

Alma Mater Studiorum-Università di Bologna

Scuola di Ingegneria

Dai Linguaggi alle Infrastrutture Software

Corso di Laurea in Ingegneria Informatica

Anno accademico 2025/2026

Prof. ENRICO DENTI

Dipartimento di Informatica – Scienza e Ingegneria (DISI)



“PROGRAMMARE” ... ?

- In passato, “occuparsi di informatica” è stato a lungo sinonimo di “programmare computer”
- Fa pensare a una *attività poco stimolante*, dove le *fasi creative – analisi e progetto* – sono già avvenute
 - tipico della costruzione di *algoritmi* di medio/piccola dimensione, normalmente già noti e descritti in letteratura
- Anni '80: “crisi del software”
insufficienza di metodologie e strumenti nati per una *visione algoritmica* dell’informatica (*programmazione in-the-small*) rispetto a progettazione, sviluppo e manutenzione di *sistemi software complessi* (*progettazione in the large*)



PROGETTARE SISTEMI SOFTWARE

- Il termine “programmare” richiama una *visione obsoleta* dell’informatica
- Nuovo obiettivo: *saper progettare e costruire sistemi software* con il *rigore* e le *garanzie* che si richiedono *a tutti gli altri prodotti dell’ingegneria*
 - qualità
 - affidabilità

Necessità di *uscire dall’artigianato*

Concepire il *software* come un prodotto industriale

Se necessario, usare *anche l’AI* ma *dominandola*



UN CAMBIO DI PARADIGMA

- Uscire dall'artigianato
 - non reinventare la ruota ogni giorno
 - impossibile fare le cose da soli "nel proprio sottoscala"
 - pieno supporto al *lavoro di squadra*
- Software come prodotto industriale
 - strutturazione precisa del *processo di produzione*
 - supporto al *versioning* del software
 - *collaudo sistematico* come parte integrante del processo
 - *automatizzazione* degli step chiave del processo
 - cura degli *aspetti legali*: *privacy by design and by default*
(General Data Protection Regulation – GDPR, 25/5/2018)



GESTIRE IL PROCESSO DI SVILUPPO

- Scrivere il software è solo una parte del problema
- Gestire il *processo di sviluppo* è altrettanto importante, nonché *critico* e costoso
- Un aspetto critico: *il controllo di versione*
 - lo sviluppo del software passa attraverso *molti file*
 - si fanno modifiche, poi si torna indietro, ci si pente
 - più persone lavorano contemporaneamente su più parti
 - pensare di farlo col *copia & incolla* significa *avviarsi al disastro*
 - non si sa più cosa c'è in ogni file
 - non si riescono a mantenere coerenti versioni successive
 - grande spreco di tempo e di soldi
 - occorre affidarsi a un *sistema di versioning*



SISTEMI DI VERSIONING

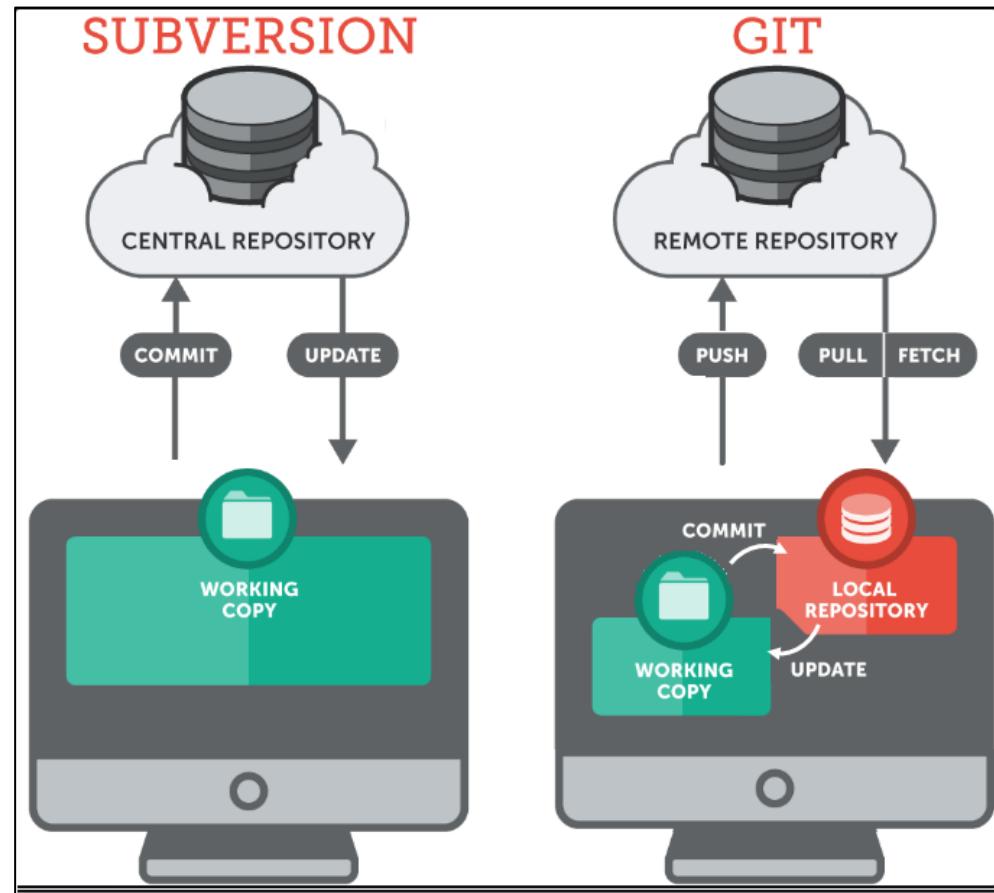
- Un **sistema di versioning** permette di:
 - tenere traccia di ogni versione di **ogni singolo file** del progetto
 - tornare indietro a versioni precedenti, senza mai perdere niente anche in caso di sbagli
 - gestire **diramazioni (branch)** sperimentali di un software, senza "pasticciare" sulla versione principale
 - riunire (**merge**) file di versioni diverse in una versione unica
 - ottenere **report** sulle differenze fra file di versioni diverse
 - gestire le **dipendenze** fra progetti diversi o fra versioni per piattaforme diverse
 - supportare il lavoro di **squadra** (chi ha fatto cosa e quando), anche nel distribuito, anche offline

SISTEMI DI VERSIONING

- I sistemi di versioning più diffusi sono **SVN** e **GIT**

- Idea base

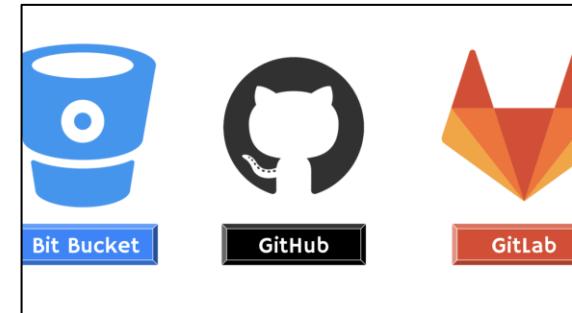
- i file del progetto locale sono copiati, a richiesta, su un **repository** che traccia le versioni
- in **SVN**, il repository è unico e **centralizzato**
- in **GIT**, ogni pc ha il suo repository (**distribuito**), che solo a richiesta è copiato nel **repository centrale remoto**
→ *oggi più diffuso*





SISTEMI DI VERSIONING

- Esistono molti siti che offrono *hosting SVN e/o GIT*
 - per GIT,
i più noti sono GitHub, GitLab, BitBucket
 - per SVN (oggi meno usato),
GitHub, RiouxSVN, Atlassian, etc.



The screenshot shows a detailed view of a GitLab project named "2P-Kt". The left sidebar includes links for Project overview, Details, Activity, Releases, Issues (42), Merge Requests (2), Requirements, CI / CD, Boards, Labels, Service Desk, Milestones, and Iterations. The main area displays a list of commits, branches, and tags. Key commit details include:

- gciatto-unibo fix halt solution rendering in :ide (commit e2c2cc3)
- .github/workflows use s01 server for central upload
- .img update README.md
- core fix js names in Scope
- documentation use de.fayard.dependencies instead of de.fayard.buildSrcVersion
- dsl-core add common directives in PrologStdLibScope
- dsl-solve make standard channels non-removable
- dsl-theory add further theoryOf methods in PrologScopeWithTheories
- dsl-unify format code in :dsl-unify
- examples Merge branch 'develop' into feature/kt-mpp++
- gradle/wrapper use gradle 6.8

On the right, project statistics are shown: 2,420 Commits, 13 Branches, 53 Tags, 2.9 MB Files, 760.6 MB Storage. Below this, a list of files and their last commits is provided, such as .github/workflows, .img, core, documentation, and others.



SISTEMI DI VERSIONING

- Serve poi un *client* sul proprio pc
 - Windows: TortoiseSVN, TortoiseGIT, SourceTree, ...
 - Mac: SourceTree, Versions, ...
 - spesso integrati negli strumenti (es. Eclipse, IntelliJ, ...)



TortoiseSVN the coolest interface to (Sub)version control

Home About Downloads Translations Support/Docs Other tools Support this project

Tortoise SVN

Info

About
About TortoiseSVN

Screenshots
Screenshots of various dialogs

Testimonials
What users say about TortoiseSVN

News Archive
News archive

TortoiseGit Windows Shell Interface to Git

The Power of Git – in a Windows Shell

TortoiseGit provides overlay icons showing the file status, a powerful context menu for Git and much more!

Learn more [about TortoiseGit](#).

[Download](#)

[Download for Mac OS X](#)
Also available for Windows



Sourcetree

Simplicity and power in a beautiful Git GUI

[Download for Mac OS X](#)

A free Git client for Windows and Mac

sourcetree-website (Git)

Commit Pull Push Branch Merge Shelve Show in Finder Terminal Settings

All Branches Show Remote Branches Ancestor Order

File status Graph

WORKSPACE

HEAD

Search

BRANCHES

BOOKMARKS

TAGS

REMOTES

SHELVED

SUBREPOSITORIES

b735bc7 Rahul Chihani [r] master [r] origin/master [r] origin/HEAD [r] Reminding me... Mar 3, 2016, 11:15 AM

d9a79d6 Tyler Tadej [r] orgin/master [r] orgin/HEAD [r] orgin/google-verification [r] Update google verification Feb 11, 2016, 21:10

3cc3290 Tyler Tadej [r] Add githook Feb 11, 2016, 19:10

dc647f9 Tyler Tadej [r] Replace outdated Atlassian logo in footer with base 4x4 logo Feb 11, 2016, 19:10

7e51288 Michael Min [r] Merge in hens_images (pull request #10) Feb 13, 2016, 10:30

24d64ff Joost Unger [r] [r] orgin/hens_images [r] Hens_images Used Timings to fix it Feb 13, 2016, 10:30

ce75063 Michael Min [r] Merge pull request #12 from bugfixes-https (pull request #12) Feb 15, 2016, 10:30

895d577 Joost Unger [r] Replacing hens images with new version of Sourcetree Feb 16, 2016, 2:59

384ed55 Rahul Chihani [r] Merge pull request #13 from bugfixes-https (pull request #13) Feb 16, 2016, 2:59

69a7499 Michael Min [r] Merge branch 'master' into 'master' Feb 16, 2016, 2:59

fe93f5e Michael Min [r] Skip records with empty serverid if no gas rejects them Feb 19, 2016, 1:10

dc0f6f5 Michael Min [r] Corrected column counting Feb 23, 2016, 2:10

051a9c3 Michael Min [r] Merge branch 'master' into 'master' Feb 23, 2016, 2:10

67236c2 Michael Min [r] Merge branch 'master' into 'master' Feb 23, 2016, 1:10

5004992 Michael Min [r] Merge in wdgpush (pull request #11) Feb 23, 2016, 1:10



ESEMPIO: Subversion (SVN)

The screenshot shows the TortoiseSVN interface over a file list. A yellow callout points to the status icons of files like '.svn' and '2p', indicating they are up-to-date with the remote repository. Another yellow callout points to the context menu's 'SVN Update' and 'SVN Commit...' items, which are highlighted with red boxes. A large yellow callout on the right points to the 'Check for modifications', 'Update to revision...', 'Merge...', and 'Add...' menu items, also highlighted with red boxes. Below the interface is a table of local files with their details.

File coerenti con la versione
presente nel repository remoto

Possibilità di aggiornare DAL repository
o di fare il "commit" di file NEL repository

Molti strumenti di ausilio
per gestire le versioni

Nome File	Data Modifica	Tipo	Dimensione
01-OOP intro.pptx	30/01/2013 14:44	Presentazione di ...	5.602 KB
01-OOP xfaq.pptx	24/01/2013 10:43	Presentazione di ...	147 KB
02-Array.pptx	24/01/2013 10:43	Presentazione di ...	203 KB
03-Package e namespace.pptx	24/01/2013 10:43	Presentazione di ...	823 KB



ESEMPIO: Subversion (SVN)

Differenze fra due versioni successive
dello stesso file

The screenshot shows the TortoiseMerge application interface comparing two versions of the Java file `Esempio2.java`.

Left View (Revision 10):

```
1 public class Esempio2 {  
2     public static void main(String[] args) {  
3         Counter c1 = new Counter();  
4         c1.reset(); c1.inc();  
5         System.out.println(c1.getValue());  
6         Counter c2 = c1;  
7         c2.inc();  
8         System.out.println(c1.getValue());  
9         System.out.println(c2.getValue());  
10    }  
11 }
```

Right View (Working Copy):

```
1 public class Esempio2 {  
2     public static void main(String[] args) {  
3         Counter c1 = new Counter();  
4         c1.reset(); c1.inc();  
5         System.out.println(c1.getValue());  
6         Counter c2 = c1;  
7         c2.inc();  
8         System.out.println(c1.getValue());  
9         System.out.println(c2.getValue());  
10    }  
11 }
```

Red circles highlight the title bars of both windows: "Esempio2.java Revision 10" and "Esempio2.java : Working Copy". A yellow speech bubble points from the text above to the right window.

At the bottom of the application window, there is a status bar with the following text:

For Help, press F1. Scroll horizontally with Ctrl-Scrollwheel Left View: ASCII CRLF Right View: ASCII CRLF / +1 Conflicts: 0 CAP NUM SCRL



ESEMPIO: Git

The screenshot shows the SourceTree application interface for managing a Git repository named 'tuprolog'. The main window displays a commit graph and a detailed log of changes. A yellow callout bubble highlights the commit log area with the text: "Modifiche tracciate: chi ha fatto cosa, dove l'ha fatto".

Commit Graph: Shows the relationship between local branches ('master') and remote branches ('origin/master', 'origin/HEAD'). Nodes represent commits, and arrows show merges.

