


1

Programmazione 2 - Marco Ronchetti



Why Java? A safer language

A *clean* object-oriented programming language

No pointer arithmetic


Automatic Memory Management
(Garbage Collection)

Automatic array and string bounds check

Fac.Sienze - Università di Trento

2

Programmazione 2 - Marco Ronchetti



"the first universal software platform"

Consists of:

The language *Easy!*

The Virtual Machine *You don't care!*

(Many) class libraries and API *That's the difficult part!*

Java: the platform for **"Internet Computing"**

Hardware independent • Scalable • Open

Fac.Sienze - Università di Trento

3

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Facoltà di Scienze - Università di Trento

Facilità

Java è basato sul C, come il C++.

- Java **TOGLIE** al C alcune caratteristiche difficili e pericolose (**puntatori**).
- Java **AGGIUNGE** al C le caratteristiche di un linguaggio object-oriented (**classi, ereditarietà, messaggi**).
- Java **INTRODUCE** una gerarchia di classi predefinite:
AWT, IO, Lang(tipi, Math, Thread), Exceptions, Net, Utils(Vector, Dictionary, Date...)

4

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Facoltà di Scienze - Università di Trento

Robustezza

La maggior parte degli errori sono legati alla gestione della memoria tramite i **PUNTATORI**:


- puntatori che puntano a locazioni illecite (non allocate)
- puntatori che puntano a locazioni lecite ma sbagliate
 - indirizzi di vettori sbagliati
- memoria allocata e non più rilasciata (memory leaks)

Soluzione di Java:

- **ABOLIZIONE DEI PUNTATORI**
- **GARBAGE COLLECTION**

5

Programmazione 2 - Marco Ronchetti




Java ha:

- eccezioni (Ada, C++)
- Garbage Collection (LISP, Smalltalk)
- libreria di classi (da Smalltalk, Objective-C)

Le specifiche del linguaggio e della Java Virtual Machine sono PUBBLICHE

6

Programmazione 2 - Marco Ronchetti



Storia di Java

- Inizio anni 90: Java nasce come “Oak”
target: **intelligent consumer electronics**.
- Successivamente, nuovo target: **set top box**
- 1994: linguaggio per la “Web” (client side)
- 1996: la prospettiva é “network computing”

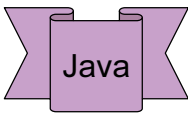

Oggi:

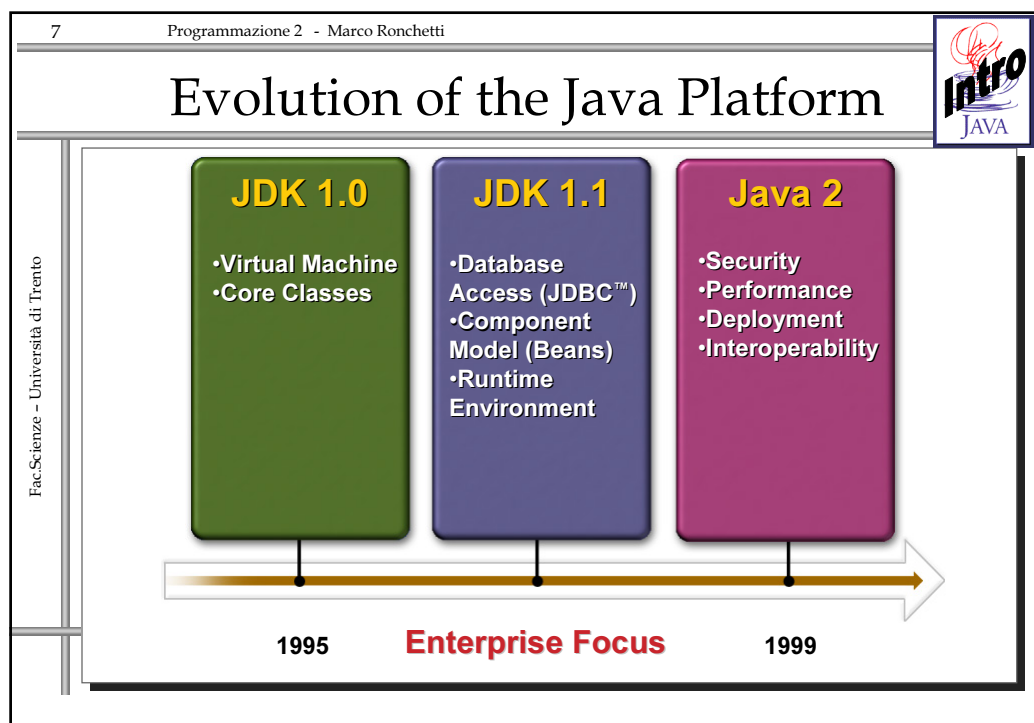
Successi

- Device-independent GUI
- Web on the server side (Servlets, JSP, EJB, XML...)

