

Tema 3

OPERATOR	KATEGORIJA	PRIMER IZRAZA	REZULTAT
~	Unarni	~a	komplementiranje (invertovanje) svakog bita (~0=1, ~1=0)
&	Binarni	a&b;	konjunkcija pojedinačnih bitova dva navedena argumenta (0&0 = 0, 0&1 = 0, 1&0 = 0, 1&1 = 1)
	Binarni	a b	disjunkcija pojedinačnih bitova dva navedena argumenta (0 0 = 0, 0 1 = 1, 1 0 = 1, 1 1 = 1)
^	Binarni	a^ b	eksluzivna disjunkcija pojedinačnih bitova dva navedena argumenta (0^0 = 0, 0^1 = 1, 1^0 = 1, 1^1 = 0)
«	Unarni	a << 1	pomeranje (šiftovanje) bitova ulevo (u primeru za jedno mesto);
»	Unarni	a >> 1	pomeranje (šiftovanje) bitova udesno (u primeru za jedno mesto).

PRIMER 1 - operatori nad bitovima & | ^ ~

```
using System;
namespace bitovi1
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 1;
            int y = 1;

            Console.WriteLine(x&y);
            Console.WriteLine(x|y);
            Console.WriteLine(x^y);
        }
    }
}
```

NAPOMENA 1: Pokazati ove rezultate i kada se promene x i y da budu 0 i 1, sve cetiri kombinacije prikazati.

PRIMER 2 - OPERATORI pomeranja

```
using System;
namespace bitovi1
{
    class Program
    {
```

```

static void Main(string[] args)
{
    int x = 1;
    int y = x << 4;
    Console.WriteLine(y);
    x = y >> 2;
    Console.WriteLine(x);
    y = x >> 5;
    Console.WriteLine(y);
}
}
}

```

NAPOMENA 1: kad se pomeri tako da izpadne broj, ostaje 0. Ispisati i objasniti ovo pomjeranje na tabli.

PRIMER 3 - OPERATORI svi nad bitovima

```

using System;
namespace bitovil
{
    class Program
    {
        static void Main(string[] args)
        {
            Byte x = 3;
            Byte y = 5;

            Console.WriteLine(x | y);
            Console.WriteLine(x & y);
            Console.WriteLine(x ^ y);
            Console.WriteLine(~x & y);
            Console.WriteLine(x << 1);
            Console.WriteLine(x << 2);
            Console.WriteLine(x >> 1);
        }
    }
}

```

```

byte a = 3; // 0000 0011 = 3
byte b = 5; // 0000 0101 = 5
Console.WriteLine(a | b); // 0000 0111 = 7
Console.WriteLine(a & b); // 0000 0001 = 1
Console.WriteLine(a ^ b); // 0000 0110 = 6
Console.WriteLine(~a & b); // 0000 0100 = 4
Console.WriteLine(a << 1); // 0000 0110 = 6
Console.WriteLine(a << 2); // 0000 1100 = 12
Console.WriteLine(a >> 1); // 0000 0001 = 1

```