

Sadržaj

① Operatori

- Aritmetički operatori
- Operatori poređenja
- Logički operatori
- Operatori nad bitovima
- Operatori pripadnosti
- Operatori identifikacije

② Grananje

③ Zadaci

④ Literatura



Sadržaj

① Operatori

- Aritmetički operatori
- Operatori poređenja
- Logički operatori
- Operatori nad bitovima
- Operatori pripadnosti
- Operatori identifikacije

② Grananje

③ Zadaci

④ Literatura



Sadržaj

① Operatori

- Aritmetički operatori
- Operatori poređenja
- Logički operatori
- Operatori nad bitovima
- Operatori pripadnosti
- Operatori identifikacije

② Grananje

③ Zadaci

④ Literatura



Aritmetički operatori

Oznaka	Značenje
+	sabiranje brojeva (celih i/ili decimalnih), unarni operator koji označava pozitivan broj sabiranje (spajanje ili konkatencija) struktura
-	oduzimanje brojeva i unarni operator koji označava negativan broj
*	množenje brojeva ili umnožavanje struktura (stringova, listi, ...)
/	deljenje brojeva
//	celobrojno deljenje brojeva
**	stepenovanje brojeva
%	ostatak pri deljenju (modul(o))

Primeri aritmetičkih operatora

Primeri aritmetičkih operatora

```
1 2 - 3 # -1
2 45 // 7 # 6
3 2 ** 4 # 16
4 10 % 3 # 1
5 45 / 5 # 9.0 rezultat deljenja je UVEK decimalni broj
6 +3 + 6 # 9 unarni i binarni operator +
7 0.1 + 0.2 # zasto imamo ovakav rezultat?
8 "abc" * 3 # "abcabcabc" upotreba za umnozavanje struktura
9 "abc" + "def" # "abcdef" upotreba za konkatenciju
   stringova
```

10

Sadržaj

① Operatori

- Aritmetički operatori
- **Operatori poređenja**
- Logički operatori
- Operatori nad bitovima
- Operatori pripadnosti
- Operatori identifikacije

② Grananje

③ Zadaci

④ Literatura



Operatori poređenja

- Najčešće ove operatore koristimo nad brojevima, ali možemo ih koristiti i nad drugim tipovima, kao na primer nad strukturama.

Oznaka	Značenje
i	strogo manje
i=	manje ili jednako
i	strogo veće
i=	veće ili jednako
!=	različito
==	jednako

Primeri operatora poredjenja

Primeri operatora poredjenja

```
1 10 > 5 # True
2 5 <= 4 # False
3 3 < 6 # True
4 10 >= 4 # True
5 4 != 6 # True
6 3 == 3 # True
7 "a" < "b" # True
8 "b" == "a" # False
9
```


Sadržaj

① Operatori

- Aritmetički operatori
- Operatori poređenja
- **Logički operatori**
- Operatori nad bitovima
- Operatori pripadnosti
- Operatori identifikacije

② Grananje

③ Zadaci

④ Literatura



Logički operatori

- Najčešće ove operatore koristimo nad istinitosnim vrednostima, ali možemo ih koristiti i nad drugim tipovima.

Oznaka	Značenje
and	i
or	ili
not	unarni operator negacije

Primeri logičkih operatora

Primeri logičkih operatora

```
1 True and False # False
2 True or False # True
3 not True # False
4 1 or 0 # 1
5 0 or 0 # 0
6 10 and 0 # 0
7 10 or 0 # 10
8 "string" or "" # "string" - neprazan string je "tacan"
9 "string" and "" # "" - kao prazan string ("netacno")
10
```

Sadržaj

1 Operatori

- Aritmetički operatori
- Operatori poređenja
- Logički operatori
- **Operatori nad bitovima**
- Operatori pripadnosti
- Operatori identifikacije

2 Grananje

3 Zadaci

4 Literatura



Operatori nad bitovima

- Rezultat primene operatora nad bitovima najbolje uočavamo, kada ih predstavimo u binarnom zapisu.
- Kako bismo to realizovali koristićemo ugrađenu funkciju `bin(x)`, gde je `x` literal ili promenljiva od koje tražimo binarni zapis.
- Ova metoda vraća rezultat tipa `str` (string).