Prospettive

intelligent consumer electronics + smartcards





8 Programmazione 2 - Marco Ronchetti

“The” Tutorials and examples

<http://java.sun.com/docs/books/tutorial/?frontpage-spotlight>

The Java™ Tutorial

A practical guide for programmers

with hundreds of complete, working examples

The Tutorial is organized into *trails*--groups of lessons on a particular subject.

Trails Covering the Basics:

Published in: **The Java Tutorial Second Edition**

[Your First Cup of Java](#): Detailed instructions to help you run your first program. *for Win32, for UNIX, for Mac*

[Getting Started](#) [Essential Java Classes](#)

[Learning the Java Language](#) [Custom Networking](#)

[Writing Applets](#) [JDK™ 1.1 -- And Beyond](#)

Trail on Constructing GUIs:

Published in: **The JFC Swing Tutorial**

[Creating a GUI with JFC/Swing](#)

Tutorial Books:

Fac.Sienze - Università di Trento


9

Programmazione 2 - Marco Ronchetti


Intro
JAVA

Fac.Scienze - Università di Trento

More Tutorials and examples



<http://www.phrantic.com/scoop/onjava.html>



<http://www.mokabyte.it/>

10

Programmazione 2 - Marco Ronchetti

Intro
JAVA

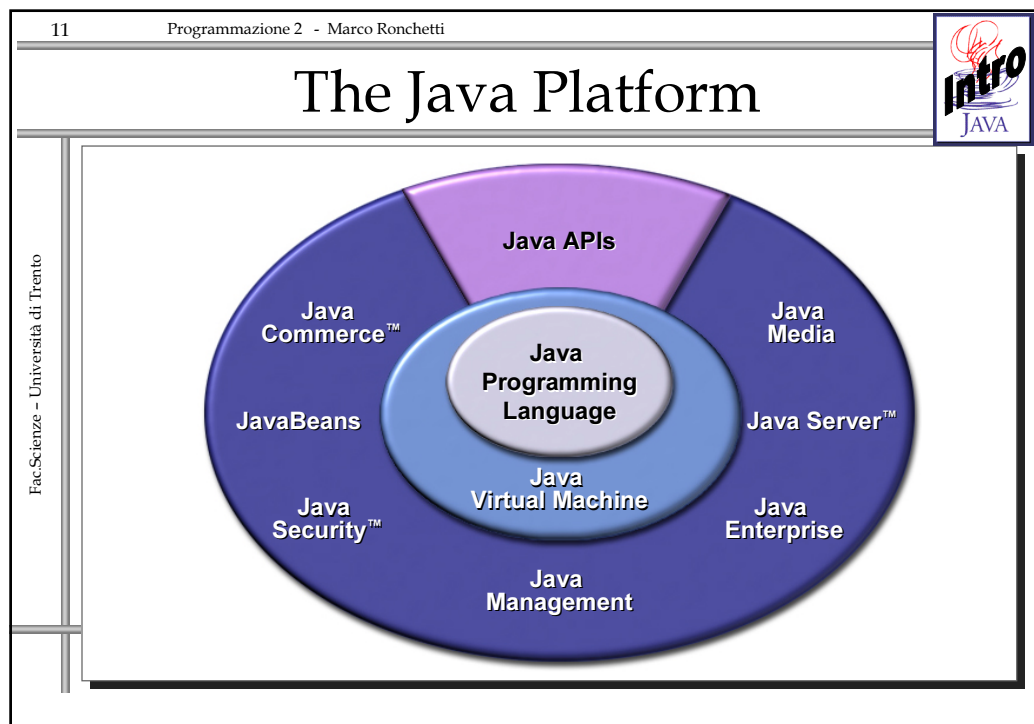
Fac.Scienze - Università di Trento

Un buon libro...

