

# DST 1

Nicholas Wickström

IDE, Högskolan i Halmstad

2009

# Översikt

- 1 Satsler
  - if-(then)-else
  - switch
  - while
  - do - while
  - for
- 2 Operatorer
- 3 Vektorer, matriser
- 4 Preprocessor

# Outline

- 1 Satsar
  - if-(then)-else
  - switch
  - while
  - do - while
  - for
- 2 Operatorer
- 3 Vektorer, matriser
- 4 Preprocessor

## Satsar - (if, if-(then)-else)

```
/* if sats */  
if (n > 0) {  
    fAvg = fSum / (float)n;  
}
```

```
/* if-then-else */  
if (n >= 0)  
    nPositive = TRUE;  
else  
    nPositive = FALSE;
```

## Satsar - (if, if-(then)-else, forts.)

```
/* Nästlad if-then-else sats, flervals */  
  
if (nColor == YELLOW)  
    nYellow++;  
else if (nColor == BLUE)  
    nBlue++;  
else if (nColor == GREEN)  
    nGreen++;  
else  
    nOthers+;
```

## Satsler - (switch)

```
switch (nColor) { /* switch-sats, flervals */
  case YELLOW:
    nYellow++;
    break;
  case BLUE:
    nBlue++;
    break;
  ...
  default:
    nOthers++;
    break;
}
```

# Satsler - (while)

```
/* Beräkna n-fakulteten (n!) */  
nfak = i = 1;  
while (++i <= n)  
    nfak *= i;
```

## Satsler - (do)

```
do {  
  if (x-y < 0)  
    nError += nSmallAdjust;  
  else  
    nError -= nSmallAdjust;  
} while (nError > MAX_ERR ||  
        nError < -MAX_ERR)
```



## Sats - (for)

```
for (expr1; expr2; expr3)  
    sats
```

ekivalent med

```
expr1;  
while (expr2) {  
    sats  
    expr3;  
}
```

## Sats - (for forts.)

```
/* x raised to n */  
for (i = 1; i <= n; i++)  
    nResult *= x;
```

```
/* Beräkna n-fakulteten (n!) (for-sats) */  
for (i = 1; i <= n; i++)  
    nfak *= i;
```

# Outline

- 1 Satsar
  - if-(then)-else
  - switch
  - while
  - do - while
  - for
- 2 Operatorer
- 3 Vektorer, matriser
- 4 Preprocessor

## Operatorer (ökning/minskning)

```
nAntal++; /* Ekv. nAntal = nAntal + 1; */  
++nAntal; /* Ekv. nAntal = nAntal + 1; */
```

```
i = 4; j = 7;  
a = ++i * --j; /* a => 30 */  
i = 4; j = 7;  
a = i++ * j--; /* a => 28 */
```

# Operatorer (Jämförelse)

< > <= >= == !=

$x < y$      $a > 3.5$      $i + j <= 4$

$x == y$      $i != 0$

## Operatorer (Logiska)

&&    ||    !

OCH, ELLER, ICKE

```
x > 0 || !(i == 2 && j < 4)
```

```
Obs!! i == !!i;  
/* Behöver inte vara sant! */
```

# Operatorer (Bit)

~ << >> & ^ |

BNEG, VSHFT, HSHFT, BOCH, BXOR, BOR

```
c = 0x8F; /* 10001111 */
```

```
c2 = ~c /* 01110000, eller 0x70 */
```

```
c3 = c >> 2 /* 00100011 */
```

```
k = k | 0x20; /* Sätt 6e biten */
```

```
k = k & 0xFFDF; /* Nolla 6e biten */
```

## Operatorer (Tilddelning)

```
i = 0; /* Enkel vanlig tilddelning */
```

```
j = i; /* Alt. i = j = 0; */
```

```
i = i + j; /* Alt. i += j; */
```



# Outline

- 1 Satsler
  - if-(then)-else
  - switch
  - while
  - do - while
  - for
- 2 Operatorer
- 3 Vektorer, matriser
- 4 Preprocessor

# Vektorer, matriser

```
int nVector[100];
int nMatrix[3][3] =
{ { 1, 2, 3}, {4, 5, 6}, {7, 8, 9} };
int nTranspose[3][3];
/* Medelvärde av vektorn */
for (i = 0; i < 100; i++)
    nSum += nVector[i]/100;

/* Transponerat av matrisen nMatrix */
for (i = 0; i < 3; i++)
    for (j = 0; j < 3; j++)
        nTranspose[i][j]=nMatrix[j][i];
```

# Funktioner

```
float calc_avg (float x, float y)
{
    return ((x + y) / 2.0);
}
```

```
/* Anropas med */
nAvg = calc_avg(a, b);
/* Begränsning! Kan bara returnera en variabel */
/* Kan lösas med poster */
/* Men egentligen behöver vi pekare! */
```

# Outline

- 1 Satsler
  - if-(then)-else
  - switch
  - while
  - do - while
  - for
- 2 Operatorer
- 3 Vektorer, matriser
- 4 Preprocessor

# Preprocessor

```
#define N 1000
/* Makro */

#define TRUE 1
#define FALSE !TRUE

#ifdef DEBUG
/* print debug info */
#endif
```

## Preprocessor forts.

```
#ifndef _MYHEADER_H_
#define _MYHEADER_H_

/* include file here */

#endif

/* Preprocessororn byter dessa innan kompileringssteget */
```