Oznaka	Značenje
<code>&</code>	i
<code> </code>	ili
<code>~</code>	unarni operator negacije
<code>^</code>	ekskluzivno ili (xor)
<code><<</code>	pomeranje bitova u levo
<code>>></code>	pomeranje bitova u desno

Primeri operatora nad bitovima

Primeri operatora nad bitovima

```
1 a = 1
2 b = 0
3 bin(a) # '0b1'
4 bin(b) # '0b0'
5 bin(a & b) # '0b0'
6 bin(a ^ b) # '0b1'
7 bin(a | b) # '0b1'
8 bin(~ a) # '-0b10'
9 c = 6
10 bin(c) # '0b110'
11 bin(c << 2) # '0b11000'
12 bin(c >> 2) # '0b1'
13
```

Sadržaj

① Operatori

- Aritmetički operatori
- Operatori poređenja
- Logički operatori
- Operatori nad bitovima
- **Operatori pripadnosti**
- Operatori identifikacije

② Grananje

③ Zadaci

④ Literatura



Operatori pripadnosti

- Proveravaju da li određena vrednost pripada nekoj kolekciji.
- Rezultat provere je uvek istinitosna vrednost.

Oznaka	Značenje
in	pripada
not in	ne pripada

Primeri operatora pripadnosti

Primeri operatora pripadnosti

```
1 "a" in "Marko" # True
2 "b" in "Marko" # False
3
```

Sadržaj

① Operatori

- Aritmetički operatori
- Operatori poređenja
- Logički operatori
- Operatori nad bitovima
- Operatori pripadnosti
- Operatori identifikacije

② Grananje

③ Zadaci

④ Literatura



Operatori identifikacije

- Proveravaju da li su održene vrednosti jednake po svojim identifikatorima.
- Identifikator vrednosti dobijamo uz pomoć ugrađene funkcije `id(x)`, gde je `x` vrednost od koje tražimo identifikator.
- Identifikator predstavlja memorijsku lokaciju gde je vrednost smeštena.
- Voditi računa da ovde ključna reč `not` ide posle operatora `is`.

Oznaka	Značenje
<code>is</code>	identični
<code>is not</code>	neidentični

Primeri grananja

Primer jednostavnog grananja

```
1 "a" is "a" # True
2 "Ana" is "ana" # False
3 "" is "" # True
4 "" is not "ANA" # True
5 a = "Ana"
6 b = a + a # "AnaAna"
7 a + a is b # False
8 a + a == b # True - ovde je operator JEDNAKOSTI
9
```

Sadržaj

① Operatori

- Aritmetički operatori
- Operatori poređenja
- Logički operatori
- Operatori nad bitovima
- Operatori pripadnosti
- Operatori identifikacije

② Grananje

③ Zadaci

④ Literatura



Grananje

- U programiranju je često neophodno uraditi nešto spram ispunjenosti nekog uslova. Na primer dodeliti ocenu studentu ako student ostvari bodove za ocenu.
- Kako je uslov logički izraz, za njegovo formiranje koristimo prethodno naučene operatore poređenja i vrednosti iz literala i/ili promenljivih.
- Grananje je blokovska naredba, što znači da se u bloku ispod uslova za grananjem navodi programski kod koji želimo da se izvrši ako je uslov zadovoljen (logički izraz je izračunat kao `True`).

Blokovske naredbe

- Blokovske naredbe u Python-u započinju tako što se na kraju naredbe koja je blokovskog tipa stavlja dvotačka (:), a potom se u novom redu, uvučeno za 4 space karaktera ili 1 tab, pišu naredbe koje su u bloku. Kraj bloka predstavlja poslednje uvučena naredba. Nakon nje se ostale naredbe pišu bez uvlačenja.
- Svaka blokovska naredba u Python-u mora da ima bar 1 liniju koda u bloku.
- Postoji još nekoliko blokovskih naredbi. Kasnije ćemo pomenuti još jednu - petlju.

