CURS JAVA SE

@IBM - Eugen Barbu

Identifiers in Java

Identifiers are:

- Text strings that represent variables, methods, classes, or objects
- Case-sensitive
- Characters can contain digit, letter, dollar sign ($) or underscore (_)

Identifiers cannot:

- Begin with digit
- Be the same as a reserved word

✓

```
An_Identifier
a_2nd_Identifier
Go2
$10
```

✗

```
An-Identifier
2nd_Identifier
goto
10$
```

Java Naming Conventions

Java is case-sensitive

- yourname, yourName, Yourname, YourName are four different identifiers

Conventions:

- **Package**: all lowercase
    - theexample
- **Class**: initial uppercase, composite words with uppercase
    - TheExample
- **Method** or field: initial lower, composite words with uppercase
    - theExample
- **Constants**: all uppercase
    - THE_EXAMPLE
- **Variables**: initial lowercase, composite words with uppercase
    - theExample

Reserved words

- Literals

`null`	`true`	`false`

- Keywords

`abstract`	`default`	`if`	`private`	`this`
`assert`	`do`	`implements`	`protected`	`throw`
`boolean`	`double`	`import`	`public`	`throws`
`break`	`else`	`instanceof`	`return`	`transient`
`byte`	`enum`	`int`	`short`	`try`
`case`	`extends`	`interface`	`static`	`void`
`catch`	`final`	`long`	`strictfp`	`volatile`
`char`	`finally`	`native`	`super`	`while`
`class`	`float`	`new`	`switch`	
`continue`	`for`	`package`	`synchronized`	

- Reserved for future use

`byvalue`	`future`	`inner`	`outer`	`var`
`cast`	`generic`	`operator`	`rest`	`volatile`
`const`	`goto`			

Java Data Types

Data type	Description
boolean	A binary value of either true or false
byte	8 bit signed value, values from -128 to 127
short	16 bit signed value, values from -32.768 to 32.767
int	32 bit signed value, values from -2.147.483.648 to 2.147.483.647
long	64 bit signed value, values from -9.223.372.036.854.775.808 to 9.223.372.036.854.775.808
float	32 bit floating point value
double	64 bit floating point value
char	16 bit Unicode character
String	N byte Unicode string of textual data. Immutable

Exemplu: Declararea variabilelor si alocarea valorilor.

Casting Java Primitive Types

- Java is a strictly typed language. Assigning the wrong type of value to a variable could result in a compile error or a JVM exception
- Casting a value allows it to be treated as another type
- The JVM can implicitly promote from a narrower type to a wider type
- To change to a narrower type, you must cast explicitly

`int a, b;`
`short c;`
`a = b + c;`
	`int d;`
	`short e;`
	`e = (short)d;`
		`double f;`
		`long g;`
		`f = g;`
		`g = f;    //error casting`

- A cast is done by putting the name of the type that you want (the result type) in parentheses before the value to be converted. An example of casting an int literal value to a byte:

```
int i = 25;
byte b = (byte) i;
```

- An example of casting a double literal value to an int:

```
int i = (int) 25.123; // The resulting value of i is 25
```

Curs 2

Programming with Java operators (1)

Curs Java SE @ IBM

Java Operators

Operator	Description	Operator Type
++,--	Postfix increment, postfix decrement	Arithmetic
++,--	Prefix increment, prefix decrement	Arithmetic
!	Boolean NOT	Logical
***,/,%**	Multiplication, division, remainder (modulus)	Arithmetic
+,-	Addition, subtraction	Arithmetic
<, <=, >, >=	Less than, less than or equal to, greater than, greater than or equal to	Relational
&&, ||	Conditional AND, Conditional OR	Logical
==, !=	Value equality and inequality	Relational

Programming with Java operators (2)

Addition

```
int sum1 = 10 + 20;          // adding two constant values
int sum2 = sum1 + 33;        // adding a variable and a constant
int sum3 = sum1 + sum2;  // adding two variables

int result = 10;             //variable equals to its own value plus another value
result = result + 20;

int sum4 = 25 + 40 +37;
```

Subtraction

```
int diff1 = 200 - 10;
int diff2 = diff1 - 5;
int diff3 = diff1 - diff2;

int result = 10;
result = result - 5;

int diff = 200 - 10 - 20;

int diff = 200 - (-10);
```

Multiplication

```
int prod1 = 10 * 20;
int prod2 = prod1 * 5;
int prod3 = prod1 * prod2;

int prod = 10 * 20 * 30;
int result = 10;
result = result * 20;
```

Division

```
int division1 = 100 / 10;
int division2 = division1 / 2;
int division3 = division1 / division2;

int division = 100 / 10 / 2;

int result = 100;
result = result / 5;
```

Remainder / Modulo

```
int value     = 100;
int remainder = value % 9;
```

Programming with Java operators (3)

Exercitiu:

Cum aflu daca un numar este par sau impar?

Exercitiu:

Sunt cele doua variabile result1 si result2 de tip int egale?

```
int result1 = 100 * 100 / 5 + 200 * 3 / 2;
 int result1 = 100 * 100 / (5 + 200) * 3 / 2;
```

Basic Math Functions

Math.abs()	**Math.ceil()**	**Math.floor()**

int abs1 = Math.abs(10);
// abs1 = 10
int abs2 = Math.abs(-20);
// abs2 = 20
	double ceil = Math.ceil(7.343);
	//ceil = 8.0
		double floor = Math.floor(7.343);
		// floor = 7.0

Math.min()	**Math.max()**	**Math.round()**

int min = Math.min(10, 20);
//min = 10
	int max = Math.max(10, 20);
	//max = 20
		double roundedDown = Math.round(23.445);
		double roundedUp = Math.round(23.545);

Math.random()	**Math.pow()**	**Math.sqrt()**
double random = Math.random();	double pow8 = Math.pow(2,8); //2 la puterea 8	double sqrtvar = Math.sqrt(9); //sqrtvar = 3
Other Math Functions		
Math.floorDiv(), Math.exp(), Math.log(), Math.log10(), Math.PI, Math.sin(), Math.cos(), Math.tan(), Math.asin(), Math.acos(), Math.atan(), Math.atan2(), Math.sinh(), Math.cosh(), Math.tanh(), Math.toDegrees(), Math.toRadians()		

Input from Scanner

Prin intermediul clasei “Scanner”, se poate citi input-ul din tastatura. Se apeleaza:

```
Scanner sc = new Scanner(System.in);
```

Variabila x va stoca valoarea de tip Integer citita de Scanner:

```
int x = sc.nextInt();
```

Valabil si pentru variabile de tip:

```
String word = sc.next();
float f = sc.nextFloat();
double num = sc.nextDouble();
```

TIP: Folositi functia autocomplete!

Exercises

Exercitiu:
Sa se calculeze aria unui triunghi. Folositi clasa Scanner pentru a putea introduce lungimile laturilor.

Exercitiu:
Sa se calculeze aria unui trapez. Folositi clasa Scanner pentru a putea introduce lungimile laturilor.

Exercitiu:
Sa se calculeze ipotenuza unui triunghi dreptunghic prin metoda lui Pitagora. Folositi clasa Scanner pentru a putea introduce lungimile celor doua catete.

Exercitiu:
Sa se scrie un program care sa citeasca 4 valori din tastatura. Calculati valoare inmultirii dintre maximul primelor doua numere si minumul ultimelor doua numere

Exercitiu:

Experimentati cu 5 functii la alegere definite in clasa Math.