

Programming DD1337

ITERATION: FOR, WHILE, FOREACH OCH ITERATOR

MARCUS DICANDER, ÖVNINGSASSISTENT FÖR GRUPP 1

Kartan över arrayer

ArrayList – Dynamisk storlek, add och remove



10 element indexerade från 0-9.

Array – Fix storlek. Add och remove besvärliga.

```
int[] primeNumbers = new int[10];
```

Dagens kompass

Försök skriva kod som är:

Förutsägbar

Lätt att ändra

Lättläst

Anatomi för for

```
for( int i = 0; i < upperBound; i++ ) {  
    System.out.println(i);  
}
```

Anatomi:

```
for ( [initialisering]; [villkor]; [uppdatering] ) {  
    // kod här;  
}
```

for ([initialisering]; [villkor]; [uppdatering]) {

En kommaseparerad lista med initialiseringar som körs en gång i loopens begynnelse innan villkoret testas första gången

Exempel:

int i = 0

alreadyExistingVariable = 0

Initialiseringen kan vara tom.

int i = 0, counter = 0

```
for ( [initialisering]; [villkor]; [uppdatering] ) {
```

Ett uttryck som testas efter initialiseringen samt mellan och efter varje varv i loopen.

Exempel:

`i < numElements`

`i <= maxValue`

`isSafe && turnCount < 1000`

`0 <= i && i < numElements`

```
for ( [initialisering]; [villkor]; [uppdatering] ) {
```

Körs näst sist i varje varv i loopen. Därefter testas villkoret igen.

Exempel:

`++i`

`i++`

`i++, j++`

`current = it.next()`

```
while([villkor]) = for(;;[villkor];)
```

while är ett specialfall av for-loopen.

Speciellt användbar då man inte vet när man ska sluta.

foreach

`for(Membership current:members)`

Bra då man ska gå igenom alla värden och inte behöver index.

Skapar otillgänglig loopvariabel (för arrayer) eller iterator (för Collections)

Synaktiskt socker

Iterator

Klass för att stega igenom Collections. Finns för alla collections.

Skapas genom factory – din Collection gör new åt dig.

Implementerar hasNext(), next() och kanske remove()