

# Tentamenshjälpmedel. Språkkonstruktioner i Java

## Typer

### Primitiva typer (variabler lagrar värdet)

prim. typ	motsv. klass	ex. literaler	default
boolean	Boolean	true, false	false
char	Character	'A', '3', '\n'	'\u0000'
int	Integer	37, -3, 12345	0
double	Double	3.1416, 1E-10	0.0

### Referenstyper (variabler lagrar en referens)

- Fältyper (arrays)  
Exempel: `int[], Ball[], double[][]`  
Objekt måste skapas: `int[] a = new int[10];`  
Initiering: `double[][] data = {{1,3},{4,8}};`  
Indicering: `a[i], 0 ≤ i < a.length`.
- Klasstyper.  
Klassnamn börjar med versal.  
Objekt måste skapas med *konstruktor*:  
`Ball b = new Ball(10,20,Color.RED);`  
`LifeModel model = new MyLifeModel(50,50);`
- Interfacetyper.  
Deklarerar metoder med resultattyp, namn, parametrar, undantag.  
Objekt kan *inte* skapas, men  
klasser kan *implementera* ett interface.

## Variabler

Variabler måste deklarerars: `int x; double[] ys;`

### Initiering

Instansvariabler och fältelement initieras till

- värdet *default* ovan för primitiva typer;
- `null` för referenstyper.

Lokala variabler *måste* initieras med tilldelning.

## Uttryck

Uttryck byggs av variabler, literaler, operatorer och funktioner/metodanrop.

### Binära operatorer (i precedensordning)

operator	argtyp	restyp	kommentar
<code>*</code> , <code>/</code> , <code>%</code>	number	number	Konkatenering
<code>+</code> , <code>-</code>	number	number	
<code>+</code>	String	String	
<code>&lt;</code> , <code>&lt;=</code> , <code>&gt;</code> , <code>&gt;=</code>	number	boolean	
<code>==</code> , <code>!=</code>	any	boolean	
<code>&amp;&amp;</code>	boolean	boolean	
<code>  </code>	boolean	boolean	

*number* betyder numerisk primitiv typ.

*any* betyder vilken typ som helst.

### Unära operatorer (i precedensordning)

operator	argtyp	restyp	kommentar
<code>!</code>	boolean	boolean	Negation
<code>+</code> , <code>-</code>	number	number	

### Strängkonkatenering

Om *a* har typ `String` betyder *a* + *b* konkatenering;  
*b* konverteras först till `String`.

### Typkonvertering

Konvertering av ett uttryck av typ `int` till `double` sätts in automatiskt när sammanhanget kräver det.

I omvänd riktning krävs explicit konvertering: `(int)e`

### Metodanrop; parameteröverföring

Aktuella arguments värden vid anropsplatsen beräknas; parametrarna initieras till dessa värden innan metodkroppen exekveras.

För referenstyper betyder det att *referensen* kopieras.

En funktions resultat ges i en `return`-sats; värdet kopieras till anropsplatsen.

## Satser

<code>l-expr = r-expr;</code>	<code>method-name(e<sub>1</sub>,...e<sub>n</sub>);</code>
<code>var++;</code>	<code>var--;</code>
<code>break;</code>	<code>continue;</code>
<code>return;</code>	<code>return expr;</code>
<code>if (test) {     statements }</code>	<code>if (test) {     statements } else {     statements }</code>
<code>while (test) {     statements }</code>	<code>do {     statements } while (test);</code>
<code>for (init; test; upd) {     statements }</code>	<code>for (type var : expr) {     statements }</code>
<code>switch (expr) {     case lit<sub>1</sub>: stmt     ...     case lit<sub>n</sub>: stmt }</code>	<code>try {     statements } catch (exc-type var) {     statements }</code>

## Klasser

- Funktionsbibliotek (Ex: `Math`, `Arrays`).  
Innehåller *bara* statiska funktioner/subrutiner.
- Mallar från vilka objekt skapas (de flesta klasser).  
Innehåller oftast *inte* statiska metoder.  
I stället instansvariabler, konstruerare, metoder.
- Huvudklassen i en applikation. Innehåller  
`public static void main(String[] args)`

## Överlastning

Flera metoder får ha samma namn om parameterlistorna skiljer sig åt i antal och/eller typer.

## Utdrag ur Java 1.6 API samt ur kursbokens klasser

Konvention: *X* står överallt för en godtycklig primitiv typ.

### Funktionsbibliotek

Alla metoder i denna kolumn är `static`.

#### `java.lang.Math`

Bland andra: `abs`, `max`, `min`, `sin`, `cos`, `exp`, `log`, `pow`, `sqrt`, `round`, `floor`, `random`.

#### `java.util.Arrays`

```
int binarySearch(X[] a, X key)
X[] copyOf(X[] orig, int len)
boolean equals(X[] a1, X[] a2)
void fill(X[] a, X val)
void sort(X[] a)
String toString(X[] a)
```

#### `java.lang.Character`

```
int digit(char ch, int radix)
char forDigit(int dig, int radix)
boolean isDigit(char ch)
boolean isLetter(char c)
char toLowerCase(char ch)
char toUpperCase(char ch)
```

#### `java.lang.Double`

```
double parseDouble(String str)
```

#### `java.lang.Integer`

```
int parseInt(String str)
```

#### `java.lang.System`

```
InputStream in
PrintStream out
PrintStream err
void exit(int status)
long currentTimeMillis()
```

### Object och String

#### `java.lang.Object`

```
boolean equals(Object obj)
String toString()
```

#### `java.lang.String`

```
implements Comparable<String>

char charAt(int index)
byte[] getBytes()
int indexOf(String str)
int indexOf(String str, int from)
int length()
String substring(int from, int to)
String toLowerCase()
String toUpperCase()
```

### AWT

#### `java.awt.Color`

```
Color(int r, int g, int b)
Color brighter()
Color darker()
int getRed()
int getGreen()
int getBlue()
boolean equals(Color c)
String toString()
```

### Bokens klasser

#### `StdIn` (Alla metoder är `static`)

```
boolean isEmpty()
int readInt()
double readDouble()
long readLong()
boolean readBoolean()
char readChar()
String readString()
String readLine()
String readAll()
```

#### `StdDraw` (Alla metoder är `static`)

```
void setXscale(double min, double max)
void setYscale(double min, double max)

void clear()
void clear(Color c)
double getPenRadius()
void setPenRadius(double r)
Color getPenColor()
void setPenColor(Color c)
Font getFont()
void setFont(Font f)
void line(double x0, double y0, double x1, double y1)
void point(double x, double y)
void circle(double x, double y, double r)
void filledcircle(double x, double y, double r)
void square(double x, double y, double r)
void filledsquare(double x, double y, double r)
void rectangle(double x, double y, double halfWidth, double halfHeight)
void filledRectangle(double x, double y, double halfWidth, double halfHeight)
void text(double x, double y, String s)
void text(double x, double y, String s, double degrees)
```

```
void picture(double x, double y, String s)
void show(int t)
void show()
void save(String filename)
```

### Picture

```
Picture(int w, int h)
Picture(String filename)
void show()
int height()
int width()
Color get(int i, int j)
void set(int i, int j, Color c)
void save(String name)
void setOriginUpperLeft()
void setOriginLowerLeft()
```

### In

```
In() (standard input)
In(String s) (s kan vara filnamn eller en URL)
char readChar()
int readInt()
double readDouble()
String readString()
String readLine()
String readAll()
boolean hasNextLine()
boolean isEmpty()
```