Commit Log:

- Merged TH-COLAMEO into master
- Di Lella tesi
- Fixed bugs
- Added Concordion Test
- no message
- Merged in andrzejolszak/tuprolog/BR-ANDRZEJ (pull request #1)
- Theory completions not added for commented-out and duplicated terms.
- Add support for category tags on completion items. Added support for sorting items on category
- The trailing dot will be added behind the scenes to the user's console input box if the user forges
- Deprecated old editor infrastructure.
- Integrated RSyntaxTextArea and Autocomplete into the IDE.
- Removed contents of build/ and bin/.

Commissione: 1f4394cc4da3bbcbff31f691234c521a2632d4b [1f4394c]
Genitori: d93a4793a6
Autore: tuProlog <roberta.calegari@unibo.it>
Data: mercoledì 21 gennaio 2015 14:03:59
Etichette: HEAD, origin/master, origin/HEAD, master

deleted Fixture dependencies

Diff View: Shows a comparison between two files in the 'classpath' directory. The right pane displays the XML code with changes highlighted.

```
<?xml version="1.0" encoding="UTF-8"?>
<classpath>
    <classpathentry kind="src" path="src"/>
    <classpathentry kind="src" path="test/fit/fixtures">
        <classpathentry excluding="TestURLClassLoader.class" path="org.eclipse.jdt.junit.TestUtil.class"/>
    <classpathentry kind="con" path="org.eclipse.jdt.junit.TestUtil.class"/>
    <classpathentry kind="lib" path="lib/fit.jar"/>

```



PROGETTARE (PER) IL COLLAUDO

- Le garanzie di qualità di un prodotto dell'ingegneria richiedono una **attività di collaudo integrata e sistematica**

IL COLLAUDO

- non si improvvisa: va *accuratamente progettato*
 - non dopo: *prima ancora di costruire il sistema*
- non è un'attività accessoria: **è parte integrante del TUO lavoro**
- come tale, va *strumentato e supportato* in modo completo
- è espresso dal *piano di collaudo*
 - il piano *guida la validazione* del sistema
- non è fare "qualche stampa" (chi le legge? se sono centinaia?)



TIPI DI COLLAUDO

Esistono diversi *approcci*, con fini diversi.

BLACK-BOX TESTING

- collaudo *progettato e svolto "ai morsetti"* di un componente software, *senza sapere com'è fatto dentro*
- scopo: verificare la correttezza del comportamento esterno, indipendentemente dalla realizzazione

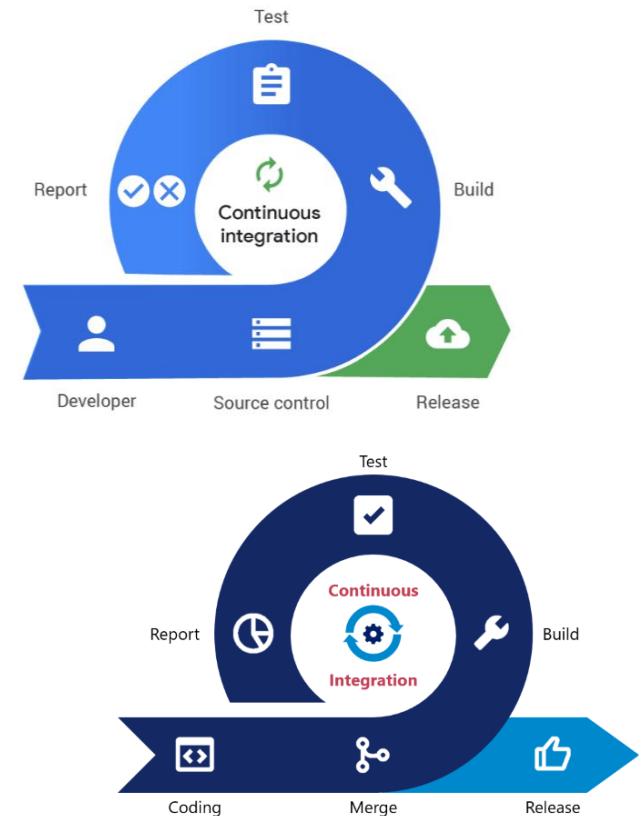
WHITE-BOX TESTING

- collaudo *progettato e svolto tenendo esplicitamente in conto com'è fatto dentro* un componente
- scopo: verificare la corretta implementazione delle varie funzioni, *stressandole una ad una* anche nei casi critici



CONTINUOUS INTEGRATION

- Il collaudo è parte integrante della strategia di sviluppo, oggi molto diffusa, detta *Continuous Integration*
 - il codice è sotto controllo di versione
 - a ogni modifica, si ri-eseguono i test e si ottengono report dettagliati
 - solo se OK, la modifica è approvata; altrimenti, si corregge o si ripristina la versione precedente
 - di tanto in tanto, si rilascia una versione ritenuta consistente (*release*)
- Tutto è sempre sotto controllo





L'IMPORTANZA DEL COLLAUDO SISTEMATICO

- Ad esempio, @Google....
[Communications of the ACM, 2018(4)]

Testing. Nearly all Google code includes corresponding tests, ranging from unit tests all the way to large-scale integration tests. Tests are integrated as a first-class concept in the build system and hermetic and distributed, just like builds. For most projects, developers write and maintain the tests for their code; projects typically have no separate testing or quality-assurance group. Google's continuous build-and-test system runs tests on every commit and notifies a developer if the developer's change broke the build or caused a test to fail. It also supports testing a change before committing to avoid breaking downstream projects.

Criteria for compiler checks. To scale-up our work, we have defined criteria for enabling checks in the compiler, setting the bar high, since breaking the compile would be a significant disruption. A

Gating commits. Some teams want specific analyzers to actually block commits, rather than just appear in the code-review tool. The ability to block commits is commonly requested by teams that have highly specific custom checks with no false positives, usually for a custom DSL or library; and

We advocate for a system focused on pushing workflow integration as early as possible. When possible, checks are enabled as compiler errors. To avoid breaking builds, tool writers take on the task of first fixing all the existing issues in the codebase, allowing us to "ratchet" the quality of Google's codebase one small step at a time, without regressions. Since we present the errors in the compiler, developers encounter them immediately after writing code, while they are still amenable to making changes. To enable this,

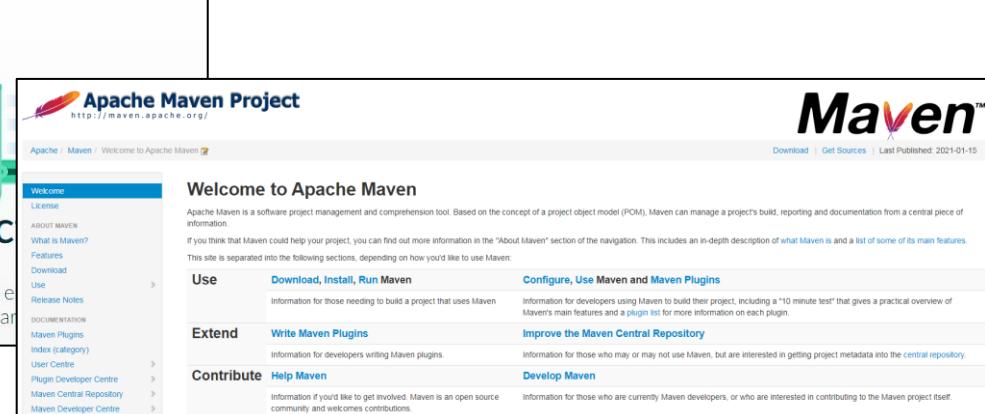


STRUMENTARE IL PROCESSO

- Per questo esistono strumenti che gestiscono / automatizzano il ciclo di sviluppi: i cosiddetti *Build tools*
- Esempi: Gradle, Maven, ...
 - in Gradle si specifica, con un apposito linguaggio, il grafo delle dipendenze fra più progetti + il relativo ciclo di sviluppo



The screenshot shows the official website for Gradle. At the top, there's a navigation bar with links for 'Docs', 'About', 'Training', and 'News'. Below the navigation is a large banner featuring a green and blue industrial-themed illustration of various machinery like a conveyor belt, a mixer, and pipes. The text 'Accelerate developer productivity' is overlaid on the banner. At the bottom of the banner, there's a subtext: 'From mobile apps to microservices, from small startups to big enterprises, Gradle helps teams build, automate and deliver better software.'

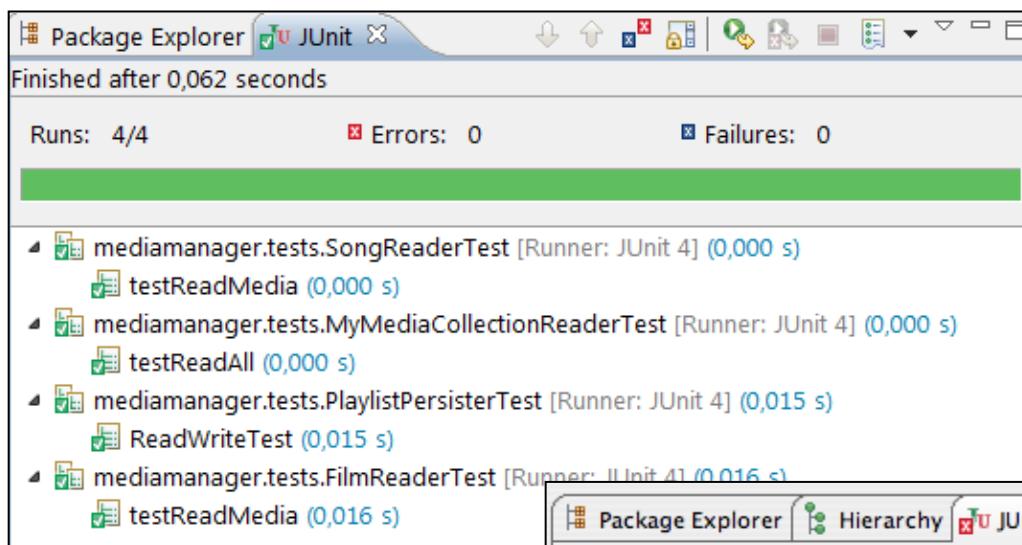


The screenshot shows the official website for Apache Maven. At the top, there's a navigation bar with links for 'Apache', 'Maven', and 'Welcome to Apache Maven'. Below the navigation is a main content area titled 'Welcome to Apache Maven'. It contains several sections: 'Use' (with links to 'Download, Install, Run Maven' and 'Configure, Use Maven and Maven Plugins'), 'Extend' (with a link to 'Write Maven Plugins'), 'Contribute' (with a link to 'Help Maven'), and 'Develop' (with a link to 'Develop Maven'). The right side of the page features the 'Maven™' logo and links for 'Download', 'Get Sources', and 'Last Published: 2021-01-15'.



STRUMENTARE IL COLLAUDO

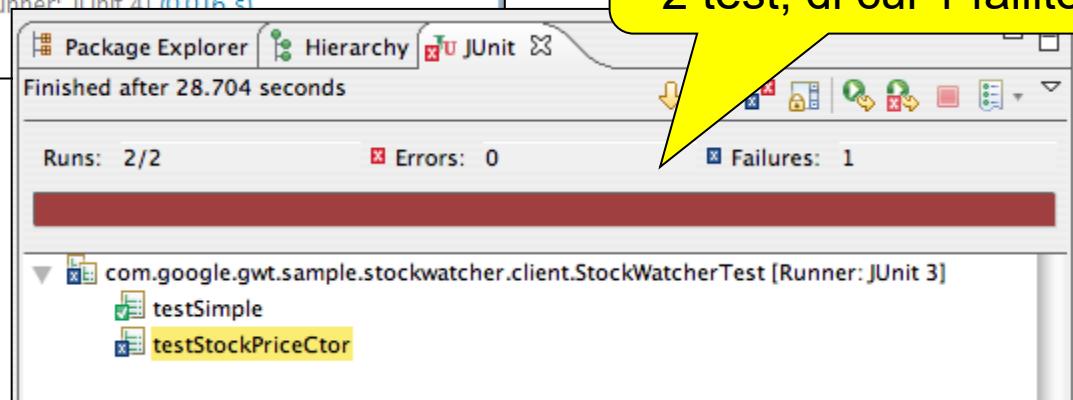
Per Java, lo strumento JUnit esegue in automatico i test progettati, riportandone i risultati in forma grafica



A **failure** is when one of your assertions fails.

An **error** is when some other exception occurs, one you haven't tested for and didn't expect.

Esecuzione di un piano di collaudo:
4 test positivi su 4

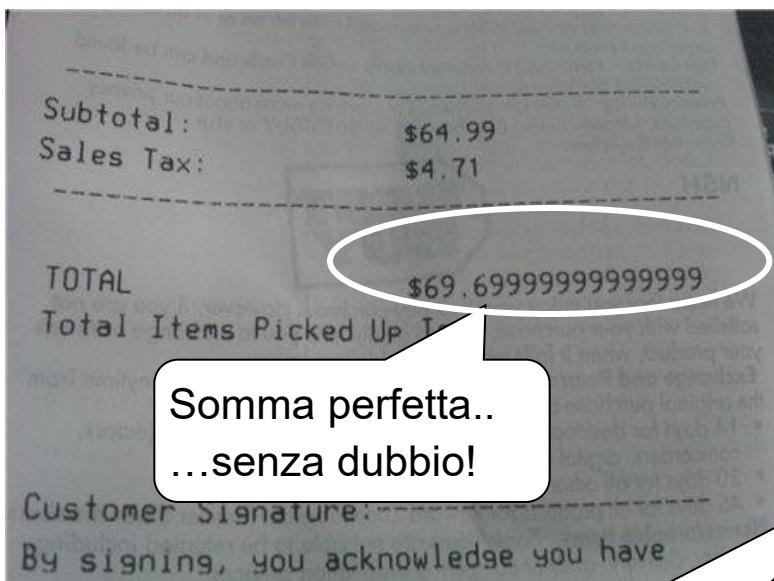


Controesempio:
2 test, di cui 1 fallito



COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si richiedono per automobili, ascensori, case...
si eviterebbero "figuracce" (e danni) come questi



E questo sarebbe il prezzo più basso?? [3/2/2011]

Price	Departure	Arrival	Travel time	Operated by	Flight number
EUR 574	06:45 Bologna 10:20 Amsterdam 16:17 Atlanta	09:05 Amsterdam 14:10 Atlanta 18:06 Miami	17h21		+ Details
EUR 1,760	07:10 Bologna 09:20 Rome	08:10 Rome 14:45 Miami	13h35		+ Details
EUR 1,711	07:15 Bologna 10:50 Paris (CDG)	09:00 Paris (CDG) 14:35 Miami	13h20		+ Details
Please note: only 2 seat(s) available at this price					
Show all flights > Taxes, surcharges and booking fee included					
<input checked="" type="checkbox"/> Next day Lowest price on this page					
Return Tuesday 22 February 2011					
Price	Departure	Arrival	Travel time	Operated by	Flight number
EUR 616	13:32 Miami 17:40 Detroit	16:40 Detroit 07:45 [i] Amsterdam	15h30		+ Details
This flight arrives 1 day later.					
09:20 Amsterdam 11:10 Bologna					
Please note: only 2 seat(s) available at this price					
Show flights All departure times					
EUR 401	17:46 Miami 21:35 Atlanta	19:49 Atlanta 11:50 [i] Amsterdam	17h04		+ Details
This flight arrives 1 day later.					
15:00 Amsterdam 16:50 Bologna					
EUR 483	18:00 Miami	08:20 [i] Paris (CDG)	14h50		+ Details
This flight arrives 1 day later.					
13:10 Paris (CDG) 14:50 Bologna					
Show all flights > Taxes, surcharges and booking fee included					
<input checked="" type="checkbox"/> Next day Lowest price on this page					



COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

Leave	Thu, Feb 17	Hawaiian Airlines 451
	Depart:11:50am	Honolulu/Oahu, HI (HNL)
	Arrive:7:20pm	Sydney, Australia (SYD)
Non-stop Economy 10hr 30min View		
	This is an overnight flight.	
Return	Sun, Feb 27	Hawaiian Airlines 452
	Depart:9:20pm	Sydney, Australia (SYD)
	Arrive:10:10am	Honolulu/Oahu, HI (HNL)
	Non-stop Economy 9hr 50min Boeing 767 View	

Come? Parte a mezzogiorno, arriva alla sera.. ed è overnight?? Ignoranza + bug: l'algoritmo non tiene conto della linea del cambiamento di data.