Grananje - if

- Najjednostavnije grananje je grananje naredbom if.
- Potrebno je da formiramo uslov i da specificiramo blok naredbi koje će se izvršiti kada je uslov zadovoljen.

Šablon naredbe

```
1 if uslov:  
2     blok naredbi  
3
```


Primeri grananja

Primeri grananja

```
1 bodovi = 10
2 if bodovi >= 51:
3     print("Student je položio predmet!")
4 # drugi primer
5 markov_indeks = "2020/270123"
6 sanjin_indeks = "2020/270321"
7 if markov_indeks != sanjin_indeks:
8     print("Razliciti studenti!")
9
```

Grananje - if/else

- Nekad nam je potrebno da izvršimo kod i kada nije zadovoljen uslov specificiran u if-u. Tada koristimo uz **if** i **else** blokovsku naredbu.

Šablon naredbe

```
1 if uslov:
2     blok naredbi
3 else:
4     blok naredbi 2
5
```

Grananje - if/elif/else

- Postoje situacije u kojima postoji više mogućih izbora (više od dva). Tada koristimo grananje sa dodatnim elif delom u kojem se obavezno specificira uslov. Ispod **elif** se navodi blok koji se izvršava po zadovoljavanju uslova.
- Naredbi **elif** može biti proizvoljno mnogo, i uvek se navodi posle **if** naredbe, a pre **else** naredbe.
- Uslovi koji se navode u **if** i **elif** delovima bi trebali da budu različiti.

Šablon naredbe

```
1 if uslov:
2     blok naredbi
3 elif uslov2:
4     blok naredbi 2
5 else:
6     blok naredbi 3
7
```

Grananje - match/case

- Počevši od verzije 3.10 dodat je novi način za grananje (poklapanje vrednosti) **match/case**.
- Koristimo ga kada je potrebno da se vrednost subjekta provere poredi sa vrednostima šablona navedenih u **case** segmentima.
- Naredbi **case** može biti proizvoljno mnogo. Subjekat se poredi po tipu i vrednosti sa zadatim šablonom, počevši od prvog do poslednjeg, a izvršavaju se akcije u onom koji bude zadovoljen.
- Naredba **case** _ se koristi kao podrazumevani, odnosno zadovoljeni šablon u slučaju da nijedan prethodni nije.

Grananje - match/case

- Pri navođenju šablona, moguće je navesti više vrednosti uz navođenje ili operatora nad bitovima (—).

Šablon naredbe

```
1 match subject:
2     case <pattern_1>:
3         <action_1>
4     case <pattern_2>:
5         <action_2>
6     case _:
7         <action_wildcard>
8
```

Sadržaj

① Operatori

- Aritmetički operatori
- Operatori poređenja
- Logički operatori
- Operatori nad bitovima
- Operatori pripadnosti
- Operatori identifikacije

② Grananje

③ Zadaci

④ Literatura



Zadatak 1

- Milica ima 17 godina.
- Ako Milica ima 18 ili više godina ispisati na konzolu da je punoletna.



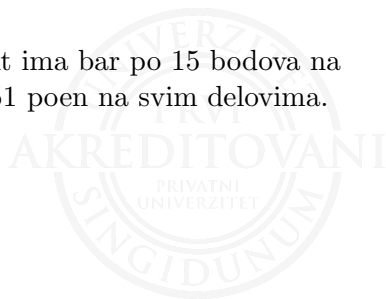
Zadatak 2

- Dopuniti prethodno rešenje tako da se u suprotnom ispisuje na konzolu da je maloletna.



