Природно-математички факултет
Универзитет у Бањој Луци

Итеративне наредбе

Аутори: Драган Матић, Милана Грбић,
Милан Предојевић, Ненад Вилендечић

2025

Садржај

Понављање

Гранање (`If-else`)

If-else структура омогућава доношење одлука у C++ програмима. Основни облик ‘if-else‘ израза је:

```
if (uslov) {
    // Ako je uslov tacan, izvrsi ovaj blok koda
} else {
    // Ako je uslov netacan, izvrsi ovaj blok koda
}
```

Пример коришћења

Следећи пример демонстрира употребу ‘if-else‘ структуре у C++:

```
#include <iostream>

int main() {
    int broj;
    std::cout << "Unesite broj: ";
    std::cin >> broj;

    if (broj % 2 == 0) {
        std::cout << "Broj je paran" << std::endl;
    } else {
        std::cout << "Broj je neparan." << std::endl;
    }
    return 0;
}
```

Задаци

1. Задатак: Максимум два броја

Написати програм који за унесена два реална броја исписује њихов максимум.

Примјер извршавања

```
Unesite prvi realan broj: 5
Unesite drugi realan broj: 12
Maksimum unesenih brojeva je: 12
```

Рјешење

```cpp
#include <iostream>

using namespace std;

int main() {
    // Deklaracija promenljivih
    double broj1, broj2;

    // Unos brojeva
    cout << "Unesite prvi realan broj: ";
    cin >> broj1;

    cout << "Unesite drugi realan broj: ";
    cin >> broj2;

    // Odredjivanje i ispis maksimuma
    if (broj1 >= broj2) {
        cout << "Maksimum unesenih brojeva je: " << broj1 << endl;
    } else {
        cout << "Maksimum unesenih brojeva je: " << broj2 << endl;
    }

    return 0;
}
```

2. Задатак: Сума позитивних бројева

Написати програм који учитава три цијела броја са тастатуре и исписује збир позитивних унесених бројева.

Примјер извршавања

```
Unesite prvi broj: 2
Unesite drugi broj: -3
Unesite treci broj: 9
Zbir pozitivnih brojeva je: 11
```

Рјешење

```cpp
#include <iostream>

using namespace std;

int main() {
    // Deklaracija promjenljivih
    int broj1, broj2, broj3;
    int zbir = 0;

    // Unos brojeva
    cout << "Unesite prvi broj: ";
    cin >> broj1;

    cout << "Unesite drugi broj: ";
    cin >> broj2;

    cout << "Unesite treci broj: ";
    cin >> broj3;

    // Provjera i racunanje zbira pozitivnih brojeva
    if (broj1 > 0) {
        zbir += broj1;
    }
    if (broj2 > 0) {
        zbir += broj2;
    }
    if (broj3 > 0) {
        zbir += broj3;
    }

    // Ispis rezultata
    cout << "Zbir pozitivnih brojeva je: " << zbir << endl;

    return 0;
}
```

Увод

Итеративне наредбе, познате као петље, омогућавају извршавање истог блока кода више пута док је неки услов испуњен. У C++ постоје три главне итеративне наредбе:

- **for** петља
- **while** петља
- **do-while** петља

Свака од њих се користи у различитим ситуацијама, у зависности од услова понављања.

`for` петља

Користи се када знамо **тачан број понављања**. Синтакса је:

```
for (<<inicijalizacija>>; <<uslov>>; <<inkrement/dekrement>>) {
  // Tijelo petlje
}
```

Пример: Испис бројева од 1 до 5

```
#include <iostream>
using namespace std;

int main() {
    for (int i = 1; i <= 5; i++) {
        cout << i << " ";
    }
    return 0;
}
```

`while` петља

Користи се када **не знамо унапријед** колико ће се пута извршити неки блок.

```
while (<<uslov>>) {
    // Tijelo petlje
}
```