IRRILEVANTE? aspetta di avere una coincidenza da prendere, e scoprire che è stata prenotata il 17 febbraio invece del 18..! Torni indietro nel tempo a prenderla?

	DEPARTING FLIGHT
	Oahu - Honolulu (HNL) to Sydney, Australia (SYD)
	Hawaiian Coach
FEB 17 HA451 Thursday, February 17, 2011 11:50 AM–Friday, February 18, 2011 7:20 PM (next day) Nonstop	
	RETURN FLIGHT
	Sydney, Australia (SYD) to Oahu - Honolulu (HNL)
	Hawaiian Coach
FEB 27 HA452 Sunday, February 27, 2011 9:20 PM–10:10 AM Nonstop	

La versione corretta

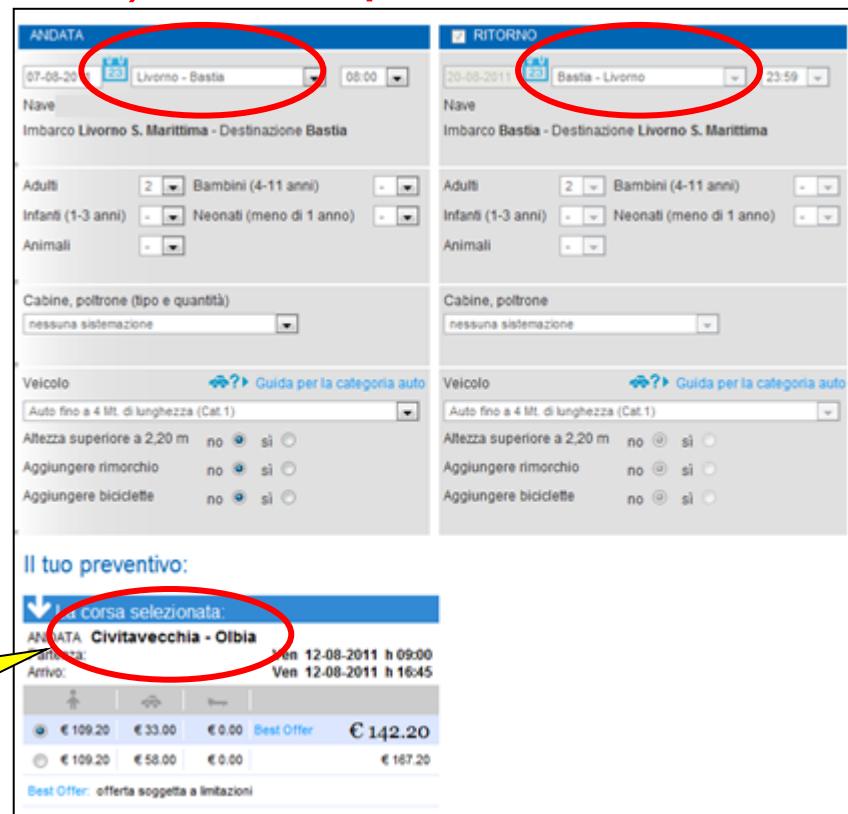
COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi



Non male il treno "not found"..
Che provenga dal binario 9 ¾ ? ☺

Tu chiedi Livorno-Bastia A/R...
...e lui ti conferma un preventivo per
Civitavecchia-Olbia??



ANDATA

07-08-2011 Livorno - Bastia 08:00

Nave

Imbarco Livorno S. Marittima - Destinazione Bastia

Adulti 2 Bambini (4-11 anni)

Infanti (1-3 anni) Neonati (meno di 1 anno)

Animali

Cabine, poltrone (tipo e quantità)
nessuna sistemazione

Veicolo Guida per la categoria auto

Auto fino a 4 Mt. di lunghezza (Cat.1)

Altezza superiore a 2,20 m no sì

Aggiungere rimorchio no sì

Aggiungere biciclette no sì

RITORNO

20-08-2011 Bastia - Livorno 23:59

Nave

Imbarco Bastia - Destinazione Livorno S. Marittima

Adulti 2 Bambini (4-11 anni)

Infanti (1-3 anni) Neonati (meno di 1 anno)

Animali

Cabine, poltrone
nessuna sistemazione

Veicolo Guida per la categoria auto

Auto fino a 4 Mt. di lunghezza (Cat.1)

Altezza superiore a 2,20 m no sì

Aggiungere rimorchio no sì

Aggiungere biciclette no sì

Il tuo preventivo:

corso selezionata:

ANDATA Civitavecchia - Olbia

Partenza: Ven 12-08-2011 h 09:00

Arrivo: Ven 12-08-2011 h 16:45

	€ 109,20	€ 33,00	€ 0,00	Best Offer	€ 142,20
<input checked="" type="radio"/>	€ 109,20	€ 58,00	€ 0,00		€ 167,20

Best Offer offerta soggetta a limitazioni



COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

OCIAL 20/11/0731: 2.320

Vogliamo parlare della data 31/07/2011
formattata come GG/MM/AAAA ma **con i dati invertiti**, sugli estratti conto di decine di migliaia di clienti ?
E nessuno se ne accorge per mesi???

..per non dire di quando un'applicazione crolla rovinosamente e nessuno ha pensato alle conseguenze..

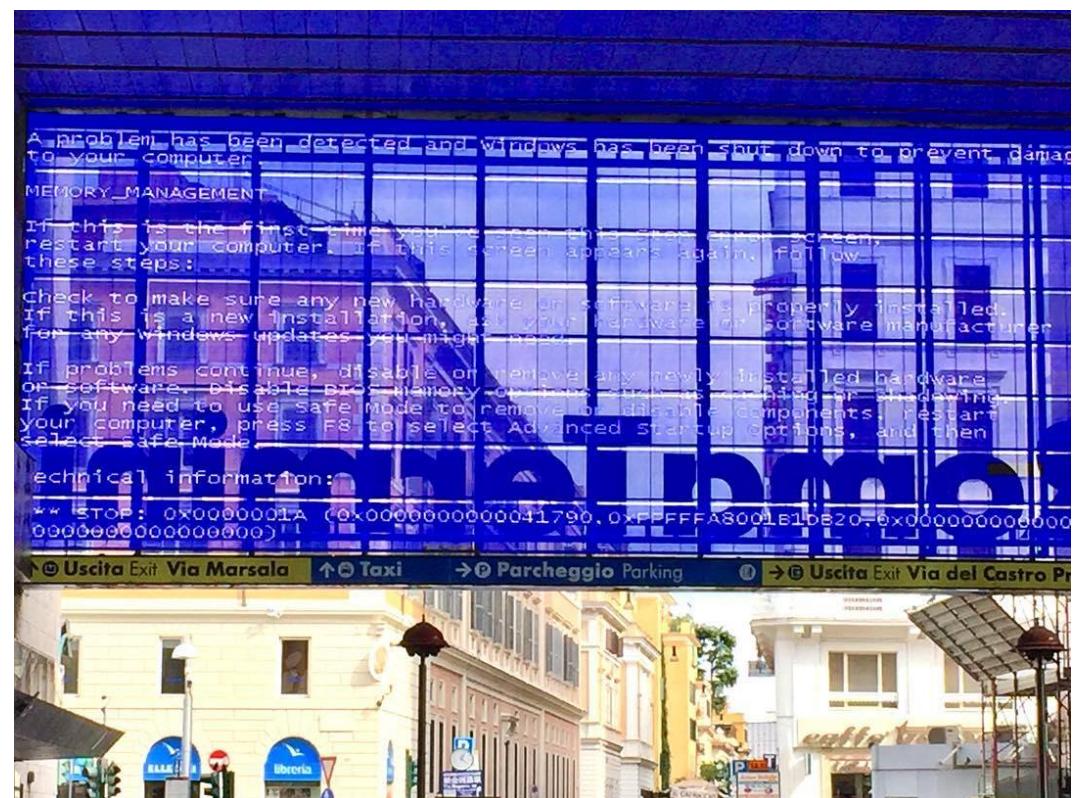




COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

..o come queste
a Roma Termini ☺





COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi



COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi



da Riveggio a Bologna Borgo Panigale

Pedaggio per auto/moto: 2,30 €

km utili ai fini del calcolo del pedaggio: 28 Km

[» Vai alle indicazioni di viaggio](#)

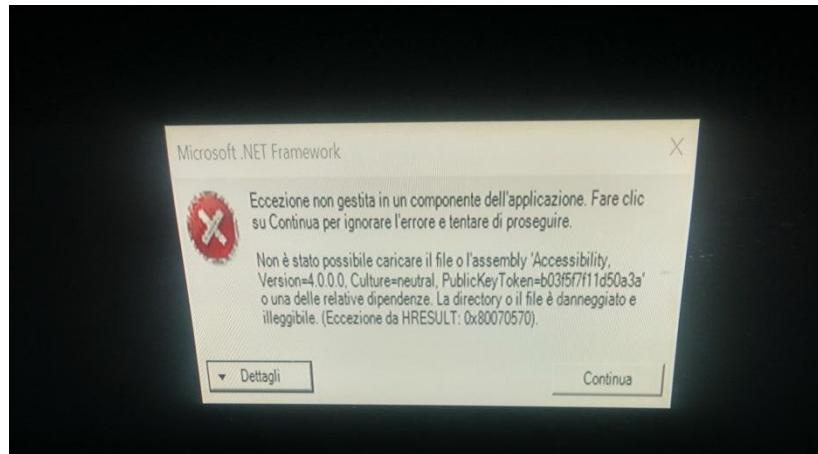
2014, BUG CORRETTO: ora giustamente fa uscire a Bologna Casalecchio e poi prendere la Tangenziale





COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi



Ah, gli sportelli automatici...



COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

Per l'occasione, abbiamo riservato per te una speciale offerta:

- Che dire? Un'offerta DAVVERO speciale... 😊
- cambio olio e filtro olio a 99€ → la tua vettura è 129€ → per il tuo ve...
- da oggi saremo aperti con orario continuato

Affrettati a cogliere questa opportunità!

Prezzi con o senza spazi?
Con la virgola o il punto?
Decidiamoci..

Su un contratto [REDACTED] è possibile [REDACTED] costo di 1,26 € al mese.
[REDACTED] che ti permette di ottenere sullo stesso contratto [REDACTED]
costo di 2,10€ al mese [REDACTED]

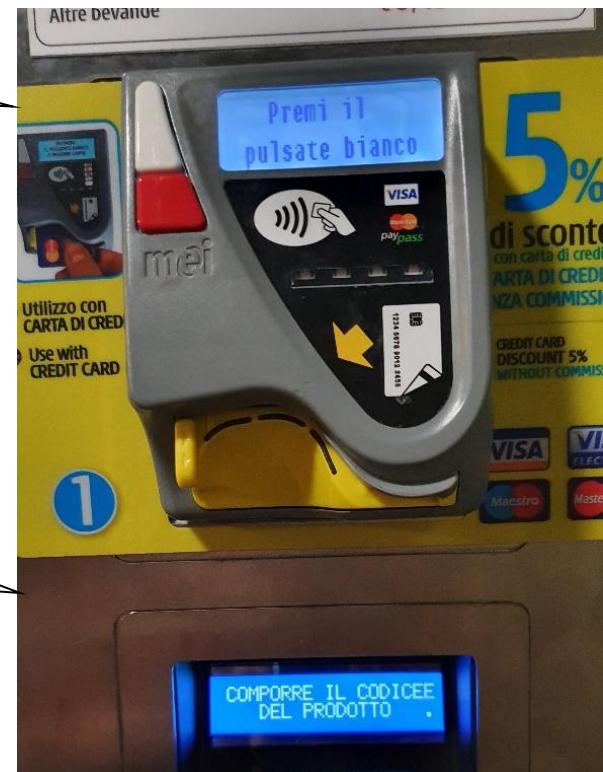
COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

Il famoso **PULSATE** bianco..

Mica male due errori nella stessa macchinetta.. e sì che si vedono! ☺

..e l'altrettanto famoso
CODICEE





COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

Ah, le carreggiate null in autostrada... **che le abbiano tolte?** ☺☺

*E sì che anche questa SI VEDE..
(sempre che qualcuno magari legga ciò che scrive...)*

Chiusure programmate

Tratta chiusa progr.

Parma-La Spezia - Carreggiata null

Parma-La Spezia Avviso: Il tratto tra gli svincoli di BERCETO e PONTREMOLI sarà chiuso in entrambe le direzioni dalle ore 22:00 del 05-11-2024 fino alle ore 05:00 del 06-11-2024 causa Lavori
Aggiornato alle 15:57 del 28/10/2024

Tratta chiusa progr.

Parma-La Spezia - Carreggiata null

Parma-La Spezia Avviso: Il tratto tra gli svincoli di BERCETO e PONTREMOLI sarà chiuso in entrambe le direzioni dalle ore 22:00 del 06-11-2024 fino alle ore 05:00 del 07-11-2024 causa Lavori
Aggiornato alle 15:58 del 28/10/2024

Chiusura Svincolo Programmata

Parma-La Spezia - Carreggiata Nord

Parma-La Spezia Avviso: Lo svincolo NODO A1 sarà chiuso in direzione Bologna dalle ore 21:00 del 04-11-2024 fino alle ore 05:00 del 05-11-2024 causa Lavori
Aggiornato alle 14:19 del 30/10/2024



COLLAUDO vs PROGETTO...?

- A volte invece sono le *assunzioni di base* non rispettate a generare assurdità al momento dell'uso..
...come in questa compagnia aerea europea

16 Posso prenotare un volo per un passeggero minorenne con il Codice di Prenotazione per due persone?

Se hai un Codice di Prenotazione per due persone uno dei due passeggeri può essere minorenne.

