

УуБ



ФОН



Е



лабси



па



Универзитет у Београду

факултет организационих наука

Катедра за софтверско инжењерство

лабораторија за софтверско инжењерство

ПРОГРАМИРАЊЕ I

ПРОГРАМИРАЊЕ I

Предавање #3

Проф. др **Саша Д. Лазаревић**, дипл. инж. инф.

sasa.lazarevic@fon.bg.ac.rs

Садржај предавања

Наредбе:

- 1) Израз
- 2) Блок
- 3) Избор: селекција или разграната управљачка структура
- 4) Понављање: итерација или циклична управљачка структура
- 5) Скок: безусловни пренос тока извршења програма

Наредбе

Наредбе одређују акције која треба да се изврше.

На пример:

- ▶ додела вредности променљивој,
- ▶ избор између две могућности,
- ▶ пренос управљања програмом на функцију,
- ▶ вишеструко понављање једног низа наредби,
- ▶ скок на одређену наредбу итд.

Садржај предавања

Наредбе:

- 1) **Израз**
- 2) Блок
- 3) Избор: селекција или разграната управљачка структура
- 4) Понављање: итерација или циклична управљачка структура
- 5) Скок: безусловни пренос тока извршења програма

Израз 1/2

- ❖ На крају наредбе типа израз (*expression statement*) налази се тачка-зarez:

<izraz>;

- ❖ Неколико типичних израза:

```
a = b;           // dodela vrednosti
zbir = sab_1 + sab_2; // sabiranje i dodela
++x;             // inkrementacija
printf("Zdravo!\n"); // poziv funkcije
```

Израз 2/2

- ❖ Ако је наредба позив функције, а резултат функције није потребан, онда се он (резултат) може експлицитно одбацити на следећи начин:

```
char ime[31];  
(void) strcpy(ime, "Pavle");
```

- ❖ Наредба се може састојати само од тачке-зареза; тада се она назива празна наредба (*null statement*). На пример, тело *for*-циклауса може да буде празна наредба:

```
for (i = 0; ime[i] != '\0'; ++i)  
    ;
```

Садржај предавања

Наредбе:

- 1) Израз
- 2) **Блок**
- 3) Избор: селекција или разграната управљачка структура
- 4) Понављање: итерација или циклична управљачка структура
- 5) Скок: безусловни пренос тока извршења програма

Блок

- ❖ Сложена наредба (*compound statement*) или блок (*block*) представља низ који се састоји од више наредби и/или декларација који су смештени између великих (витичастих) заграда, тако да чине једну наредбу:

`{ <niz deklaracija i naredbi> }`

- ❖ За разлику од простих наредби, блок се не завршава тачка-зарезом
- ❖ Декларације се обично налазе на почетку блока, пре свих наредби. Међутим, од стандарда C99 дозвољено је да декларација буде на произвољном месту у блоку
- ❖ Идентификатори декларисани у блоку имају блоковску област важења (*block scope*), тј. доступни су само од њихове декларације до краја блока

Садржај предавања

Наредбе:

- 1) Израз
- 2) Блок
- 3) **Избор: селекција или разграната управљачка структура**
- 4) Понављање: итерација или циклична управљачка структура
- 5) Скок: безусловни пренос тока извршења програма

Избор: селекција или разграната управљачка структура

- ❖ Наредба избора (*selection statement*) може преусмерити ток програма у различите правце зависно од датог услова
- ❖ У ПЈ Ц постоје две наредбе избора:
 - ▶ *if* наредба
 - ▶ *switch* наредба

Избор: *IF* наредба

- ❖ Синтакса (општи облик *if*-наредбе):

```
if ( <izraz> ) <naredba_1>  
[ else <naredba_2> ]
```

- ❖ Део **else** није обавезан
- ❖ Семантика: Прво се проверава вредност израза **<izraz>**, да би се одредило која од две наредбе мора да се изврши. Израз је увек скаларног типа. Ако је његова вредност *TRUE* (тј. различита од нуле) извршиће се **<naredba_1>**, иначе извршава се **<naredba_2>** (ако је наведена)
- ❖ Наредбе **<naredba_1>** и **<naredba_2>** могу бити сложене

Избор: *SWITCH* наредба 1/2

❖ Синтакса (општи облик *switch*-наредбе):

```
switch ( <izraz> ) {  
    case <konstantna_vrednost_1>:  
        <naredba_1>  
        break;  
    case <konstantna_vrednost_2>:  
        <naredba_2>  
        break;  
    ...  
    [ default:  
        <naredba_n> ]  
}
```

Избор: *SWITCH* наредба 2/2

- ❖ Семантика: Израз `<izraz>` је целобројног типа и израчунава се једном. Затим се добијена вредност израза проверава са константним вредностима у лабелама ***case***. Ако се вредност израза подудара са неким од константних вредности лабела, ток програма се премешта на наредбу која следи иза лабеле ***case***. Уколико се не подудара ни са једним од наведених константних вредности, програм наставља да се извршава од лабеле ***default*** (уколико постоји).
- ❖ Видети програмски код:

P03_01_if.c и ***P03_02_switch.c***

Садржај предавања

Наредбе:

- 1) Израз
- 2) Блок
- 3) Избор: селекција или разграната управљачка структура
- 4) **Понављање: итерација или циклична управљачка структура**
- 5) Скок: безусловни пренос тока извршења програма

Понављање: итерација или циклична управљачка структура

- ❖ Наредба понављања (*iteration statement*) омогућава вишестуко извршавање наредбе. Наредба може бити проста или сложена
- ❖ У ПЈ Ц постоје три наредбе понављања:
 - ▶ *while* наредба
 - ▶ *for* наредба и
 - ▶ *do-while* наредба
- ❖ Пример: Сумирати првих десет природних бројева.

```
int suma = 0, i = 1;
```

Постављање почетних вредности

```
while (i <= 10)
```

Услов циклуса: контролна променљива

```
{
```

```
    suma = suma + i;
```

Обрада података

```
    i = i + 1;
```

Модификација контролне променљиве

```
}
```

Почетак и крај тела циклуса

Понављање: итерација или циклична управљачка структура

- ❖ Синтакса и семантика цикличних управљачких структура обрађени су кроз примере. Видети програмски код:

`P03_03_while.c` и `P03_04_iteracija.c`

Садржај предавања

Наредбе:

- 1) Израз
- 2) Блок
- 3) Избор: селекција или разграната управљачка структура
- 4) Понављање: итерација или циклична управљачка структура
- 5) **Скок: безусловни пренос тока извршења програма**

Скок:

безусловни пренос тока извршења програма

❖ За домаћи:

- ▶ break
- ▶ continue
- ▶ goto
- ▶ return

Шта ћете радити на вежбама

Следеће предавање

Функције:

- 1) Дефиниције ϕ -ја
- 2) Декларације ϕ -ја
- 3) Извршавање функција
- 4) Показивачи као аргументи и резултати ϕ -ја
- 5) Уметнуте ϕ -је
- 6) Рекурзивне ϕ -је
- 7) Φ -је са неодређеним бројем параметара

УуБ



ФОН



Е

Е

лабСи



па



Универзитет у Београду

факултет организационих наука

Катедра за софтверско инжењерство

лабораторија за софтверско инжењерство

ПРОГРАМИРАЊЕ I

ПРОГРАМИРАЊЕ I

Предавање #3

Проф. др **Саша Д. Лазаревић**, дипл. инж. инф.

sasa.lazarevic@fon.bg.ac.rs

Универзитет,
факултет,
катедра,
лабораторија,
предмет

