, len(self.kategorije) - 1)  
        return self.kategorije[ri]  
  
    def slucajni_znak(self, kategorija):  
        ri = random.randint(0, len(kategorija) - 1)
```

```
        return kategorija[ri]

print("Lozinke bez specijalnih znakova:")
gen2 = GeneratorLozinke(False)
for i in range(5):
    print(gen2.generiraj(16))

print("\nLozinke sa specijalnim znakovima:")
gen1 = GeneratorLozinke(True)
for i in range(5):
    print(gen1.generiraj(16))
```

## Zadatak 20

Napišite program koji definira klase Točka i Pravokutnik. Neka Točka čuva podatke o x i y koordinati, a Pravokutnik čuva podatke o koordinatama gornjeg lijevog kuta te širini i visini. Na klasi Pravokutnik napišite i metodu koja prima točku i vraća nalazi li se točka unutar pravokutnika. Demonstrirajte rad metode. Pretpostavite da se (0,0) nalazi u gornjem lijevom kutu, a da x i y rastu u desno i prema dolje.

*Moguće rješenje:*

```
class Točka:
    def __init__(self, x, y):
        self.x = x
        self.y = y

class Pravokutnik:
    def __init__(self, gore_lijevo, sirina, visina):
        self.gore_lijevo = gore_lijevo
        self.sirina = sirina
        self.visina = visina

    def kolizija(self, t):
        if (t.x >= self.gore_lijevo.x and
            t.x <= self.gore_lijevo.x + self.sirina and
            t.y >= self.gore_lijevo.y and
            t.y <= self.gore_lijevo.y + self.visina):
            return True
        else:
            return False

t = Točka(16, 15)

p = Pravokutnik(
    gore_lijevo=Točka(5, 5),
    sirina=10,
    visina=10
)

if p.kolizija(t):
    print("Točka je unutar pravokutnika")
else:
    print("Točka je izvan pravokutnika")
```