Il sito di prenotazione ti chiederà di inserire però la data di due passeggeri adulti. Per far sì che uno dei due passeggeri sia impostato come bambino, dovrai inserire una data falsa di nascita e indicare la data di nascita corretta nel campo "Esigenze speciali di trasporto".

Stiamo lavorando per agevolare la prenotazione anche per i passeggeri minorenni, a breve sarà possibile procedere in automatico.

Per maggiori informazioni [CLICCA QUI](#).

Ma certo! Chi avrebbe mai potuto pensare che *anche un minorenne potesse prendere un aereo..?!*

Dominio del problema
mal analizzato

Rimedio anche peggiore: MENTIRE nei moduli
e poi usare (malamente) un campo pensato per qualcos'altro!



COLLAUDO vs PROGETTO...?

- A volte invece sono le *assunzioni di base* non rispettate a generare assurdità al momento dell'uso..
...come in questo caso molto "bolognese" 😊

09:28	par	Bologna, ALDINI		Autobus 33
09:31	arr	Bologna, PORTA SAN MAMOLO		Bologna, PORTA SAN MAMOLO
09:39	par	Bologna, PORTA SAN MAMOLO		Autobus 33
09:45	arr	Bologna, CARDUCCI		Bologna, BOLOGNA STAZIONE CENTRALE

PIANIFICATORE 2016

Per andare da ALDINI a CARDUCCI basterebbe restare sul 33... e invece veniva suggerito di *scendere a porta S.Mamolo e aspettare il successivo!*

Che fosse per socializzare..? 😊

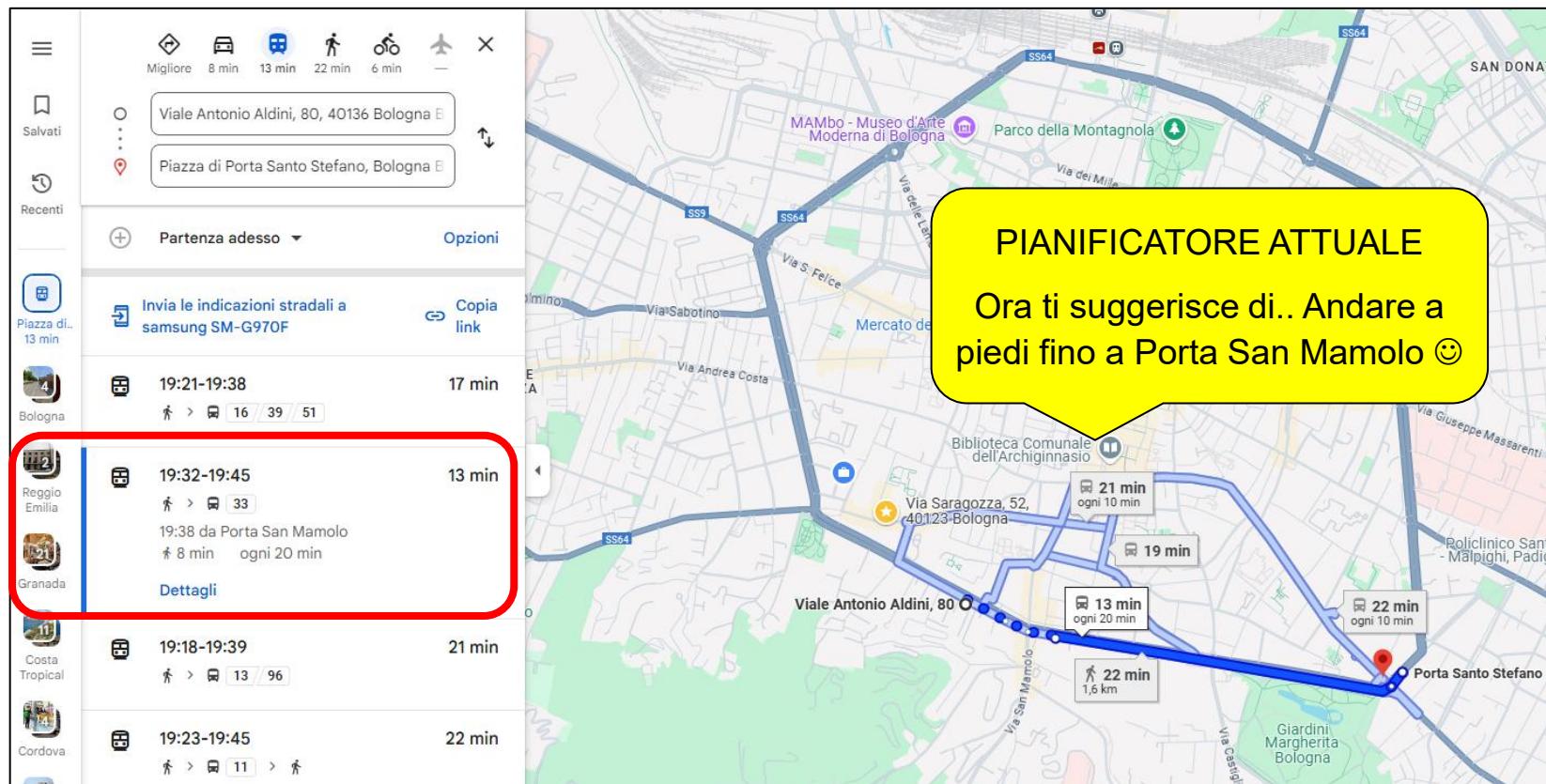
IN REALTÀ...

- chi ha scritto il software ha ipotizzato che tutte le linee abbiano due capilinea..
- *ma una linea circolare viola tale assunzione* ed è stata quindi modellata *come due tronconi* "andata": Stazione/Porta S.Mamolo "ritorno": Porta S.Mamolo/Stazione
- il software, tarato (giustamente) sull'assunzione di base, vede un itinerario composto.



COLLAUDO vs PROGETTO...?

- A volte invece sono le *assunzioni di base* non rispettate a generare assurdità al momento dell'uso..
...come in questo caso molto "bolognese" 😊



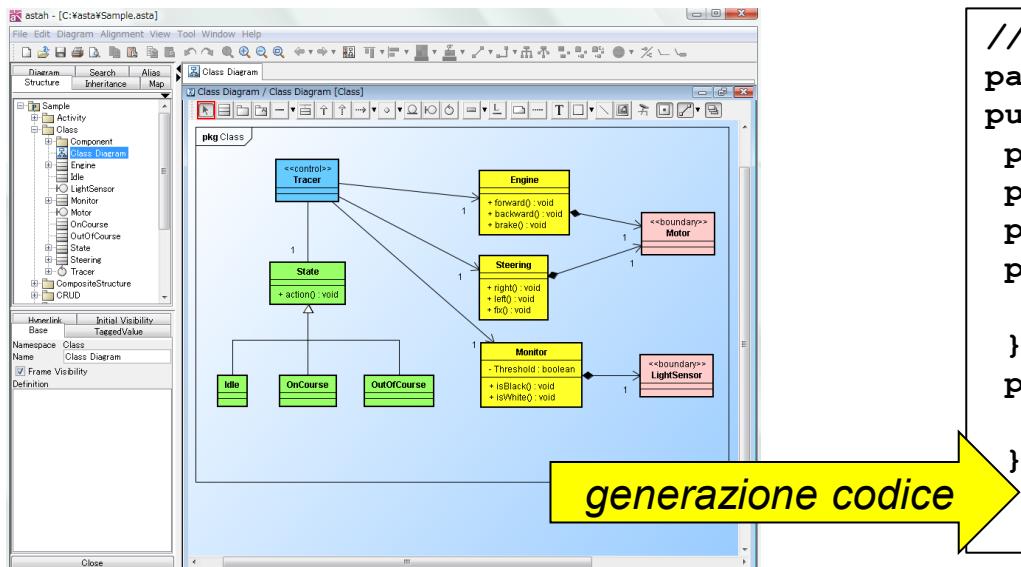


General Data Protection Regulation

- Dal 25/5/2018 ha sostituito la Data Protection Directive
- *Obbligo* di seguire i principi della GDPR
 - **privacy by design & by default**
(non solo misure tecniche, ma anche organizzative)
 - minimalità, proporzionalità
 - **anonimizzazione, pseudonimizzazione**
 - trasferimento dati fuori dalla EU (occhio al cloud!)
 - adeguatezza delle misure di sicurezza
- Non è qualcosa che si possa "aggiungere dopo", a sistema già progettato: *va considerato fin dall'inizio*
 - ma non è (solo) un vincolo: è *un'opportunità per creare valore*

LA SFIDA DELL'AUTOMATIZZAZIONE

- Già da anni esistevano strumenti che generavano *lo scheletro del codice* partendo dal *modello del sistema*
- Il progettista doveva solo aggiungere la *business logic*



```
// Monitor.java
package Class;
public class Monitor {
    private boolean Threshold;
    private Tracer tracer;
    private LightSensor lightSensor;
    public void isBlack() {
        ...
    }
    public void isWhite() {
        ...
    }
}
```



LA SFIDA DELL'AUTOMATIZZAZIONE

- Oggi, strumenti di AI in rapida avanzata stanno diventando capaci di generare intere parti di codice

scrivi un programma C che ordini un vettore di interi letto da tastiera

- ◆ Assolutamente! Ecco un programma C che ordina un vettore di interi utilizzando l'algoritmo di ordinamento Bubble Sort, uno dei più semplici da implementare:

C

```
#include <stdio.h>

void bubbleSort(int arr[], int n) {
    int i, j, temp;
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                // Scambia gli elementi se sono nell'ordine sbagliato
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

void printArray(int arr[], int size) {
```

```
int main() {
    int n;
    printf("Inserisci il numero di elementi del vettore: ");
    scanf("%d", &n);

    int arr[n];
    printf("Inserisci gli elementi del vettore:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    bubbleSort(arr, n);

    printf("Vettore ordinato:\n");
    printArray(arr, n);

    return 0;
}
```

```
Inserisci il numero di elementi del vettore: 5
Inserisci gli elementi del vettore:
3
21
12
5
91
Vettore ordinato:
3 5 12 21 91
```



LA SFIDA DELL'AUTOMATIZZAZIONE

- Oggi, strumenti di AI in rapida avanzata stanno diventando capaci di generare intere parti di codice

scrivi un programma C che implementi le liste e mostra un esempio d'uso

Assolutamente! Scrivere un programma C che implementi le liste è un ottimo modo per approfondire la comprensione delle strutture dati e dei puntatori.

Cos'è una lista in C?

Una lista è una struttura dati dinamica composta da una sequenza di elementi collegati tra loro tramite puntatori. Ogni elemento, chiamato nodo, contiene:

- Dati: Il valore che si desidera memorizzare.
- Puntatore: Un riferimento al nodo successivo nella lista.

Implementazione in C

Ecco un esempio di come

```
int main() {
    struct nodo* testa = NULL; // Inizializziamo la lista a NULL

    // Aggiungiamo alcuni elementi alla lista
    aggiungiInizio(&testa, 3);
    aggiungiInizio(&testa, 2);
    aggiungiInizio(&testa, 1);

    // Stampare la lista
    printf("Lista: ");
    stampaLista(testa);

    return 0;
}
```

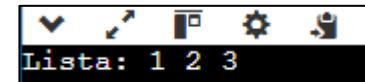
```
C

#include <stdio.h>
#include <stdlib.h>

// Struttura di un nodo della lista
struct nodo {
    int dato;
    struct nodo *next;
};

// Funzione per creare un nuovo nodo
struct nodo* creaNodo(int dato) {
    struct nodo* nuovoNodo = (struct nodo*)malloc(sizeof(struct nodo));
    nuovoNodo->dato = dato;
    nuovoNodo->next = NULL;
    return nuovoNodo;
}

// Funzione per aggiungere un nodo all'inizio della lista
void aggiungiInizio(struct nodo** testa, int dato) {
    struct nodo* nuovoNodo = creaNodo(dato);
    nuovoNodo->next = *testa;
    *testa = nuovoNodo;
}
```





LA SFIDA DELL'AUTOMATIZZAZIONE

- Senz'altro utile per alleggerirsi dai compiti ripetitivi
 - MA occhio al risultato: i generatori di testo sono progettati più per produrre *testo gradevole* che *testo realmente preciso*
 - serve sempre una mente lucida e *competente* per valutare ciò che viene prodotto!
- In prospettiva, si liberano tempo ed energie per concentrarsi ancor più sul *progetto*, la *vera dimensione dell'Ingegneria*
 - saremo sempre più *direttori d'orchestra*, non «programmatori»
 - servirà sempre *maggior competenza*, non meno



IL VOSTRO FUTURO...

- Saper progettare **sistemi di qualità** è e sarà sempre più necessario per la vostra professionalità e il vostro futuro

Assunzioni non stagionali fino a 29 anni prime 32 figure professionali ⁽¹⁾ p				24 Dicembre 2012	IL SOLE 24 ORE	estratto
Professione	Cat. ⁽²⁾	Valore assoluto 2012	In ass.			
Comessi delle vendite al minuto	PCS	23.520				
Addetti alla gestione dei magazzini e professioni assimilate	I	5.740				
Camerieri e professioni assimilate	PCS	5.560				
Addetti agli affari generali	I	5.270				
Addetti a funzioni di segreteria	I	5.150				
Personale non qualif. servizi pulizia uffici ed esercizi commerciali	PNQ	3.410				
Baristi e professioni assimilate	PCS	3.270				
Cuochi in alberghi e ristoranti	PCS	3.130				
Analisti e progettisti di software	PIS	2.890				
Addetti alla preparazione, alla cottura e alla distribuzione di cibi	PCS	2.860				
Muratori in pietra, mattoni, refrattari	OS	2.620				
Contabili e professioni assimilate	PT	2.530				
Facchini, addetti allo spostamento merci	PNQ	2.500				
Addetti sportelli assicurativi, bancari e altri intermed. finanziari	I	2.350				
Tecnici della vendita e della distribuzione	PT	2.240				

Il bilancio

Le caratteristiche delle assunzioni under 30 programmate dalle imprese per il 2012

200 mila	33%	40%
Le assunzioni di under 30 Sono gli inserimenti di giovani programmati dalle imprese per il 2012, in flezione di circa il 30% rispetto al 2011	Non stagionali Più di un'assunzione su tre non stagionale (circa 144 mila) è rivolta ad under 30, ma il numero si è dimezzato rispetto al periodo pre-crisi	Primo del Centro Nel Centro Italia 4 assunzioni su dieci sono per giovani; nel Settentrione si registrano, invece, le dinamiche peggiori rispetto al 2011 (-32%)
13 mila	24 mila	2.900
Milano e Roma al top Nelle province di Milano e Roma si registrano più assunzioni in valore assoluto (entrambe oltre quota 13 mila); all'opposto Ragusa, Prato e Pisa	Il record dei commessi I profili più richiesti sono i commessi, seguiti da magazzinieri, camerieri, impiegati e segretari (tutti con oltre 5 mila assunzioni)	Progettisti software Tra le figure «high skilled», si distinguono progettisti software (2.900), contabili (2.500) e tecnici per la vendita e la distribuzione (2.200)



GLI INTROVABILI (2012)

Indagine Unioncamere

Lavoro: ecco le figure più ricercate ma che le aziende non trovano. Situazione nelle varie Regioni

I LAVORI PIU' RICERCATI

Secondo una ricerca di Unioncamere sono oltre 65 "introvabili" nel 2012, il 16,1% del totale delle assunzioni previste che le imprese italiane intendono fare nell'anno (406.100).