Пример: Испис бројева од 1 до 5

```
#include <iostream>
using namespace std;

int main() {
```

```
    int i = 1;
    while (i <= 5) {
        cout << i << " ";
        i++;
    }
    return 0;
}
```

do-while петља

Ова петља **увек изврши бар једну итерацију** јер се услов провјерава након прве итерације.

```
do {
    // tijelo petlje
} while (<<uslov>>);
```

Пример: Испис бројева од 1 до 5

```
#include <iostream>
using namespace std;

int main() {
    int i = 1;
    do {
        cout << i << " ";
        i++;
    } while (i <= 5);

    return 0;
}
```

Закључак

Петља	Када се користи?
`for`	Када знамо број понављања
`while`	Када не знамо број понављања унапред
`do-while`	Када је потребно да се изврши бар једном

Задаци

1. Задатак: Дјелиоци природног броја

Прави дјелиоци природног броја су сви дјелиоци осим јединице и самог тог броја. Написати програм који за унесен природан број n исписује све његове праве дјелиоце. У случају неисправног уноса, исписати одговарајућу поруку о грешци.

Примјер извршавања

```
Unesi prirodan broj: 18
Pravi djelitelji broja 18 su: 2 3 6 9
```

Изузеци

- Уколико корисник не унесе пророданан број прекинути извршење и исписати одговарајућу поруку.
- Уколико је унесени број прост исписати одговарајућу поруку.

Рјешење

```cpp
#include <iostream>
using namespace std;

int main() {
    int n;
    cout << "Unesi prirodan broj: ";
    cin >> n;

    if (n <= 0) {
        cout << "Greska: Neispravan unos." << endl;
        return 1;
    }

    cout << "Pravi djelitelji broja " << n << " su: ";
    bool imaDjelitlje = false;
    for (int i = 2; i <= n / 2; i++) {
        if (n % i == 0) {
            cout << i << " ";
            imaDjelitlje = true;
        }
    }

    if (!imaDjelitlje) {
        cout << "(Nema pravih djelitelja, broj je prost)";
    }

    cout << endl;
    return 0;
}
```

2. Задатак: Унос до -1

Написати програм који од корисника захтијева унос цијелих бројева све док корисник не унесе -1. Након што корисник унесе -1 програм треба да испише суму унесених бројева.

Примјер извршавања

```
Unesi cijele brojeve (-1 za kraj):
4
4
12
-1
Suma unesenih brojeva: 20
```

Рјешење

```cpp
#include <iostream>
using namespace std;

int main() {
    int broj, suma = 0;

    cout << "Unesi cijeli broj: ";
    cin >> broj;

    while (broj != 0){
        suma += broj;
        cout << "Unesi cijeli broj: ";
        cin >> broj;
    }

    cout << "Suma unesenih brojeva: " << suma << endl;

    return 0;
}
```

3. Задатак: Уклањање нула са десне стране

Написати програм који за унесени цијели број исписује број добијен уклањањем свих нула са десне стране унесеног броја.

Примјер извршавања

```
Unesi cijeli broj: 455100
Broj bez nula sa desne strane: 4551
```

Рјешење

```cpp
#include <iostream>
using namespace std;

int main() {
    int broj;

    cout << "Unesi cijeli broj: ";
    cin >> broj;

    if (broj == 0) {
        cout << "Rezultat: 0" << endl;
        return 0;
    }

    while (broj % 10 == 0) {
        broj /= 10;
    }

    cout << "Broj bez nula sa desne strane: " << broj << endl;

    return 0;
}
```

Домаћи

1. Задатак

Написати програм који учитава позитиван цијели број n, а потом и n цијелих бројева. Израчунати и исписати збир оних бројева који су истовремено непарни и негативни. У случају неисправног уноса, исписати одговарајућу поруку о грешци.

Примјер извршавања

```
Unesite broj n: 5
Unesite n brojeva:
1
-5
-6
3
-11
Zbir neparnih i negativnih: -16
```

2. Задатак

Написати програм који учитава цијели број и исписује његове цифре у обрнутом поретку.

3. Задатак

Написати програм који за унесени цијели број провјерава и исписује да ли се цифра 5 налази у његовом запису.