Gratis in forma elettronica:
Thinking in Java
Bruce Eckel

<http://www.mindview.net/Books>

In Italiano:
Thinking in Java
Bruce Eckel
Ed. Apogeo
(in libreria)



12 Programmazione 2 - Marco Ronchetti

Differenze tra Java e C++

A slide titled "Differenze tra Java e C++" (Differences between Java and C++). It features a humorous formula: $?(Java == ((C++)- -)++)$. The slide is part of a presentation by Marco Ronchetti, as indicated by the header. The footer of the slide area contains the text "Fac.Sienze - Università di Trento".

Fac.Sienze - Università di Trento

Intro
JAVA

13

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Forma di un programma

In Java tutto e' una "classe".
Lo scheletro minimo di un programma e':

```
import ...;  
class myProgram {  
    public static void main (String args[]) {  
        ...  
    }  
}
```

import <= Include "intelligente"
(senza bisogno di #ifdef)
NON c'è precompilatore!

Fac.Sienze - Università di Trento

14

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Nomi

I programmi Java includono nomi per identificare alcune entità di programmazione
(**packages, classes, interfaces, methods, variables, statement**)

Nomi validi sono composti da un numero illimitato di lettere e numeri **UNICODE**, iniziare con una lettera.

I nomi non possono essere Java keywords.

Fac.Sienze - Università di Trento

15

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Keywords

Le keywords usate attualmente sono

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

Oltre a queste, alcune keywords sono riservate
per usi futuri:

by value cast const future generic goto inner operator
outer rest var

Fac.Scienze - Università di Trento

16

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Commenti


3 forme di commento:

```
/* C style */  
/* Questo tipo di commento  
può proseguire su pi linee */  
/* NOTA: ATTENZIONE AI /*COMMENTI*/ NIDIFICATI! */  
// C++ style  
// Una intera riga commentata  
a=a+3; // Commento su una linea di codice  
/**documentation */  
/**Stile di commento usato da JAVADOC  
per la generazione automatica di  
documentazione */
```

Fac.Scienze - Università di Trento

17

Programmazione 2 - Marco Ronchetti

Intro
JAVA


Tipi di dato primitivi

FacScienze - Università di Trento

Type	Contains	Default	Size	Min/Max Value
boolean	true or false	false	1 bit	N.A. / N.A.
char	Unicode char	\u0000	16 bits	\u0000 / \uFFFF
Byte	signed integer	0	8 bits	-128 / 127
short	signed integer	0	16 bits	-32768 / 32767
int	signed integer	0	32 bits	-2147483648 / 2147483647
long	signed integer	0	64 bits	-9223372036854775808 / 9223372036854775807
float	IEEE 754 f.p.	0.0	32 bits	+/-3.40282347E+38 / +/-1.40239846E-45
double	IEEE 754 f.p.	0.0	64 bits	+/-1.79769313486231570E+308 / +/-4.94065645841246544E-324

18

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Unicode

Java characters, strings, and identifiers are composed of 16-bit Unicode characters. This makes Java programs relatively easy to internationalize for non-English-speaking users.

Most platforms cannot display all 38,885 currently defined Unicode characters

The Unicode character set is compatible with ASCII and the first 256 characters (0x0000 to 0x00FF) are identical to the ISO8859-1 (Latin-1) characters 0x00 to 0xFF.

Unicode `\u` escape sequences are processed before the other escape characters

FacScienze - Università di Trento

19

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Literals (costanti)

Fac.Sienze - Università di Trento

interi (sempre int, long se serve)
0777 ottale 0xFF esadecimale 77L long

reali
10.4 1.04E01 double 10.4F 1.04E01F float

boolean
true false

carattere
tutte le escape sequences del C sono riconosciute (\n \t \' \'\" \\ ...)
Unicode: \u0022 has exactly the same meaning to the compiler as "

stringhe
“questa e’ una stringa”

20

Programmazione 2 - Marco Ronchetti

Intro
JAVA

String Literals

Fac.Sienze - Università di Trento


Strings in Java are not a primitive type, but are instances of the String class.

However, because they are so commonly used, string literals may appear between quotes in Java programs, just as they do in C:
“pippo”

When the compiler encounters such a string literal, it automatically creates the necessary String object.