Tra i livelli di istruzione, la difficoltà di reperimento è maggiore sui laureati. Dei 58.890 laureati che le imprese intendono assumire con un contratto non stagionale, uno su cinque è introvabile (pari a 11.770 unità).

Per coloro che hanno un diploma la difficoltà di reperimento è minore (27.000 unità su 166.000 assunzioni previste).

Tra i laureati è difficile per le aziende trovare programmatore, analista informatico, progettisti di impianti industriali e revisori contabili. Ma ciò non guarda alla professioni con maggiore possibilità di inserimento sul mercato del lavoro: infermiere con 3.450 assunzioni previste, 950 dei quali difficili da reperire (27,4% del totale).

Ha molte possibilità di sbocco anche l'addetto alle pulizie (1.200 assunzioni previste tra i laureati) con 2.950 assunzioni previste e il 24% di difficoltà di reperimento. Molto facile per le aziende invece tro-

Le professioni dei laureati: le più introvabili e quelle di maggiore sbocco

Nonostante la crisi e il crescente numero di soggetti in cerca di occupazione, di fronte alle assunzioni di laureati, ad esempio, le imprese incontrano elevate difficoltà di reperimento quando si trovano ad assumere determinate figure professionali. E' il caso del Progettista di sistemi informatici, professione per la quale i candidati ideali sono i laureati in Informatica: sono 900, delle mille circa previste per quest'anno, le assunzioni difficili da reperire, pari a quasi l'85% del totale. Nel campo dell'informatica, tra le prime venti professioni "introvabili" si incontrano, con tassi di difficoltà attorno al 30% delle assunzioni previste per quest'anno (e tutte con la prevalente richiesta di una laurea in Ingegneria informatica), anche il Consulente di software (un centinaio gli introvabili), l'Analista programmatore (circa 150 difficili da reperire) e il Programmatore informatico (più di 300 gli introvabili). Infine, con il 22% di difficoltà di reperimento, rientra anche lo Sviluppatore di software (più di 300 gli "introvabili").

Progettisti, ma non solo di sistemi informatici, perché quest'anno le imprese hanno difficoltà a trovare anche Progettisti meccanici (per cui si richiedono soprattutto ingegneri meccanici e navali), introvabili per quasi un terzo delle richieste, pari ad oltre 500 unità sulle 1.800 circa assunzioni previste. E sebbene i valori assoluti siano più bassi, piuttosto "irreperibili" sono anche altre figure di progettista, come quello di impianti industriali (sempre con preferenza per gli ingegneri meccanici e navali) e quello nel settore dei servizi (maggiormente richiesti i laureati in economia bancaria, finanziaria e assicurativa), dove circa il 40% delle assunzioni sono difficili da reperire (circa un centinaio in entrambi i casi).

Tra le prime posizioni per difficoltà di reperimento si collocano anche i Revisori contabili (per i quali le imprese cercano soprattutto laureati in scienze economico-aziendali, del marketing e dell'amministrazione), introvabili in oltre un terzo dei casi, quasi 300 in valori assoluti su un totale di ben oltre 700 assunzioni previste per quest'anno.

I laureati nei diversi indirizzi di Ingegneria, comunque, rappresentano i candidati ideali per ben 9 delle prime 20 professioni in cui i "dottori" sono ritenuti introvabili, a dimostrazione che questo titolo di studio offre altissime opportunità di assunzione, a patto sempre di un'adeguata preparazione dei candidati, meglio se arricchita da un'esperienza, unita anche ad una flessibilità nello svolgere le professioni per cui sono chiamati a ricoprire.

Considerando invece le professioni di maggiore sbocco occupazionale, sempre tra i "dottori",



GLI INTROVABILI (2013)

Commissione Europea & Il Sole24ore

About this Site | Contact | FAQ | Cookies | Search | Legal notice



EMPLOYMENT, SOCIAL AFFAIRS & INCLUSION

European Commission > Employment, Social Affairs & Inclusion > ... > News

[POLICIES AND ACTIVITIES](#) | [NEWS AND MULTIMEDIA](#) | [PUBLICATIONS AND DOCUMENTS](#) | [Search](#)

The top 5 occupations with best employment opportunities were:

1. Personal care workers in health services
2. Software applications developers and analysts
3. Administrative and specialised secretaries
4. Mining, manufacturing and construction supervisors
5. Primary school and early childhood teachers

GRAFICI

Assunzioni: consistenza e andamento



Le prime cinque professioni tradizionali in crescita
L'Osservatorio europeo dei posti di lavoro vacanti (Evm) segnala **cinque occupazioni** tradizionali tra le più richieste e in crescita di assunzioni: servizio alla persona nell'ambito della sanità, sviluppatore di applicazioni software e analista, segretario amministrativo e specializzato, supervisore di miniera, di produzione e di costruzione e insegnante elementare e per la prima infanzia. L'agenda per nuove competenze e nuove occupazioni nell'ambito di Europa 2020 punta il faro sul numero insufficiente di infermieri formati e un numero crescente di infermieri e ostetrici vicini all'età pensionabile. Oltre ai tecnici medici e farmaceutici sono professioni in espansione nell'ambito sanitario anche dentisti, farmacisti e fisioterapisti.



GLI INTROVABILI (2014)

Indagine Unioncamere (*Il Sole24ore*, 21/11/2014)

Dall'esperto software allo stuccatore: introvabile una professione su 10

di Alessia Tripodi 21 novembre 2014 Commenti (15)

Tweet 15

Consiglia 213

g+1 4

My24



A A



Ingegneri in Lombardia, stuccatori edili e musicisti in Veneto, accompagnatori turistici nel Lazio, organizzatori di eventi in Piemonte. È questa la mappa delle figure "introvabili" per le imprese italiane secondo i dati dell'Osservatorio Excelsior di Unioncamere presentati ieri a Verona in occasione di «Job&Orienta».

DOCUMENTI

4

▼ [I numeri del gap](#)

Secondo i dati, nel 2014 le lauree più ricercate ma più difficili da reperire sono quelle in ingegneria, soprattutto in Lombardia. Delle 530 assunzioni di ingegneri esperti di software in tutta Italia, infatti, 210 (circa il 40%) sono considerate introvabili, così come i 300 profili di analista programmatore sui 930 ricercati in tutto il Paese nel 2014 (32%). Nella classifica delle



GLI INTROVABILI (2015)

Unioncamere (*Il Sole24ore*, 27/11/2015)

In Italia ci sono 76mila profili professionali introvabili

Statisticci, ingegneri, analisti e progettisti, ma anche tanti operatori nel settore del mobile e arredamento, abbigliamento, moda ed elettronico

Adriano Palazzolo - Ven, 27/11/2015 - 15:50



Se gli statisticci, gli **ingegneri** e i laureati in materie scientifiche "sono i più difficili da reperire" tra i dottori, problemi non trascurabili sono evidenziati anche tra i diplomati e quanti hanno conseguito una qualifica professionale. Sebbene per questi ultimi le difficoltà di reperimento siano mediamente inferiori a quella dei laureati (rispettivamente, il 10 e il 9,1% del totale delle assunzioni), per alcuni indirizzi le imprese segnalano quote anche consistenti di "introvabili".

Tra i diplomati, le difficoltà salgono decisamente sopra la media nel caso dell'indirizzo **telecomunicazioni** (20,4%, pari a 150 delle 760 unità programmate), dell'indirizzo informatico (19,1%, per 970 unità delle 5.060) e dell'indirizzo grafico-pubblicitario (18,8%, per 230 delle 1.230 assunzioni previste).

Tra i candidati con qualifica di formazione professionale, i problemi nel reperimento sono particolarmente sensibili nel caso dell'indirizzo **tessile**, abbigliamento e moda, arrivando a quasi un'assunzione su tre, mentre si attestano intorno a una su 6 nel caso dell'indirizzo legno-mobile-arredo e di quello elettrotecnico.

Riguardo le professioni dei laureati, Unioncamere registra oltre 2mila opportunità per gli analisti informatici (2.110 i ricercati, la metà dei quali introvabili), seguiti dai **progettisti** di impianti per l'automazione industriale (670 totali, 210 di difficile reperimento), quindi i tecnici specialisti di applicazioni informatiche (3.130, 540 dei quali ritenuti difficili da reperire).

(*Panorama*, 16/2/2015)

I dati

Secondo le ricerche condotte dall'agenzia di selezione del personale Page Personnel, i profili professionali con 3 – 5 anni di esperienza più richiesti nel nostro Paese trovano sbocco innanzitutto nell'area trasporti e spedizioni, che si rivela un mercato in continua espansione. Ottime prospettive anche per tecnici e ingegneri, con un focus su **ingegneri informatici** e **ingegneri meccanici** che possono aspirare a professioni ben retribuite, ben tutelate e che assicurano esperienze di lavoro stimolanti ed interessanti, ma ancora molto di nicchia e poco conosciute. Infine, il settore digital si fa notare perché è caratterizzato da una **disoccupazione prossima allo zero**.



GLI INTROVABILI (2017)

Unioncamere (*Il Sole24ore*, 2017)

24 ORE

IMPRESA & TERRITORI | Giovani e lavoro: introvabili



INTROVABILI

Professione	Numeri
Operai metalmeccanici ed elettromeccanici	14.990
Tecnici informatici e della produzione	14.430
Operatori della cura estetica	6.580
Progettisti e ingegneri	5.540
Formatori	4.590
Informatici, fisici e chimici	3.820

Le "primule rosse" sono gli informatici, insieme ai formatori e agli ingegneri. Ma anche i laureati in lingue e matematica sono spesso introvabili. In barba alla disoccupazione giovanile oltre i livelli di guardia (35,4% a giugno), ci sono profili che le imprese faticano a selezionare, con il rischio di posti di lavoro destinati a restare vuoti. A segnalare il sistema informativo Excelsior di Unioncamere e Anpal (Agenzia nazionale politiche attive del lavoro), che evidenzia come su 339mila giovani che le aziende puntano a inserire tra luglio e settembre di quest'anno, il 23% è considerato difficile da "rintracciare" sul mercato.

sia sulle chance lavorative in sé, visto che i neolaureati in ingegneria godono di tassi di occupazioni e livelli retributivi superiori alla media. Secondo i numeri di AlmaLaurea, i dottori magistrali in ingegneria che lavorano, a cinque anni dal titolo, sono l'88% del totale, dietro solo a medicina (93%). Anche le retribuzioni mensili nette sono tra le più alte in assoluto: 1.717 euro, contro i 1.405 euro di media degli altri gruppi disciplinari.



GLI INTROVABILI (2018)

Unioncamere (2018)



Most sought-after jobs in Germany

The ten most frequently sought jobs in Germany:

- 1 Software developer and programmer
- 2 Electronics engineer, electrician
- 3 Healthcare worker and nurse
- 4 IT consultant, IT analyst
- 5 Economist, business administrator
- 6 Account manager, client consultant
- 7 Production assistant
- 8 Sales representative, sales assistant
- 9 Sales manager, product manager
- 10 Architect, structural engineer

Source: DEKRA Akademie 2018



GLI INTROVABILI (2020)

Unioncamere (2020)



Le professioni più richieste nell'ambito tech

La ricerca di Experis invece si è focalizzata sulle figure tech più ricercate del 2021. Si tratta di professionisti diventati fondamentali per la transizione tecnologiche delle imprese, e che possono percepire stipendi molto generosi, proprio per la loro specializzazione. Sono i **developer, machine learning specialist, cloud architect, cyber security manager, data scientist/data analyst, e system administrator**. Se vi state chiedendo cosa siano e in cosa consistano, sappiate che hanno a che fare con la crescita impetuosa dell'**intelligenza artificiale** e del **machine learning**, i nuovi metodi di programmazione che permettono a computer e robot di apprendere informazioni partendo dai dati. I compatti che si stanno dotando maggiormente di questi profili sono quelli dei **servizi, finanza, telecomunicazioni, pharma e sanità**.

Le 5 professioni più richieste, secondo Union Camere

L'analisi del Sistema Informativo Excelsior promosso da Union Camere e Ampal evidenzia che le professioni più richieste sono cinque e sono ben distribuite in tutti i settori. La prima riguarda il settore della programmazione e sviluppo è in assoluto l'ambito lavorativo più dinamico degli ultimi anni con il digitale che ha acquisito un ruolo strategico essenziale in tutte le aziende. Il **Developer**, l'**ingegnere informatico** che si occupa di progettare, sviluppare e gestire non solo siti internet e app, ma anche codici per sistemi operativi e software di uso interno all'azienda, andando a supportare così il percorso di Digital Transformation che tutte le imprese stanno affrontando.

In seconda posizione c'è il **Web Marketing Specialist** che assume un ruolo strategico sempre più decisivo, pianificando e realizzando la presenza online delle aziende. Spazio anche nell'ambito delle risorse umane con l'**HR Specialist**, professionista nella ricerca, selezione e gestione del personale. In quarta posizione ci sono le figure professionali di **qualità e controllo nella grande distribuzione** e in quinta le professioni relative alla **sicurezza digitale** come **AI Specialist, Data Protection Officer, Big Data Analyst**.

