

# Vježba 2

Programiranje 1 (550) - ak. god. 2024./2025.

## Logički operatori

Logički operatori uspoređuju logičke (Boolean) izraze i rezultiraju logičkom vrijednošću. Oni su nam potrebni za pisanje kompleksnijih logičkih izraza koje koristimo za zadavanje uvjeta odlučivanja u današnjoj vježbi (ali i u ostalim vježbama). Pregled logičkih operatora dan je u donjoj tablici.

VB.NET operator	Opis
Not	Negacija - vraća logički suprotnu vrijednost od ulaza
And ili AndAlso	Konjunkcija - vraća True ako su oba ulaza True
Or ili OrElse	Disjunkcija - vraća True ako je barem jedan ulaz True
Xor	Ekskluzivna disjunkcija - vraća True ako su ulazi različiti

Primjetite da uz operatore And i Or postoje i njihove takozvane „kratkospojne” verzije AndAlso i OrElse. To su bolje optimizirane verzije tih operatora na način da se, za operator AndAlso, ako je prvi ulaz False drugi ni ne razmatra, jer ne utječe na rezultat. Slično je i za operator OrElse, gdje, ako je prvi ulaz True, drugi ulaz se ne razmatra jer ne utječe na rezultat, što rezultira većom brzinom izvođenja programa ako imamo kompleksne logičke izraze koje je potrebno evaluirati.

## Operatori usporedbe

Operatore usporedbe koristimo za usporedbu dvije vrijednosti, a njihov rezultat je uvijek tipa Boolean, tj. True ili False.

Ako se koriste kod numeričkih tipova podataka (brojeva), funkcioniraju na klasični način. Upotreba nad kompleksnijim tipovima podataka može biti drukčija, o čemu će biti riječi u nekoj od idućih vježbi.

Operator	Opis
=	Operator jednakosti
<>	Operator nejednakosti
>	Veći
<	Manji

Operator	Opis
>=	Veći ili jednak
<=	Manji ili jednak

### Ostali operatori usporedbe

Osim navedenih, postoji još nekoliko operatora usporedbe koji se mogu koristiti: `Is`, `IsNot`. Oni se koriste za usporedbe objekata i bit će prezentirani naknadno.

## Aritmetički operatori

Aritmetički operatori se koriste za izvođenje matematičkih operacija nad podacima brojčanog tipa i dani su u sljedećoj tablici:

Operator	Opis	Napomena
+	Zbrajanje	
-	Oduzimanje	
*	Množenje	
/	Dijeljenje	Rezultat je <i>floating point</i> broj
\	Cjelobrojno dijeljenje	Rezultat je cijeli broj, decimalni dio se odbacuje
^	Potenciranje	
MOD	Modulus	Rezultat je ostatak od dijeljenja

## Operatori pomicanja bitova

Bitovni operatori se koriste za **pomicanje bitova** u lijevo ili u desno za određeni broj mjesta unutar nekog **cjelobrojnog tipa**. Zbog svojstava binarnog sustava, pomicanje bitova u lijevo za jedno mjesto je ekvivalentno množenju s 2, za dva mjesta, množenju s 4, itd. Dok je pomicanje bitova u desno ekvivalent dijeljenju s 2, s 4, itd.

Operator	Opis
<<	Pomicanje u lijevo
>>	Pomicanje u desno

Ovi operatori se u načelu izvode brže od aritmetičkih operatora množenja i dijeljenja.

## Bitovni logički operatori

Bitovni logički operatori rukuju s pojedinačnim bitovima unutar binarnog zapisa nekog cijelog broja. Oni primjenjuju dani logički operator (NOT, AND, OR, XOR) na svaki par bitova iz zadanog para cijelih brojeva, kao u sljedećem primjeru:

## VB

```
1 Dim a, b As Integer
2 a = 148
3 b = 37
4 Console.WriteLine(a And b) 'Ispis je 4
```

Operacija iz gornjeg primjera funkcionira tako da se prvo uzme binarni zapis brojeva pohranjenih u varijablama `a` i `b`, te zatim primjeni zadani logički operator na svaki bit tog broja:

$$\begin{array}{r} 10010100 \\ \text{And } 00100101 \\ \hline 00000100 \end{array}$$

Ovi operatori se mogu koristiti primjerice za provjeru ima li neki točno određeni bit unutar binarnog zapisa broja određenu vrijednost ili primjerice za provjeru parnosti broja, te za postavljanje određenih bitova unutar binarnog zapisa nekog broja na određenu vrijednost.

Sljedeći primjer pokazuje kako možemo postaviti određene bitove unutar zapisa broja u varijabli `broj`. Ova naredba će postaviti bitove  $2^0$  i  $2^2$  u stanje 1.

## VB

```
1 broj = broj Or 5
```

U sljedećem primjeru, naredbu `And` možemo iskoristiti ta bismo provjerili je li bit na poziciji  $2^1$  ima vrijednost 1:

## VB

```
1 Console.WriteLine(broj And 2)
```

Ako ispis bude jednak 2, to znači da je testirani bit ( $2^1$ ) u jedinici (jer je  $2^1=2$ ). Sljedeća naredba će provjeriti jesu li bitovi na pozicijama  $2^0$  i  $2^2$  u jedinici:

## VB

```
1 Console.WriteLine(broj And 5)
```

Drugi operand iznosi 5 jer je  $2^0 + 2^2 = 5$ . Iz istog razloga, ako su oba testirana bita u jedinici, rezultat će biti 5, ako je samo  $2^0$  u jedinici rezultat će biti 1 ( $2^0 = 1$ ), dok ako je samo  $2^2$  u

jedinici, rezultat će biti 4 ( $2^2 = 4$ ). Ako nijedan od ta dva testirana bita nije u jedinici, rezultat će biti 0.