21

Programmazione 2 - Marco Ronchetti

Intro
JAVA


Operatori

Gruppo	Funzione	Operatori
Arithmetic	comparazione unitari algebrici postfissi	=, !=, <, <=, >, >= +, - +, -, *, /, % ++, --
Bit	shift bitwise comparison	<<, >>, >>> ~, &, , ^
Boolean	relazionali logici	=, != !, &, , ^, &&,
String	concatenazione	+

Fac.Sienze - Università di Trento

22

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Operatori

Since Java does not allow you to manipulate pointers directly, it does not support the reference and dereference operators *, -, >, and &, nor the sizeof operator.

Java also adds some new operators:

The + operator applied to String values concatenates them. If only one operand of + is a String, the other one is converted to a string.

Java does not support operator overloading--the language designers decided (after much debate) that overloaded operators were a neat idea, but that code that relied on them became hard to read and understand.

FacScienze - Università di Trento

23

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Operatori

The **instanceof** operator returns true if the object *o* on its left-hand side is an instance of the class *C* or implements the interface *I* specified on its right-hand side.

It also returns true if *o* is an instance of a subclass of *C* or is an instance of a subclass of some class that implements *I*.

instanceof returns false if *o* is not an instance of *C* or does not implement *I*.

It also returns false if the value on its left is null.

If **instanceof** returns true, it means that *o* is assignable to variables of type *C* or *I*. The **instanceof** operator has the same precedence as the *<*, *<=*, *>*, and *>=* operators.

Fac.Sienze - Università di Trento

24

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Tipi di dato derivati (reference data)

Java, come tutti i linguaggi OO, permette di definire **NUOVI TIPI DI DATO** (classi).

Alcuni tipi di dato (classi) sono predefinite:
ad esempio le stringhe. (**String**)

tipo

identificatore

Operatore di creazione

costruttore

```
Point punto = new Point(10,10);
```

No Structures or Unions

Java does not support C struct or union types. Note, however, that a class is essentially the same thing as a struct, but with more features. And you can simulate the important features of a union by subclassing.

Fac.Sienze - Università di Trento

25

Programmazione 2 - Marco Ronchetti

Intro
JAVA

"Java non ha i puntatori"

Ma è vero?

```
Point punto = new Point(10,10);
```

l'identificatore di un oggetto ("punto")
sembra proprio un puntatore!

Quel che Java non ha è
l'aritmetica dei puntatori

Fac.Sienze - Università di Trento

26

Programmazione 2 - Marco Ronchetti

Intro
JAVA

Confronto dell'operatore new

```
in C++: Point * punto = new Point(10,10);  
in Java: Point punto = new Point(10,10);
```

punto.x di Java equivale a punto->x del C++

In Java gli oggetti sono accessibili
SOLO per referenza

Fac.Sienze - Università di Trento

27

Programmazione 2 - Marco Ronchetti

Intro
JAVA

memory management

La gestione (dinamica) della memoria e' automatica, tramite

- la creazione (operatore new) e
- la distruzione (garbage collection) di oggetti.

GC interviene quando serve memoria.

GC elimina gli oggetti per i quali non vi sono piu' riferimenti attivi.

GC puo' essere attivato su richiesta esplicita: `System.gc()`

Fac.Sienze - Università di Trento

28

Programmazione 2 - Marco Ronchetti

Intro
JAVA


memory management

Operazioni da eseguirsi alla nascita di un oggetto vanno definite nel metodo "costruttore".

Ogni classe deve avere uno (o piu') costruttori.

Operazioni da associarsi con l'eliminazione di un oggetto possono essere definite nel metodo "distruttore" `finalize()` (opzionale)

Fac.Sienze - Università di Trento

29	Programmazione 2 - Marco Ronchetti	
<h1>Arrays</h1>		
Fac.Scienze - Università di Trento	<p>E' possibile definire arrays di tutti i tipi di dati (elementari o classi). In fase di DEFINIZIONE non e' necessario specificare la dimensione del vettore.</p> <p>Solo al momento della ALLOCAZIONE viene richiesto lo spazio desiderato.</p> <pre>String[] strings; // this variable can refer to any String array strings = new String[10]; // one that contains 10 Strings strings = new String[20]; // or one that contains 20.</pre> <pre>float f[][] = new float[5][3]; //array bidimensionale</pre> <pre>char s[]={'+','-','*','/','=' ,'C'}; // array inizializzato in creazione</pre>	