Gli indirizzi di studio più richiesti					
	Entrate previste (v.a.)	% sul totale	% diff. reperimento	% con esperienza	
Indirizzo ingegneria	31.510	16%	40%	84%	
Indirizzo economico	8.520		45%	80%	
Segnamento e formazione	8.020		31%	78%	
Indirizzo sanitario e paramedico	3.840		40%	91%	
Indirizzo scienze matematiche, fisiche e informatiche	2.930		44%	93%	
	1.820		56%	91%	



GLI INTROVABILI (2021)

Unioncamere (2021-22)

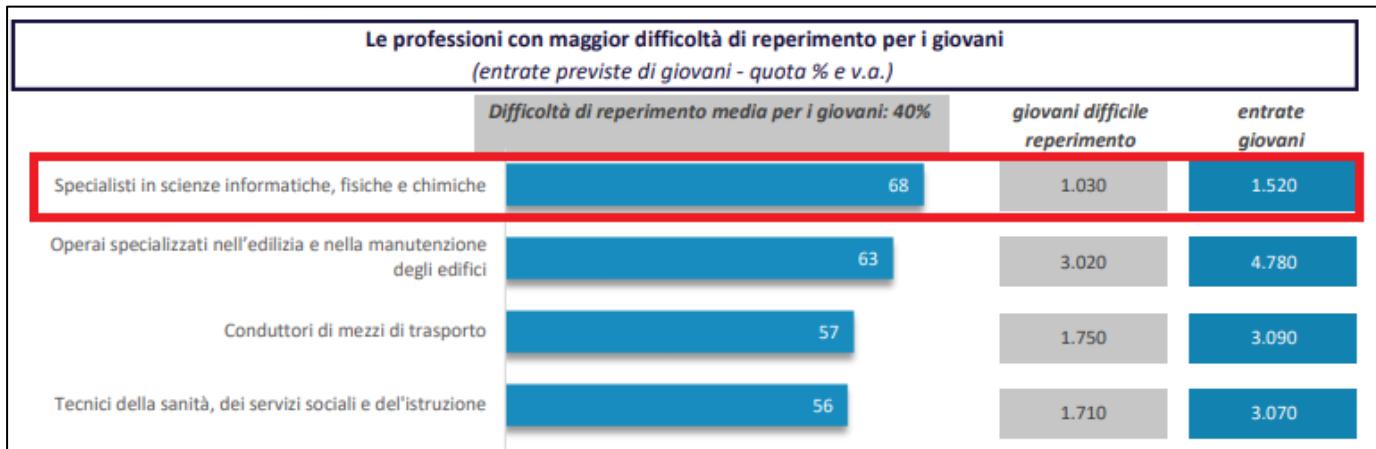
Gli indirizzi di studio più richiesti				
	Entrate previste (v.a.)	% sul totale	% diff. reperimento	% con esperienza
UNIVERSITARIO	88.330	19%	43%	87%
Indirizzo economico	25.360	27%	82%	
Indirizzo ingegneria	21.670	53%	90%	
Indirizzo insegnamento e formazione	9.580	42%	92%	
Indirizzo sanitario e paramedico	6.960	60%	91%	
Indirizzo scienze matematiche, fisiche e informatiche	6.560	70%	86%	

I profili STEM maggiormente difficili da reperire

Q: Di quali figure in ambito STEM in particolare avrebbe bisogno la sua azienda?



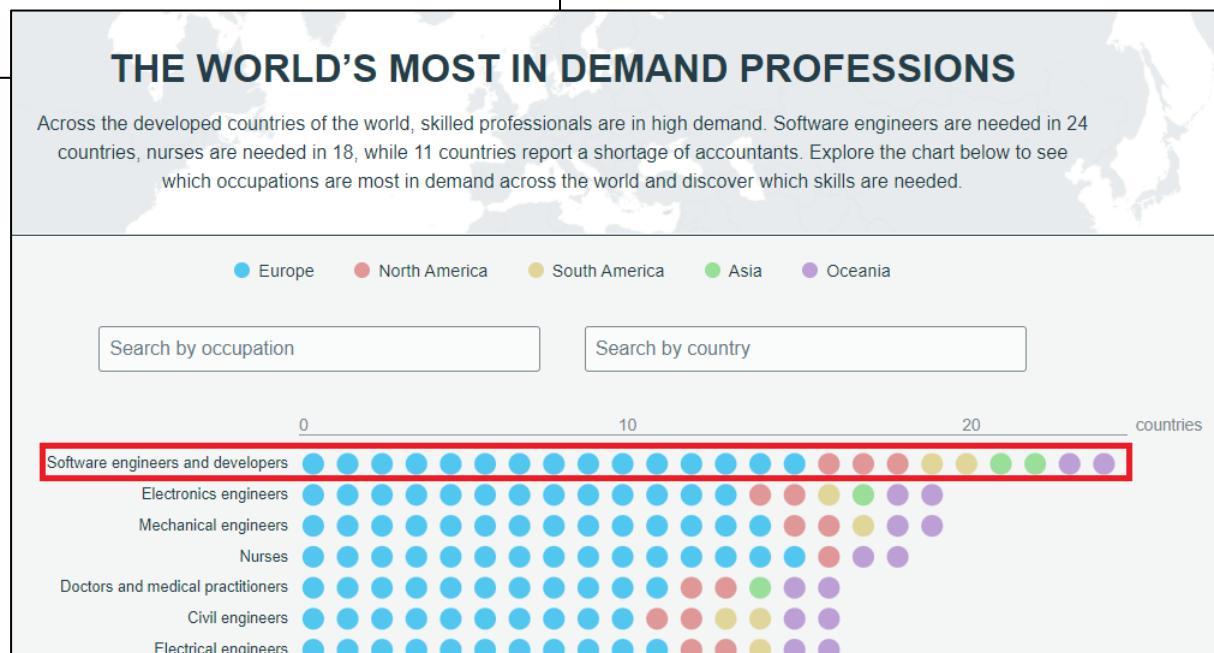
Imprese che hanno cercato risorse STEM ma non le hanno trovate tutte, in collaborazione con SWG e grazie al contributo di Monitor Deloitte





GLI INTROVABILI (2022)

LE PROFESSIONI PIÙ RICHIESTE NEL MESE DI GENNAIO 2022 *				
secondo la classificazione delle professioni - ISTAT 2011 (3 digit)				
	ENTRATE PREVISTE GENNAIO 2022	Ripartiz x 1000	Difficoltà a reperire (%)	ENTRATE PREVISTE gennaio - marzo 2022
ENTRATE	457.650	1.000	38,6	1.158.150
DIRIGENTI	1.340	2,9	54,0	2.510
PROFESSIONI INTELLETTUALI, SCIENTIFICHE E DI ELEVATA SPECIALIZZAZIONE	35.580	77,7	43,2	77.350
Specialisti in scienze matematiche, informatiche, chimiche, fisiche e naturali	5.920	12,9	61,9	13.470
Ingegneri e professioni assimilate	7.400	16,2	52,9	15.280
Specialisti delle scienze gestionali, commerciali e bancarie	7.490	16,4	33,8	14.550
Specialisti in discipline artistiche espressive	2.100	4,6	17,0	4.820
Altri specialisti dell'educazione e della formazione	2.500	5,5	36,9	5.150
PROFESSIONI TECNICHE				
Tecnici informatici, telematici e delle telecomunicazioni				
Tecnici in campo ingegneristico				



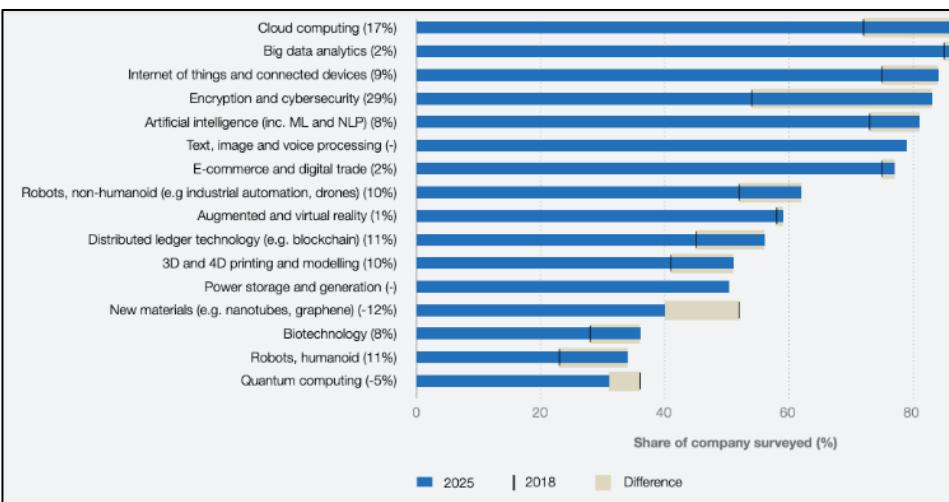


GLI INTROVABILI (2023)

LE PROFESSIONI PIÙ RICHIESTE NEL MESE DI GENNAIO 2023 *

secondo la classificazione delle professioni - ISTAT 2011 (3 digit)

	ENTRATE PREVISTE GENNAIO 2023	Ripartiz. x 1000	Difficoltà a reperire (v.a.)	Difficoltà a reperire (%)	ENTRATE PREVISTE gennaio - marzo 2023
ENTRATE	503.670	1.000	229.455	45,6	1.307.070
DIRIGENTI	1.840	3,6	1.216	66,1	4.040
PROFESSIONI INTELLETTUALI, SCIENTIFICHE E DI ELEVATA SPECIALIZZAZIONE	40.540	80,5	19.255	47,5	93.560
Specialisti in scienze matematiche, informatiche, chimiche, fisiche e naturali	7.550	15,0	3.667	48,6	18.730
Ingegneri e professioni assimilate	7.890	15,7	4.179	53,0	18.430
Specialisti nelle scienze della vita	3.750	7,4	2.632	70,2	7.610
Specialisti delle scienze gestionali, commerciali e bancarie	7.330	14,5	2.736	37,3	16.730
Altri specialisti dell'educazione e della formazione	3.250	6,5	1.541	47,4	6.480
PROFESSIONI TECNICHE	86.020	170,8	44.415	51,6	201.410
Tecnici informatici, telematici e delle telecomunicazioni	10.940	21,7	6.361	58,1	26.560
Tecnici in campo ingegneristico	8.690	17,2	5.192	59,7	22.250
Tecnici della gestione dei processi produttivi di beni e servizi	6.510	12,9	4.161	63,9	16.610
Tecnici della salute	14.830	29,4	9.311	62,8	31.380
Tecnici dell'organizzazione e dell'amministrazione delle attività produttive	8.430	16,7	3.579	42,5	17.930
Tecnici delle attività finanziarie e assicurative	2.960	5,9	1.210	40,9	6.230



Gli indirizzi di studio più richiesti

Indirizzo	Entrate previste (v.a.)	% sul totale
Indirizzo economico	96.060	19%
Indirizzo ingegneria	25.940	23.300
amento e formazione	11.500	
anitario e paramedico	9.220	

Indirizzo scienze matematiche, fisiche e informatiche 5.710

TOP FIVE IN-DEMAND ROLES

Whoever holds the talent holds the future. These are the most sought-after professions by employers globally.

1	IT & DATA	
2	SALES & MARKETING	
3	OPERATIONS & LOGISTICS	
4	MANUFACTURING & PRODUCTION	
5	CUSTOMER FACING & FRONT OFFICE	



GLI INTROVABILI (2024)

Forbes

pronta alle sfide del futuro.

Ma andiamo a vedere nel dettaglio le 15 professioni lavorative più richieste:

1. **Addetto allo sviluppo commerciale** (può essere indicato anche come *Sales development representative*) – Che cosa fa: Gli addetti allo sviluppo commerciale cercano potenziali nuovi clienti, proponendo prodotti idonei alle loro esigenze durante tutto il ciclo di vendita. Aree principali dove si assume: Milano, Roma, Torino | Attuale distribuzione per genere: 52% donne; 47% uomini
2. **Ingegnere dell'intelligenza artificiale** (può essere indicato anche come *Artificial intelligence engineer o Machine learning engineer*) – Che cosa fa: Gli ingegneri dell'intelligenza artificiale utilizzano l'AI e modelli di apprendimento automatico per sviluppare soluzioni che possono aiutare le aziende ad aumentare la propria efficienza. Aree principali dove si assume: Roma, Milano, Torino | Attuale distribuzione per genere: 25% donne; 75% uomini
3. **Analista Soc** (può essere indicato anche come *SOC analyst o cyber security analyst*) – Che cosa fa: Gli analisti SOC si occupano della sicurezza informatica aziendale, monitorando le attività su siti web, server e database alla ricerca di eventuali minacce. Aree principali dove si assume: Roma, Milano, Napoli | Attuale distribuzione per genere: 25% donne; 75% uomini
4. **Consulente per la sostenibilità** (può essere indicato come *sustainability specialist*) – Che cosa fa: I sustainability specialist creano, supervisionano e implementano strategie volte al raggiungimento degli obiettivi di sostenibilità delle organizzazioni. Aree principali dove si assume: Milano, Roma, Torino | Attuale distribuzione per genere: 65% donne; 35% uomini
5. **Ingegnere del Cloud** (può essere indicato come *cloud engineer*) – Che cosa fa: I cloud engineer svolgono in azienda attività legate al cloud computing, come il monitoraggio e il mantenimento dell'infrastruttura e dei server. Aree principali dove si assume: Milano, Roma, Torino | Attuale distribuzione per genere: 19% donne; 81% uomini
6. **Ingegnere dei dati** (può essere indicato come *data engineer*) – Che cosa fa: I data engineer lavorano con set di dati aziendali, creando programmi e sistemi in grado di acquisirli, aggregarli e trasformarli. Aree principali dove si assume: Milano, Roma, Torino | Attuale distribuzione per genere: 29% donne
7. **Responsabile acquisti** (può essere indicato anche come *Procurement buyer o Procurement specialist*) – Che cosa fa: I responsabili acquisti gestiscono il processo di approvvigionamento di beni e servizi necessari al raggiungimento degli obiettivi aziendali, definendo strategie e modalità d'acquisto. Aree principali dove si assume: Roma, Milano, Bologna | Attuale distribuzione per genere: 55% donne; 45% uomini
8. **Ingegnere della cibersicurezza** (può essere indicato come *cyber security engineer*) – Che cosa fa: I cyber security engineer creano e implementano soluzioni volte a proteggere i sistemi informatici da attacchi hacker. Aree principali dove si assume: Roma, Milano, Napoli | Attuale distribuzione per genere: 24% donne; 76% uomini
9. **Consulente cloud** (può essere indicato come *cloud consultant*) – Che cosa fa: I consulenti cloud seguono i servizi di cloud computing nelle aziende, aiutandole a prendere decisioni a vantaggio delle proprie operazioni. Aree



GLI INTROVABILI (2024)

Jobs in the US after MS

International students who complete their MS degrees in the US often have access to various job opportunities. The US job market values advanced degrees and international perspectives, making MS degree holders desirable candidates. Industries such as technology, engineering, healthcare, finance, and research often seek individuals with specialised knowledge and skills gained through MS programmes. Depending on the specialisation, MS graduates can gain access to some of the fastest-growing fields in the country. Here are some jobs available for such individuals.