Zadatak 3

- Student je na I kolokvijumu ostvario 23 poena, na II kolokvijumu 14 poena, a na završnom ispitu 21 poen.
- Proveriti da li je student položio ispit, i ako jeste ispisati da je ispit položen.
- Napomena: ispit je položen ako student ima bar po 15 bodova na kolokvijumima i ako ima ukupno bar 51 poen na svim delovima.



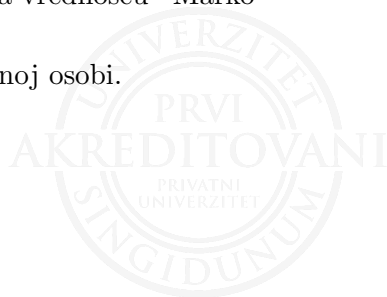
Zadatak 4

- Dopuniti prethodni zadatak tako da se studentu ispisuje i koju je ocenu osvojio. Pravila ocenjivanja spram osvojenih bodova su sledeća:
 - ocena 6: 51-60 bodova,
 - ocena 7: 61-70 bodova,
 - ocena 8: 71-80 bodova,
 - ocena 9: 81-90 bodova i
 - ocena 10: 91-100 bodova.



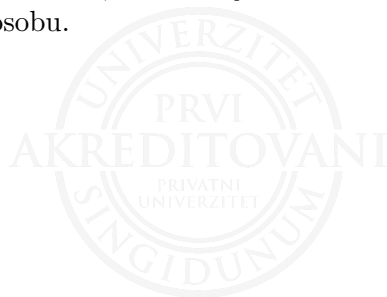
Zadatak 5

- Tražiti od korisnika da unese svoje ime. Sačuvati ime u promenljivu.
- Definirati promenljivu tražena osoba sa vrednošću "Marko Markovic"
- Ispitati da li postoji uneto ime u traženoj osobi.



Zadatak 6

- Dopuniti prethodni zadatak tako da u slučaju kada pogodi ime mu se na konzolu ispiše pogodili ste traženu osobu, a u slučaju da ne pogodi ispisati niste pogodili traženu osobu.



Zadatak 7

- Dopuniti prethodni zadatak tako da se od korisnika dodatno traži da unese i prezime.
- Sačuvati unos u promenljivu.
- Pomoću aritmetičkog operatora + spojiti sadržaje promenljivih za ime i prezime u treću promenljivu - ime i prezime.
- Prepraviti grananje tako da se sada proverava da li je ime i prezime jednako sa traženom osobom.

Zadatak 8

- Prepraviti prethodni zadatak tako da se ime i prezime formira spajanjem imena, zatim stringa sa jednim razmakom, a potom prezimena.



Zadatak 9

- Tražiti od korisnika unos celog broja koji predstavlja ostvarene bodove na ispitu.
- Pomoću naredbe grananja i šablona za formiranje konačne ocene datom na **stranici predmeta** odrediti postignutu ocenu.
- Štampati postignutu ocenu na konzolu.

Zadatak 10

- Tražiti od korisnika unos studijskog programa (npr. SII, IT, TH, PE).
- Upotrebom `match-case` izraza u zavisnosti od izabranog smera ispisati šta se testira na prijemnom ispitu.
- U slučaju SII i IT se rade 3 testa: inteligencije, opšte informisanosti i informatički test, i intervju; u slučaju PE intervju na srpskom jeziku; u slučaju TH intervju na engleskom jeziku.

Zadatak 11

- Rešiti prethodni zadatak upotrebom `if-else` izraza.



Sadržaj

① Operatori

- Aritmetički operatori
- Operatori poređenja
- Logički operatori
- Operatori nad bitovima
- Operatori pripadnosti
- Operatori identifikacije

② Grananje

③ Zadaci

④ Literatura



Literatura

- Knjige:
 - Osnove programiranja - Python, Vladislav Miškovic.
 - Illustrated guide to Python 3, Matt Harrison.
- Web stranice:
 - Python dokumentacija.
- Materijali sa predavanja i vežbi.

