

\*\*\*\*\*

# Tema 3

\*\*\*\*\*

## TEME

- Naredba grananja
  - Logicki izrazi imaju poredjenje koristenjem: <, <=, >, >=, ==, !=
  - Logicki operatori se koriste za kombinovanje vise log izraza: && (and) i || (or)
  - Sintaksa if naredbe
  - Sintaksa ternarnog izraza
  - Sintaksa switch naredbe
  - Koristiti viticaste zagrade, ako imamo vise naredbi koji se desavaju unutar jedne grupe uslova
- Hijerarhije uslova i nacin pravljenja slucajeva koji zadovoljavaju odredjena pravila zadatka

### PRIMER 1: Proveriti za dva uneta cela broja da li su istog znaka. // OBAVEZAN PRIMER ZA IF

#### Korak 1:

```
bool istogZnaka;
if (a > 0)
    if (b > 0)
        istogZnaka = true;
    else // b < 0
        istogZnaka = false;
else // a < 0
    if (b > 0)
        istogZnaka = false;
    else // b < 0
        istogZnaka = true;
if (istogZnaka)
    cout << "da" << endl;
else
    cout << "ne" << endl;
```

#### Korak 2: krace, log operatori

```
bool istogZnaka;
if ((a > 0) && (b > 0)) || ((a < 0) && (b < 0))
    istogZnaka = true;
else
    istogZnaka = false;
```

#### Korak 3: ternarni izraz

```
bool istogZnaka;
((a>0)&&(b>0))||((a<0)&&(b<0)) ? istogZnaka=true : istogZnaka=false;
```

## OBJAŠNENJE:

- Napraviti paralelu između pokrivanja svih slučajeva u dužim i u kraćim oblicima implementacije

### PRIMER 2: Detekcija broja dana u mesecu // **OBAVEZAN PRIMER - osnovni za switch**

```
// proverava da li je data godina prestupna
bool prestupna(int godina) {
// godina je prestupna ako je deljiva sa 4 i nije deljiva sa 100,
// ili ako je deljiva sa 400
    return (godina % 4 == 0 && godina % 100 != 0) || (godina % 400 == 0);
}

int main() {
    // učitavamo mesec i godinu
    int mesec, godina;
    cin >> mesec >> godina;
    // određujemo broj dana u tom mesecu
    int brojDana = 0;
    switch(mesec) {
        // januar, mart, maj, jul, avgust, oktobar, decembar
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            brojDana = 31;
            break;
        // april, jun, septembar, novembar
        case 4: case 6: case 9: case 11:
            brojDana = 30;
            break;
        // februar
        case 2:
            brojDana = prestupna(godina) ? 29 : 28;
            break;
    }
    // ispisujemo rezultat
    cout << brojDana << endl;
    return 0;
}
```

## OBJAŠNENJE:

- Napraviti paralelu između pokrivanja svih slučajeva u dužim i u kraćim oblicima implementacije

## BONUS PRIMERI - NE MORAJU DA SE RADE AKO NEMA VREMENA, ZADATI ZA VEZBU

**PRIMER 1:** Jucerasnji datum na osnovu uneta tri broja koji predstavljaju dan, mesec i godinu.

13 2016 -> 29 2 2016

13 1900 -> 28 2 1900

11 2014 -> 31 12 2013

### RESENJE:

Imati na umu razvrstavanje slucajeva

- U vecini slucajeva je da se smanji samo dan za 1
- Specijalan slucaj je ako je danasnji datum bio 1
  - Tada je prethodni 31, 30, 29, 28
- U skladu sa tim proveravati i mesece i godinu

### KOD:

```
#include <iostream>

using namespace std;

// proverava da li je data godina prestupna
bool prestupna(int godina) {
    // godina je prestupna ako je deljiva sa 4 i nije deljiva sa 100,
    // ili ako je deljiva sa 400
    return (godina % 4 == 0 && godina % 100 != 0) || (godina % 400 == 0);
}

int brojDanaUMesecu(int m, int g) {
    int brojDana;
    switch(m) {
        // januar, mart, maj, jul, avgust, oktobar, decembar
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            brojDana = 31;
            break;
        // april, jun, septembar, novembar
        case 4: case 6: case 9: case 11:
            brojDana = 30;
            break;
        // februar
        case 2:
            brojDana = prestupna(g) ? 29 : 28;
            break;
    }
}
```

```

return brojDana;
}

int main() {
// učitavamo mesec i godinu
int dan, mesec, godina;
int noviDan, noviMesec, novaGodina;

cin >> dan >> mesec >> godina;
// odredjujemo broj dana u tom mesecu

if (dan != 1) {
    noviDan -= 1;
    noviMesec = mesec;
    novaGodina = godina;
} else {
    // dan je 1
    // novidan je ili 28 ili 29 ili 30 ili 31

    if (mesec == 1) {
        noviMesec = 12;
        novaGodina = godina - 1;

    } else {
        noviMesec = mesec - 1;
        novaGodina = godina;
    }
    noviDan = brojDanaUMesecu(noviMesec, novaGodina);
}
// ispisujemo rezultat
cout << noviDan << " " << noviMesec << " " << novaGodina << endl;
return 0;
}

```

### **OBJAŠNENJE:**

- Imati na umu redosled uslova koji se koriste

### **PRIMER 2:** Ishod seta na osnovu dobijenih gemova

7 5 -> pobedio prvi

3 7 -> neispravno

5 7 -> pobedio drugi

1 6 -> pobedio drugi

4 4 -> nije završeno

-2 3 -> neispravan unos

## RESENJE:

```
#include <iostream>

using namespace std;

void pobednik(int p, int d) {
    string tekst = p>d ? "prvi" : "drugi";
    cout << "Pobednik je " << tekst << endl;
}

int main() {

    int g1, g2;
    cin >> g1 >> g2;
    // Dva slucaja: u toku i nije u toku

    // sta nije dozvoljeno
    // if g1 ili g2 > 7
    if (g1 > 7 || g2 > 7 || g1 < 0 || g2 < 0 || (g1==7 && g2 < 5) || (g2==7
&& g1 < 5))
        cout << "Neispravan unos!" << endl;
    else if ((g1 == 6 and g2 < 5) || (g2 == 6 and g1 < 5) || (g1 == 7 and
g2 == 5) || (g1 == 5 and g2 == 7)){
        // regularan kraj utakmice
        cout << "Regular kraj utakmice!" << endl;
        pobednik(g1, g2);
    }
    else if ((g1 == 7 and g2 == 6) || (g1 == 6 and g2 == 7)) {
        // kraj u taj brejku
        cout << "Kraj utakmice u taj brejku!" << endl;
        pobednik(g1, g2);
    }
    else {
        // nije kraj utakmice
        cout << "Utakmica jos traje!" << endl;
    }
    return 0;

}
```

## OBJAŠNENJE:

- Odvojiti sve posebne slucajeve i onda samo u pravilnim ishodima pozvati funkciju pobednik