1. Software Engineer/Developer
2. Data Scientist/Analyst
3. Financial Analyst
4. Project Manager
5. Marketing Manager
6. Research Scientist
7. Civil Engineer

GLI INDIRIZZI DI LAUREA DI PIU' DIFFICILE REPERIMENTO (valori %)



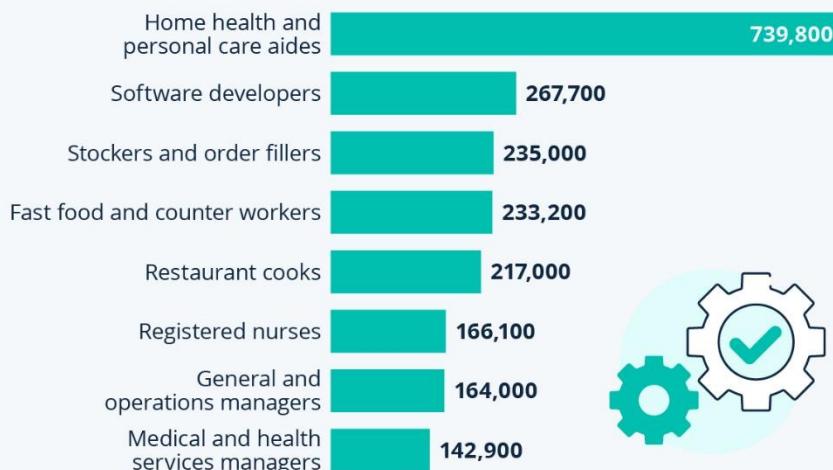
Fonte: Unioncamere – Ministero del Lavoro e delle Politiche sociali, Sistema Informativo Excelsior, 2024



GLI INTROVABILI (2025)

Wanted: The Most In-Demand Jobs of the Next Decade

Occupations with the largest projected increase in employment in the U.S. between 2024 and 2034



Source: U.S. Bureau of Labor Statistics



statista

TOP 10 HIGHEST PAYING JOBS IN 2025

1		Chief Executive Officer (CEO)	\$850,000+
2		Surgeon	\$400,000+
3		Anesthesiologist	\$380,000+
4		Investment Banker	\$300,000+
5		Corporate Lawyer	\$250,000+
6		AI Engineer	\$190,000+
7		Data Scientist	\$180,000+
8		Software Architect	\$180,000+
9		Pilot (Long-haul)	\$160,000+



GLI INTROVABILI (2025)

Previsione di assunzione delle figure con alte competenze informatiche, digitali e tecnologie 4.0

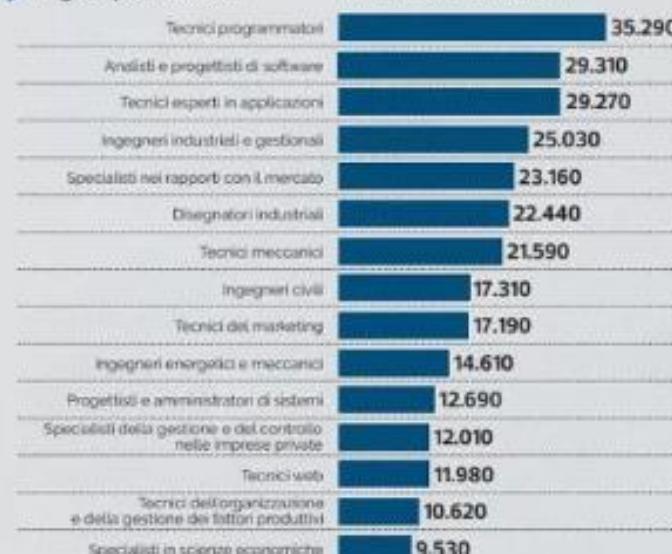
1.277.000

23,2%

del totale delle assunzioni

► Figure più richieste

TOTALE ASSUNZIONI (VAL. ASS.)



Profili difficili da trovare sul mercato

562.120

44%

del totale

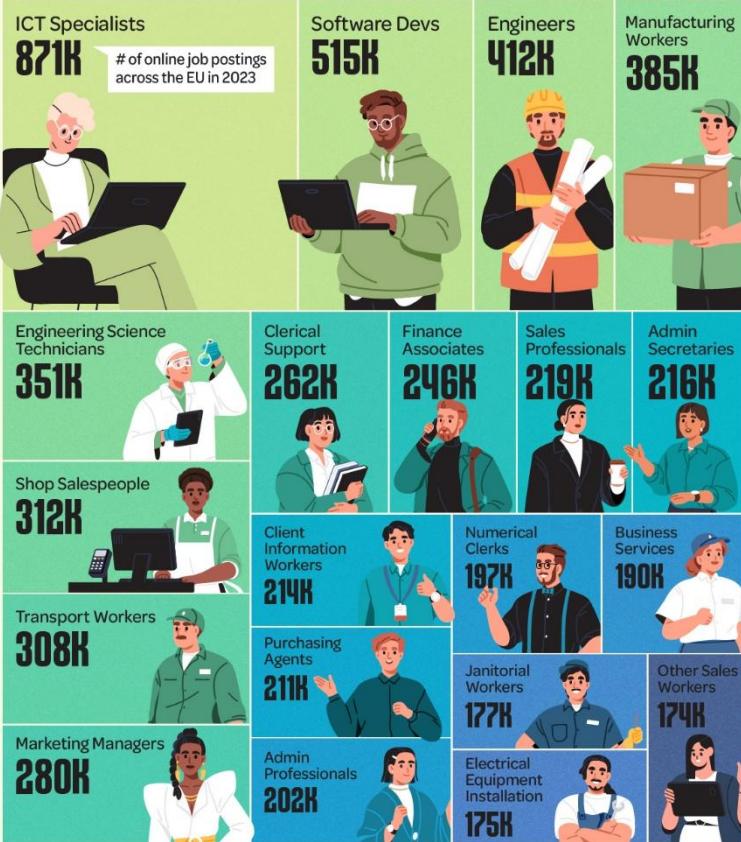
► Incidenza di professioni dell'innovazione sul totale delle assunzioni previste per microsettore, 2023 (Val. %)





GLI INTROVABILI (2025)

Europe's Most In-Demand Jobs



VISUAL
CAPITALIST

A single job advertisement can represent multiple vacancies or vice-versa.
Figures rounded. Source: Eurostat



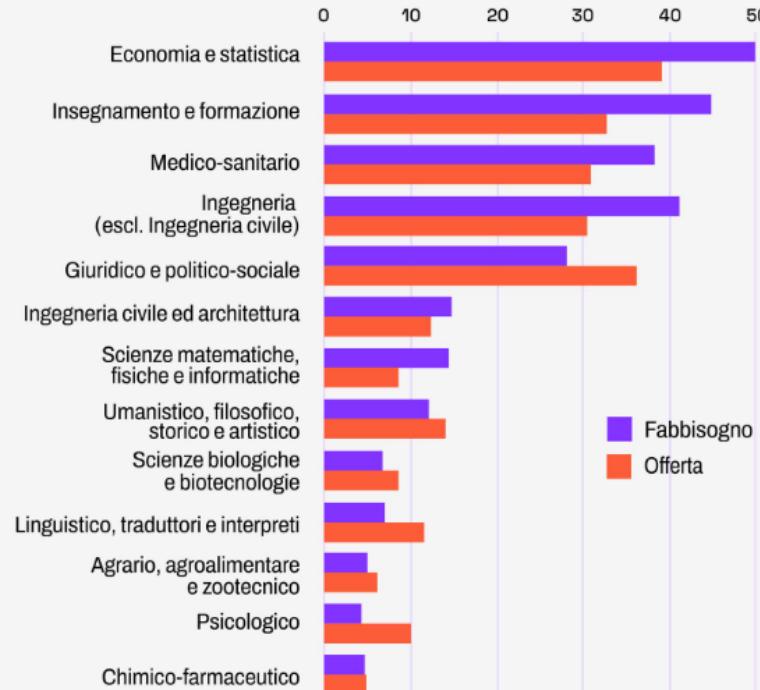
Where Data Tells the Story



Quali saranno le lauree più richieste nei *prossimi 5 anni*?



Offerta e fabbisogno di persone con una formazione terziaria* per settore nel periodo 2024-2028, in migliaia



*Il dato include le persone che hanno terminato un percorso di studi in Università, in un Istituto tecnico superiore o in una scuola di Alta formazione artistica, musicale e coreutica

Fonte: Previsione dei fabbisogni occupazionali e professionali in Italia a medio termine, Unioncamere



DIVENTARE INGEGNERI INFORMATICI

- Per diventare validi ingegneri informatici bisogna "solo" *saper programmare bene*, in linguaggi moderni e di moda?
- NO !**
- Serve molto di più:
 - *saper pensare*, riflettere sulle cose (anche filosoficamente)
 - *saper impostare l'attività di progetto* di un sistema complesso
 - comprendere che il progetto non è il codice
 - comprendere che progettare non è implementare
 - interiorizzare l'idea che **il collaudo è parte integrante della propria attività**, non qualcosa "da fare dopo" o delegabile ad altri



IL PROBLEMA DEL PROGETTO

Problema:

spesso si pensa (solo) in base ai linguaggi conosciuti

MA

i linguaggi "classici" (C, ...) non offrono strumenti mentali tarati per affrontare la progettazione di sistemi software complessi

Problema del "foglio bianco":

- da dove si parte?
- come si comincia a "immaginare" la soluzione ?
- le metodologie, gli approcci usati per affrontare problemi "piccoli" (algoritmi) scalano su problemi "grandi" (sistemi software)?
Si riescono ancora a usare con efficacia?



IL TEMA DELLA SCALABILITÀ

- L'essere umano ha la naturale tendenza a pensare che tutto sia *scalabile*
 - ha funzionato bene su 3 elementi, perché non dovrebbe funzionare altrettanto bene con 300 o 3000 o 300000..?
 - ho sempre usato questo approccio per progettare finora, perché non dovrebbe reggere anche più in grande domani?
 - ho sempre fatto così, perché mi vuoi costringere a cambiare?
- Purtroppo, ciò è spesso una *illusione mentale*
 - l'essere umano è *intrinsecamente pigro*: ripensare ai propri approcci, cambiare abitudini e metodi di lavoro *costa fatica*
 - ..ma nei fatti *gli strumenti (fisici e mentali) quasi mai scalano all'aumentare della complessità!*
 - per avvitare cinque viti basta un cacciavite.. *ma per la Tour Eiffel?*



LINGUAGGI & CONTESTO

I linguaggi "classici", come il C:

- sono nati quando l'informatica era allo *stadio artigianale*
 - l'attenzione era sul definire un "linguaggio per codificare algoritmi"
 - i programmi erano piccoli sistemi, gestiti tipicamente da una singola persona (o poche persone) secondo il proprio stile di lavoro
 - l'hardware non era certo quello di oggi..!
- **non erano pensati per supportare *un'attività di progetto***
 - il progetto "era" il codice (o al massimo un diagramma di flusso..)
- **non erano pensati per supportare *una attività di collaudo sistematica* di carattere ingegneristico - industriale**
 - il collaudo (quando c'era..) era un'attività "*fai da te*", non strumentata
 - i test (quando esistevano) erano tipicamente *mischinati nel codice*



LINGUAGGI & CONTESTO

Un tale approccio oggi **non può reggere «nel grande»:**

- è *impensabile* progettare e mantenere col "fai da te" un sistema complesso, costoso e spesso "mission critical"
- il linguaggio diviene solo *uno* degli strumenti di una *attività ben più ampia di progetto e sviluppo*, che va organizzata
- soprattutto, *il linguaggio non può più limitarsi a fornire solo gli elementi linguistici*, lasciando al progettista l'onere di affrontare tutta la complessità:
deve fornire concetti, metafore e strumenti adatti a governare la complessità dei sistemi software



VERSO NUOVI LINGUAGGI ED ECOSISTEMI

I modelli di progettazione e i linguaggi moderni sono nati per cercare di superare questo limite

- al di là delle differenze, essi introducono *nuovi concetti, nuove metafore e nuovi strumenti mentali* per riuscire a *concepire e governare il progetto, lo sviluppo e il collaudo* di sistemi software complessi
- la varietà di metafore e concetti è *la ragione di fondo per l'esistenza di tanti linguaggi* e il loro continuo proliferare

I modelli e linguaggi A OGGETTI costituiscono uno dei *maggiori casi di successo* degli ultimi decenni
ultimamente sempre più spesso *BLENDED* con concetti chiave provenienti dai *LINGUAGGI FUNZIONALI*



NEW LIFE TO OLD IDEAS

Paradigmi funzionali alla riscossa

- Per decenni, i paradigmi di programmazione e i linguaggi *funzionali*, nonostante il notevole contributo di idee "forti", sono stati penalizzati dalla *sintassi* poco *user-friendly*, *inadatta al "grande pubblico"*
 - spesso confinati nell'accademia, o in gruppi di "affezionati utenti"
 - spesso visti "con sospetto" dai "veri programmati" (...)
- Ma nell'ultimo decennio *l'aumento della complessità dei sistemi* ha dato loro nuova linfa in nuove, moderne forme
 - **LINGUAGGI BLENDED:** *idee originate dal paradigma funzionale re-innestate e adattate sui moderni linguaggi "main stream"* – sfruttando le infrastrutture esistenti ☺



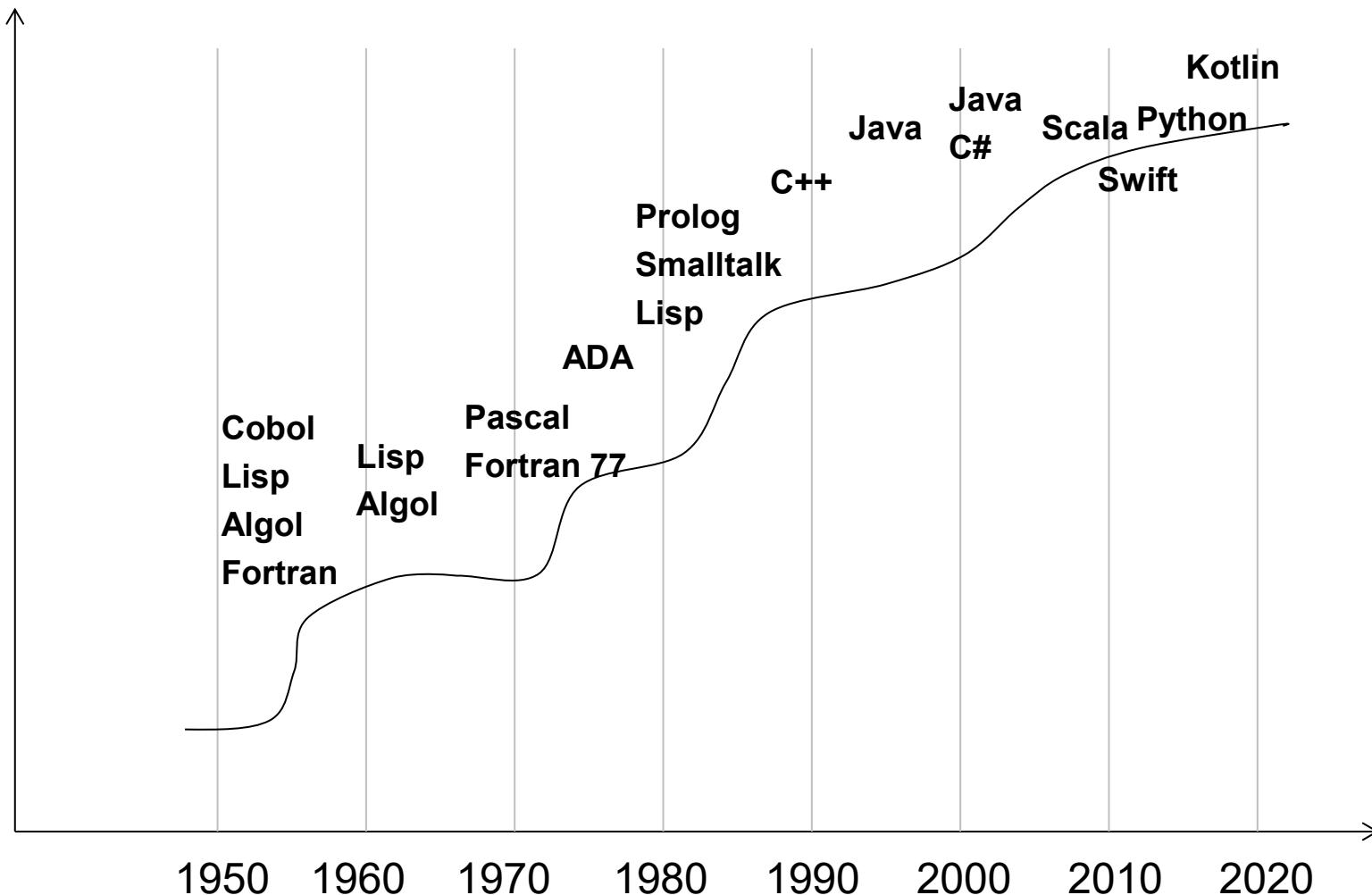
NUOVI PUNTI DI VISTA, NUOVE ISPIRAZIONI

Alcune idee "from the functional programming world"

- distinzione variabili / valori (**var** vs **val**)
- abolizione dei tipi primitivi: "*everything is an object*"
- costrutti come espressioni: "*everything is an expression*"
- strutture dati immodificabili: "*compute by synthesis*"
- funzioni come first-class entities
 - *chiusure, lambda expression*
 - *operatori come funzioni, operatori come metodi*
 - *lazy evaluation*
- stile più conciso
- ...



