Kontrola toka izvršenja

Vladimir Filipović

vladaf@matf.bg.ac.rs

vladaf@matf.bg.ac.rs

Naredbe i blokovi

```
x = 0;
i++;
printf (…);
```

```
{ }
```

```
[ ]
```

```
if, else, while, for    -    {  }
```

vladaf@matf.bg.ac.rs

If - else

```
if (izraz)
    naredba1
else
    naredba2
```

```
if (n > 0)
    if (a > b)
        z = a;
    else
        z = b;
```

⇔

```
if (n > 0)   {
    if (a > b)
        z = a;
}
else
    z = b;
```

```
if (n >= 0)
    for (i = 0; i < n; i++)
        if (s[i] > 0)    {
            printf("…");
            return i;
        }
else          /* POGREŠNO */
    printf("error - - n is negative\n");
```

```
if (a > b)
    z = a;
else
    z = b;
```

vladaf@matf.bg.ac.rs

Else - if

```
if (izraz)
    naredba
else  if (izraz)
    naredba
else  if (izraz)
    naredba
else  if (izraz)
    naredba
else
    naredba
```

```
/* binsearch:  find x in v[0] <= v[1] <= ... <= v[n-1] */
int binsearch(int x, int v[ ], int n)
{
    int low, high, mid;

    low = 0;
    high = n - 1;
    while (low <= high) {
        mid = (low+high)/2;
        if (x < v[mid])
            high = mid - 1;
        else if (x  > v[mid])
            low = mid + 1;
        else    /* found match */
            return mid;
    }
    return -1;   /* no match */
}
```

Switch

```
switch (izraz)  {
    case konst-izraz:  naredbe
    case konst-izraz:  naredbe
    default:  naredbe
}
```

```
#include <stdio.h>

main()  /* count digits, white space, others */
{
    int c, i, nwhite, nother, ndigit[10];

    nwhite = nother = 0;
    for (i = 0; i < 10; i++)
        ndigit[i] = 0;
    while ((c = getchar()) != EOF) {
        switch (c) {
        case '0': case '1': case '2': case '3': case '4':
        case '5': case '6': case '7': case '8': case '9':
            ndigit[c-'0']++;
            break;
        case ' ':
        case '\n':
        case '\t':
            nwhite++;
            break;
        default:
            nother++;
            break;
        }
    }
    printf("digits =");
    for (i = 0; i < 10; i++)
        printf(" %d", ndigit[i]);
    printf(", white space = %d, other = %d\n",
        nwhite, nother);
    return 0;
}
```



vladaf@matf.bg.ac.rs

Petlje - while i for

```
while (izraz)
    naredba
```

```
for (izraz1 ; izraz2 ; izraz3)
    naredba
```

⟺

```
izraz1 ;
while (izraz2)  {
    naredba
    izraz3 ;
}
```

```
for ( ; ; ) {
    …
}
```

```
while ((c = getchar()) == ' ' || c == '\n' || c == '\t')
    ;   /* preskočiti znake praznog prostora */
```

```
for (i = 0; i < n; i++)
    …
```

```
#include <ctype.h>

/* atoi:  convert s to integer; version 2 */
int atoi(char s[ ])
{
    int i, n, sign;

    for (i = 0; isspace(s[i]); i++)        /* skip white space */
        ;
    sign = (s[i] == '-') ? -1 : 1;
    if (s[i] == '+' || s[i] == '-')        /* skip sign */
        i++;
    for (n = 0; isdigit(s[i]); i++)
        n = 10 * n + (s[i] - '0');
    return sign * n;
}
```

```
/* shellsort:  sort v[0]...v[n-1] into increasing order */
void shellsort(int v[ ], int n)
{
    int gap, i, j, temp;

    for (gap = n/2; gap > 0; gap /= 2)
        for (i = gap; i < n; i++)
            for (j=i-gap; j>=0 && v[j]>v[j+gap]; j-=gap) {
                temp = v[j];
                v[j] = v[j+gap];
                v[j+gap] = temp;
            }
}
```

```
#include <string.h>

/* reverse:  reverse string s in place */
void reverse(char s[])
{
    int c, i, j;

    for (i = 0, j = strlen(s)-1; i < j; i++, j--) {
        c = s[i];
        s[i] = s[j];
        s[j] = c;
    }
}
```

```
for (i = 0, j = strlen(s)-1; i < j; i++, j--)
    c = s[i], s[i] = s[j], s[j] = c;
```



Petlje - do-while

do
    naredba
while (*izraz*) ;

```
/* itoa:  convert n to characters in s */
void itoa(int n, char s[ ])
{
    int i, sign;

    if ((sign = n) < 0)                 /* record sign */
        n = -n;                         /* make n positive */
    i = 0;
    do {                                /* generate digits in reverse order */
        s[i++] = n % 10 + '0';          /* get next digit */
    } while ((n /= 10) > 0);            /* delete it */
    if (sign < 0)
        s[i++] = '-';
    s[i] = '\0';
    reverse(s);
}
```



vladaf@matf.bg.ac.rs

Naredbe break i continue

```
/* trim:  remove trailing blanks, tabs, newlines */
int trim(char s[])
{
    int n;

    for (n = strlen(s)-1; n >= 0; n--)
        if (s[n] != ' ' && s[n] != '\t' && s[n] != '\n')
            break;
    s[n+1] = '\0';
    return n;
}
```

```
for (i = 0; i < n; i++) {
            if (a[i] < 0)   /* preskočiti negativne elemente */
                continue;
            … /* obraditi nenegativne elemente */
}
```

vladaf@matf.bg.ac.rs

Naredba goto i oznake

```
for ( … )
    for ( … )  {
        …
        if (disaster)
            goto error;
   }
…

error:
   clean up the mess
```

```
   for (i = 0; i < n; i++)
      for (j = 0; j < m; j++)
         if (a[i] == b[j])
            goto found;
   /* nije pronađen nijedan zajednički element */
   …
found:
               /* pronađen jedan: a[i] == b[j] */
   …
```

```
found = 0;
for (i = 0; i < n && !found; i++)
   for (j = 0; j < m && !found; j++)
      if (a[i] == b[j])
         found = 1;
if (found)
   /* pronađen jedan: a[i-1] == b[j-1] */
   …
else
   /* nije pronađen nijedan zajednički element */
   …
```