

2IP65

Lecture 2

Input/Output
Repetition
Arrays

Let op

- zalen voor donderdag gewijzigd
- zalen voor Basiswiskunde gewijzigd
- schrijf je in bij Studyweb (voornl. om mailtjes te krijgen)
- Wiki `huizing.name/javacourse` heeft RSS
- stuur vragen naar contact2ip65@gmail.com
 - of probeer Twitter :-)
- gebruik geen packages

Three things

Syntax – Conventions – Semantics

1. Syntax: if (...)
2. Conventions: indent after each {
3. Semantics: when guard is true, execute first part

Output

- ④ console output:
 - ④ `System.out.println("x is "+x);`
 - ④ without new line: `System.out.print("x is "+x);`
 - ④ numeric arguments are automatically converted to their textual decimal representation
- ④ dialog output: `JOptionPane.showMessageDialog(null, "boe!");`

Input

- console input: `scanner.next(); scanner.nextInt(); scanner.nextDouble();`

every next-command reads an input word and advances to the next word

- `import java.util.Scanner;`

- Declare variable: `Scanner scanner;` (or any other name)

- Create and connect object: `scanner = new Scanner(System.in);`

- `scanner.hasNext()` is true when there is input left (always true in interactive situation, not always true when reading from file (such as in Peach))

- `scanner.hasNextInt(), hasNextDouble()` etc. are true when such a number is on top of input

- dialog input: `JOptionPane.showInputDialog(null, "type wat");`



A fast computer can perform one instruction every nanosecond (1 GHz). Suppose an incredibly fast programmer writes one instruction every second.

It will take him 30 years to program one second of computer time...

Solution? ... Repetition!

Repetition (while)

- Aim: execute some statements more than once
- Means: while-construct
- Meaning: repeat loop-body while guard is true; inspection before each execution
- (this is in fact the third trick)

Repetition example

```
int x = 0;
while (x<3) {
    System.out.println(x);
    x = x+1;
}
System.out.println(
    "ready!");
```

=

```
int x = 0;

if (x<3) continue else stop
    System.out.println(x);    0
    x = x+1;

if (x<3) continue else stop
    System.out.println(x);    1
    x = x+1;

if (x<3) continue else stop
    System.out.println(x);    2
    x = x+1;

if (x<3) continue else stop

System.out.println(
    "ready!");                ready!
```

output

Repetition - syntax

```
while ( <boolean expression> ) {           "guard"  
    <statements>                           "loop body"  
}
```

- boolean expression as in if: something with the value true or false

- examples:

`x < 2`

`x == y`

`x > 0 && x < 10`

Boolean Operators

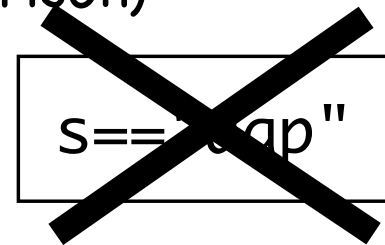
operator	description	example expression	result
<	less than	2 < 3	true
>	greater than	2 > 2	false
<=	less than or equal to	2 <= 2	true
>=	greater than or equal to	3 >= 5	false
==	equal to	1 == 1	true
!=	not equal to	1 != 1	false
&&	logical <u>and</u>	0 < 2 && 2 < 3 0 < 2 && 3 < 2	true false
	logical <u>or</u>	0 < 2 3 < 2 2 < 0 3 < 2	true false
!	negation, logical <u>not</u>	! true !(2 < 0)	false true

do not mix up = (assignment) and == (comparison)

do not use == to compare Strings

use instead:

s.equals("aap") or "aap".equals(s)



for-loop

- when you know the number of iterations (number of times you execute the loop-body) on beforehand
- when you want to do something for each member of a set (range) of values

```
for (int i = 0; i<100; i=i+1) {  
    System.out.println(  
        "Ik mag de accolades om loop-body's niet vergeten"  
    );  
}
```

for-loop - syntax

```
for ( <initialisation>; <guard>; <increment> ) {  
    <statements>  
}
```

- <initialisation>: starting assignment to counter
- <increment>: statement that "increases" counter (i=i+1, j=j-1, k=k+x, etc.)
- <guard>: boolean expression as in while

do-while loop

- when you have to do something at least once

```
double balance;  
String choice;  
// init balance  
do {  
    System.out.println("Type amount for deposit.");  
    deposit = sc.nextDouble();  
    balance = balance + deposit;  
    System.out.println("New balance is " + balance + ".");  
  
    System.out.println(  
        "Do you want to make a deposit? Type y/n");  
    choice = sc.next();  
} while (choice.equals("y"));
```

do-while loop

- when you have to do something at least once

```
int som = 0;
int cijfer = 0;

do {
    som = som + cijfer;
    System.out.println("type cijfer, -1 om te stoppen");
    cijfer = scanner.nextInt();
} while (cijfer > 0);
```

Remarks

- `i++` and `i+=1` are shorthand for `i=i+1`

idea of `+=` works with many operators, such as `*=`, `-=`, and `%=`

- don't mix them! `i=i++` is wrong!

- It is in some cases allowed to write the body of a loop or the branch of an if without braces (`{ }`).

Don't do this. It easily leads to errors.