

Programiranje 1

Programski jezik C

4. čas

Mirko Spasić

SWITCH

- Ova naredba se još zove i naredbom višestrukog grananja
- Njome proveravamo da li je neki izraz jednak nekoj od više konstantnih celobrojnih vrednosti, i u zavisnosti od toga, izvršavaju se određene akcije
- Ovom naredbom možemo da zamenimo višestruko pojavljivanje naredbe if

SWITCH

- Opšti oblik naredbe je:

```
switch (izraz) {  
    case konstantan_izraz1: naredbe1  
    case konstantan_izraz2: naredbe2  
    ...  
    default: naredbe  
}
```

SWITCH

- Svaki konstantni izraz predstavlja određeni slučaj. Ako neki slučaj odgovara vrednosti izraza, izvršavanje počinje od tog slučaja, tj. izvršavaju se sve naredbe koje odgovaraju tom slučaju, **ali i sve ostale** koje odgovaraju slučajevima navedenim posle tog slučaja (obratiti pažnju na **break** u svakom slučaju)
- Svi izrazi slučajeva moraju biti različiti

SWITCH

- Slučaj default ako se navodi, mora se navesti poslednji. Naredbe koje se navedu u ovom slučaju se izvršavaju ako se izraz u uslovu ne poklapa ni sa jednim prethodno navedenim slučajem
- Ako se default ne navede, i ako se izraz u uslovu ne poklapa ni sa jednim slučajem, onda se ne izvršavaju nikakve naredbe

SWITCH - primer

```
#include<stdio.h>
int main() {
    int n;

    printf("Unesi paran broj manji od 10\n");
    scanf("%d",&n);

    switch(n)
    {
        case 0:
            printf("Uneli ste nulu\n");
            break;
        case 2:
            printf("Uneli ste dvojku\n");
            break;

```

```
        case 4:
            printf("Uneli ste cetvorku\n");
            break;
        case 6:
            printf("Uneli ste sesticu\n");
            break;
        case 8:
            printf("Uneli ste osmicu\n");
            break;
        default:
            printf("Pogrešan unos!\n");
    }

    return 0;
}
```

SWITCH - primer

- *Napisati program koji vrši brojanje pojavljivanja karaktera a, b i c korišćenjem switch naredbe.*

```
#include <stdio.h>
```

```
int main() {
```

```
    int c;
```

```
    int br_a=0, br_b=0, br_c=0;
```

SWITCH - primer

```
while ( ( c = getchar() ) != EOF ) {  
    switch(c) {  
        /* Obratiti paznju da nije case 0: niti case '0'; */  
        case 'a':  
            br_a++;  
            break; /* Isprobati veziju bez break */  
        case 'b':  
            br_b++;  
            break;  
        case 'c':  
            br_c++;  
            break;  
    }  
}  
printf("Br a : %d\nBr b : %d\nBr c : %d\n",br_a, br_b, br_c);  
return 0;  
}
```


SWITCH - primer

- Jednostavan kalkulator

Program učitava komandu sa ulaza (u vidu jednog karaktera) i na osnovu te komande učitava dva broja nad kojima izvršava odgovarajuću operaciju.

Ovaj postupak se ponavlja dok se ne unese komanda za kraj programa.

*Program ilustruje primenu **do-while** petlje i **switch** naredbe za implementaciju menija, za potrebe interaktivnog rada.*

SWITCH - primer

```
#include <stdio.h>
```

```
int main() {  
    char c;  
    int a,b;
```

```
    /*
```

Ova petlja predstavlja tipican nacin za implementaciju interaktivnog menija:

u svakoj iteraciji se ispisuje meni, a od nas se ocekuje da unesemo karakter za komandu.

U zavisnosti od unetog karaktera se preduzima odgovarajuca akcija (u ovom slucaju unos dva broja i prikaz rezultata odgovarajuce operacije).

Petlja se izvrsava sve dok se ne unese komanda za kraj ('q' u ovom slucaju).

```
    */
```

```
do {
```

```
    /* Prikaz menija */
```

```
    printf("Unesite:\n");
```

```
    printf("\t'a' za sabiranje,\n\t's' za oduzimanje,\n");
```

```
    printf("\t'm' za mnozenje,\n\t'd' za deljenje,\n");
```

```
    printf("\t'q' za kraj.\n");
```

```
    /* Ucitavamo komandu */
```

```
    scanf("%c", &c);
```

```
    /* Izbor akcije na osnovu komande koja je uneta */
```

```
    switch(c) {
```

```
    /* Slucaj sabiranja */
```

```
    case 'a':
```

```
        printf("Unesite dva cela broja: ");
```

```
        scanf("%d%d", &a, &b);
```

```
        printf("%d+%d=%d\n", a, b, a + b);
```

```
        break;
```

```
    /* Slucaj oduzimanja */
```

```
    case 's':
```

```
        printf("Unesite dva cela broja: ");
```

```
        scanf("%d%d", &a, &b);
```

```
        printf("%d-%d=%d\n", a, b, a - b);
```

```
        break;
```

SWITCH - primer

```
/* Slucaj mnozenja */
case 'm':
    printf("Unesite dva cela broja: ");
    scanf("%d%d", &a, &b);
    printf("%d*%d=%d\n", a, b, a *
b);
    break;
/* Slucaj deljenja */
case 'd':
    printf("Unesite dva cela broja: ");
    scanf("%d%d", &a, &b);
    printf("%d/%d=%d\n", a, b, a / b);
    break;
/* Slucaj izlaska -- ne radimo nista */
case 'q':
    break;
```

```
/* Slucaj pogresne komande
(sve ostalo sto nije navedeno gore) */
default:
    printf("Pogresna komanda!\n");
    break;
}
} while (c != 'q');
/* Iz petlje izlazimo kada je c == 'q' */
return 0;
}
```

USLOVNI IZRAZ

- Uslovni izrazi obezbeđuju alternativni način pisanja **if-else** konstrukcije.

- Tako se

```
if (izraz1)
    izraz2;
else
    izraz3;
```

može zapisati kao

```
izraz1 ? izraz2 : izraz3;
```

- Primer korišćenja uslovnog izraza:

```
min = (a<b)? a : b;
max = (a>b)? a : b;
```

USLOVNI IZRAZ - primer

- Brojanje malih i velikih slova

Program čita tekst sa ulaza, prevodi sva mala slova u velika, i pri tom broji mala i velika slova.

Program ilustruje operatore ',' (zarez), i '?:' (uslovni operator).

USLOVNI IZRAZ - primer

```
#include <stdio.h>
```

```
int main() {
```

```
    int c;
```

```
    int br_malih;
```

```
    int br_velikih;
```

```
    /* Prepisujemo ulaz na izlaz, uz promenu malih slova u velika.
```

```
    Takodje, brojimo mala i velika slova. */
```

```
    for (br_malih = 0, br_velikih = 0 ; ( c = getchar() ) != EOF ; ) {
```

```
        putchar(c >= 'a' && c <= 'z' ? c - 'a' + 'A' : c);
```

```
        if (c >= 'a' && c <= 'z')
```

```
            br_malih++;
```

```
        if (c >= 'A' && c <= 'Z')
```

```
            br_velikih++;
```

```
    }
```

```
    printf("Broj velikih slova: %d\nBroj malih slova: %d\n", br_velikih, br_malih);
```

```
    return 0;
```

```
}
```