## Odlučivanje

### IF-ELSE programska struktura

U VB.NET programskom jeziku, osnovna struktura za odlučivanje je IF-ELSE. Ona ima nekoliko varijanti, a osnovna sintaksa je sljedeća:

#### VB

```
1  If uvjet Then
2      ' ovdje dolazi kod koji se izvršava
3      ' ako je uvjet ispunjen
4  End If
```

Kod koji se nalazi unutar `If` naredbe se izvršava samo ako je zadani uvjet ispunjen, tj. ako je rezultat njegove evaluacije `True`. Uvjet može biti logička konstanta tipa Boolean, ali može biti i kompleksniji logički izraz.

Primjer:

#### VB

```
1  Dim broj As Integer = 17
2  If broj > 15 AndAlso broj < 20 Then
3      Console.WriteLine("Broj {0} je veći od 15, a manji od 20",
4      broj)
5  End If
```

U gornjem primjeru, poruka se ispisa ako je broj koji se provjerava veći od 15 i manji od 20. Tu je uvjet kompleksan i sastoji se od dva logička izraza povezana logičkim operatorom.

Ovu strukturu je moguće poopćiti tako da se može zadati i uvjet što se treba napraviti „inače” (ELSE), ako uvjet nije ispunjen:

#### VB

```
1  If uvjet Then
2      ' ovdje dolazi kod koji se izvršava
3      ' ako je uvjet ispunjen
4  Else
5      ' ovdje dolazi kod koji se izvršava
6      ' ako uvjet NIJE ispunjen
7  End If
```

Struktura se također može koristiti i ako postoji više uvjeta koje možemo provjeriti:

## VB

```
1  If uvjet1 Then
2      ' ovdje dolazi kod koji se izvršava
3      ' ako je uvjet1 ispunjen
4  ElseIf uvjet2 Then
5      ' ovdje dolazi kod koji se izvršava
6      ' ako <uvjet1> NIJE ispunjen, a <uvjet2> ispunjen
7  Else
8      ' ovdje dolazi kod koji se izvršava
9      ' ako niti jedan od uvjeta NIJE ispunjen
10 End If
```

Na analogan način je moguće definirati strukturu s proizvoljnim brojem uvjeta koje treba provjeriti. To ponekad može biti nepraktično i nepregledno, pa uskače i `Select Case` struktura.

## SELECT CASE programska struktura

`VB Select Case` programska struktura se koristi da bismo izvršili jedan od niza slučajeva, u ovisnosti o vrijednosti jednog izraza kojeg promatramo:

## VB

```
1  Select Case izraz
2      Case prvaVrijednost
3          ' izvršava se ako je
4          ' izraz=prvaVrijednost
5      Case drugaVrijednost
6          ' izvršava se ako je
7          ' izraz=drugaVrijednost
8      ...
9      Case Else
10         ' izvršava se ako izraz nije
11         ' jednak nekoj od ponuđenih vrijednosti
12 End Select
```

Također, moguće je grupiranje nekoliko slučajeva istovremeno, kao što je ilustrirano u sljedećem primjeru.

## VB

```
1  Dim broj as Integer = 8
2  Select Case broj
3      Case 1 To 5
```

```

4           Console.WriteLine("Broj je u rasponu [1,5] ")
5       Case 6, 7, 8
6           Console.WriteLine("Broj iznosi 6, 7 ili 8")
7       Case 9
8           Console.WriteLine("Broj ima vrijednost 9")
9       Case Else
10          Console.WriteLine("Broj je manji od 1 ili je veći od
11          9")
11      End Select

```

U gornjem primjeru, uz zadani broj koji ima vrijednost 8, izvršit će se slučaj koji uključuje broj 8 i ispisati odgovarajuća poruka.

## Zadaci

1. Napišite program u kojem korisnik putem konzole unosi tri broja, te se ispisuje najveći od njih.
2. Učitati 3 troznamenkasta broja. Ispisati broj koji sadrži najveću znamenku.
3. Korisnik putem konzole unosi cijeli broj. Ispišite sadrži li njegov binarni zapis bit težine 8. Ako ne sadrži, taj bit postavite u vrijednost 1. Ispišite dobiveni broj.
4. Napišite program u kojem korisnik putem konzole unosi cijeli broj. Potom provjerite je li broj paran. Ako je paran ispišite odgovarajuću poruku, a ako nije, provjerite je li broj djeljiv s 3. Ispišite odgovarajuću poruku u ovisnosti o rezultatu.
5. Napišite program koji će od korisnika tražiti unos veličine stranica trokuta. Provjerom unesenih stranica potrebno je ispisati o kojoj vrsti trokuta se radi.
6. Napišite program u kojem korisnik unosi broj putem konzole. Potom provjerite je li uneseni broj djeljiv sa 7, 11 i 13. Ispišite odgovarajuće poruke za svaku djeljivost ili jedinstvenu poruku ako nije djeljiv s niti jednim od navedenih brojeva.
7. Napišite program za kalkulator. Korisnik unosi dva broja i operator. U ovisnosti koji operator je odabran, treba izvršiti operaciju i ispisati rezultat. Ako unos operatora nije valjan, ispišite poruku o grešci.