VERSO NUOVI LINGUAGGI





VERSO NUOVI LINGUAGGI

Scala

Scala 1.0	2004
Scala 2.0	2006
...	
...	
Scala 3.3.1	2023
Scala 3.7.2	2025

C# history

C# 1.0	2002
C# 2.0	2005
C# 3.0	2007
C# 4.0	2010
C# 5.0	2012
C# 6.0	2015
C# 7.0	2017
C# 8.0	2019
C# 9.0	2020
C# 10.0	2021
C# 11.0	2022
C# 12.0	2023
C# 13.0	2024
C# 14.0	2025

Kotlin

kotlin 1.0	2016
kotlin 1.2	2017
kotlin 1.3	2018
Kotlin 1.6	2021
...	
Kotlin 2.2.21	2025

Java history

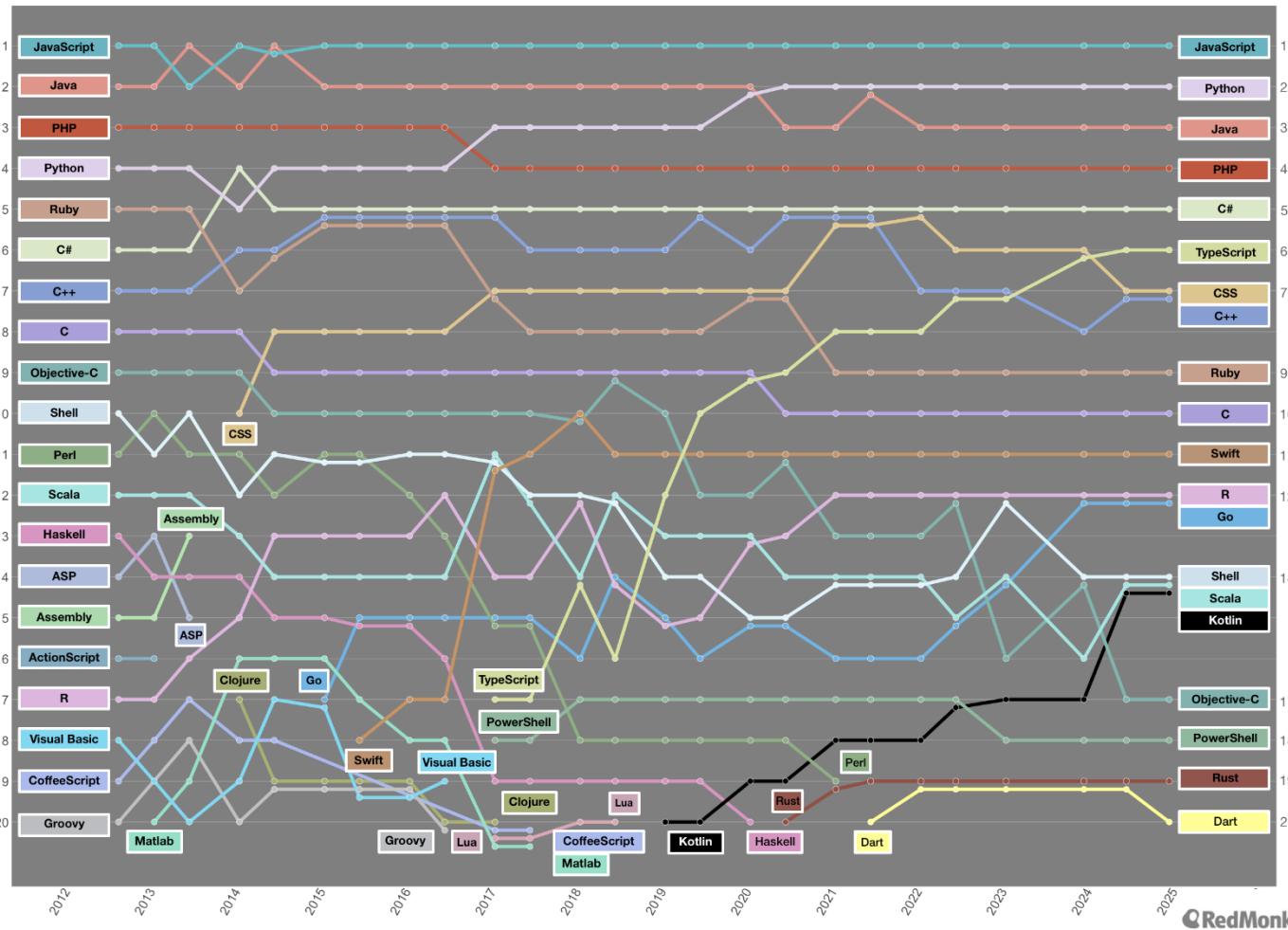
JDK 1.0 (1996)
JDK 1.1 (1997)
J2SE 1.2 (1998)
J2SE 1.3 (2000)
J2SE 1.4 (2002)
J2SE 5.0 (2004)
Java SE 6 (2006)
Java SE 7 (2011)
Java SE 8 (2014)[LTS]
Java SE 11 (2018)[LTS]
...
Java SE 21 (2023) [LTS]
Java SE 23 (2024)
Java SE 25 (2025) [LTS]



VERSO NUOVI LINGUAGGI

RedMonk Language Rankings

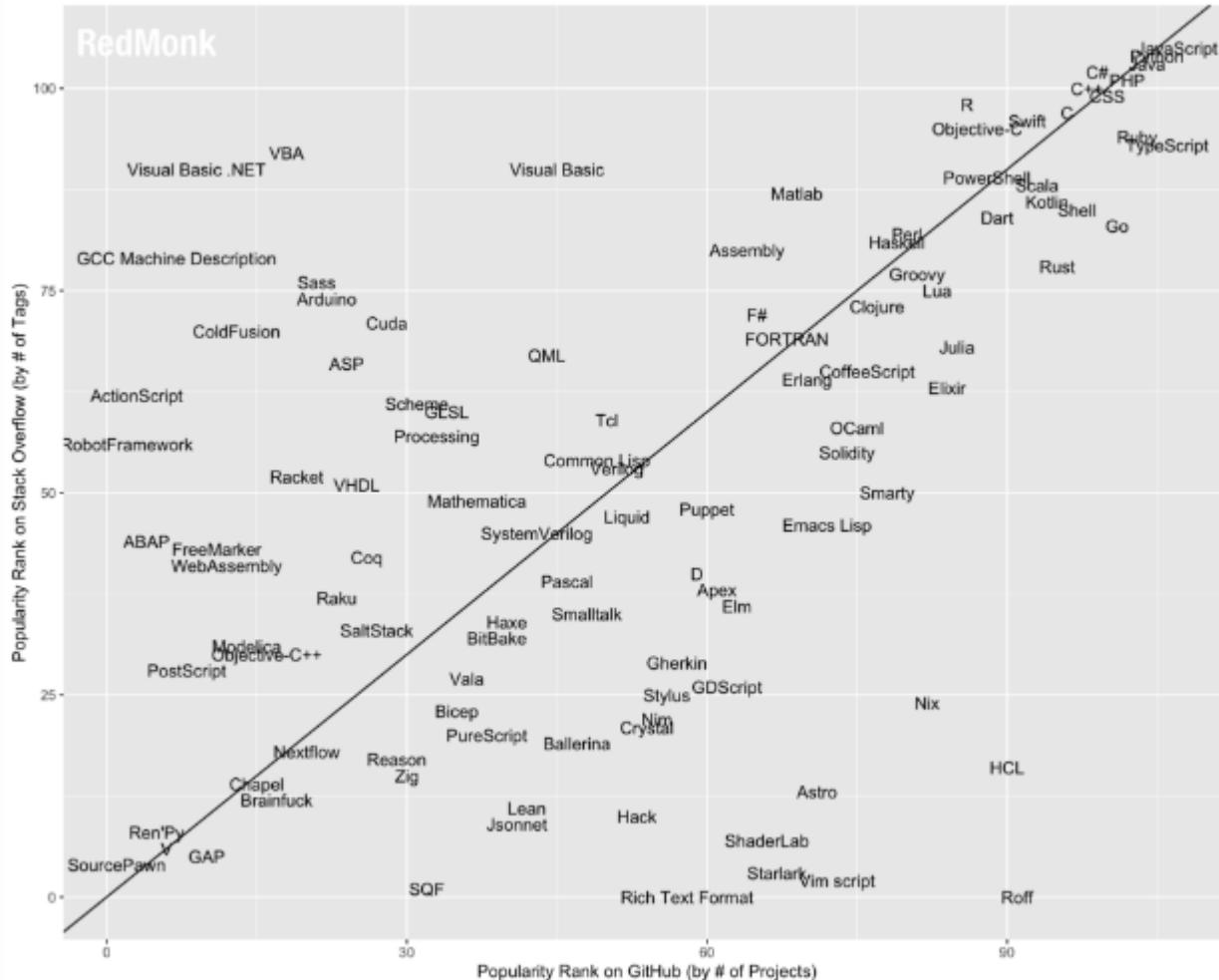
September 2012 - December 2024





VERSO NUOVI LINGUAGGI

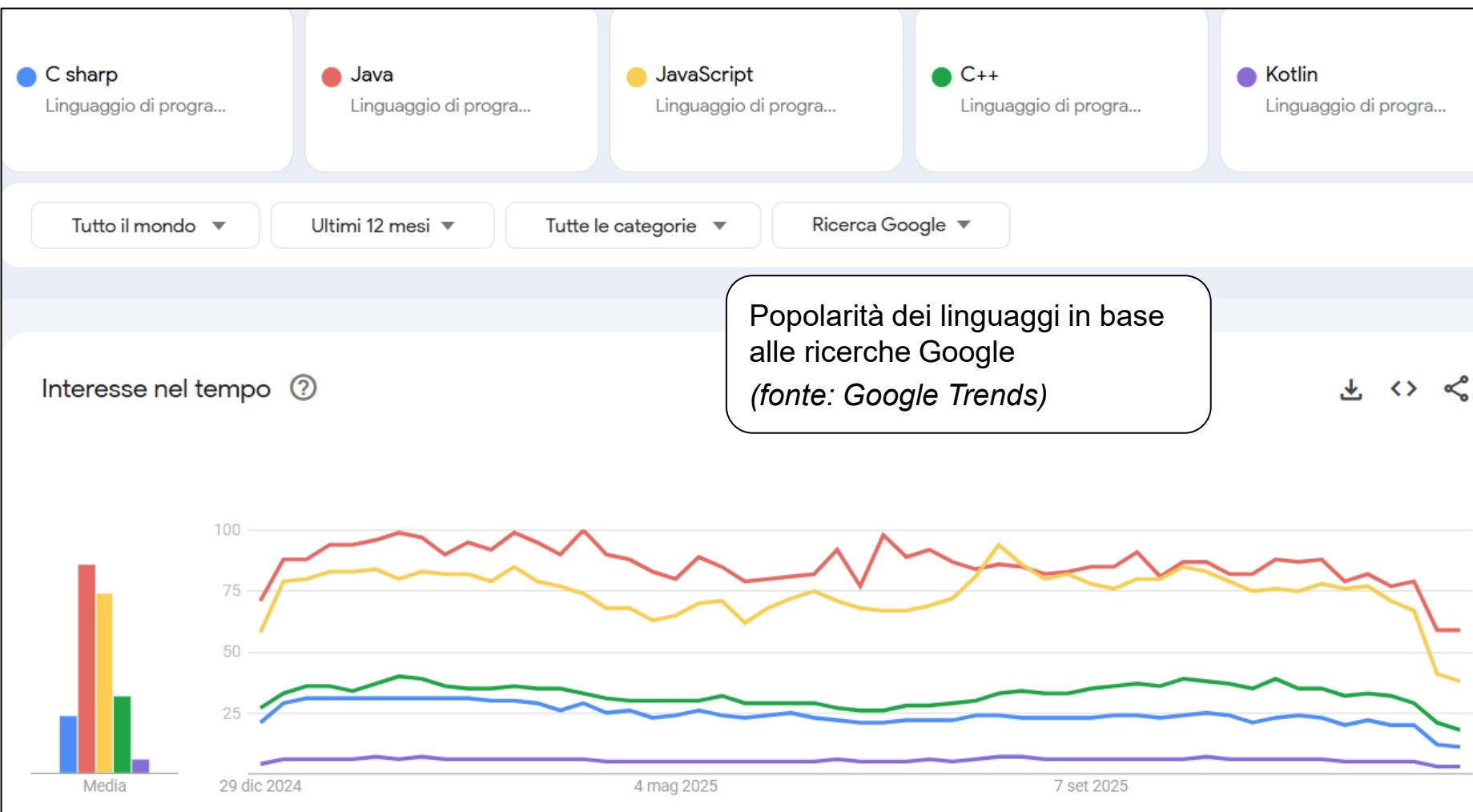
RedMonk Q124 Programming Language Rankings



- 1 JavaScript
- 2 Python
- 3 Java
- 4 PHP
- 5 C#
- 6 TypeScript
- 6 CSS
- 8 C++
- 9 Ruby
- 10 C
- 11 Swift
- 12 Go
- 12 R
- 14 Shell
- 14 Objective-C
- 16 Scala
- 17 Kotlin
- 18 PowerShell
- 19 Rust
- 19 Dart

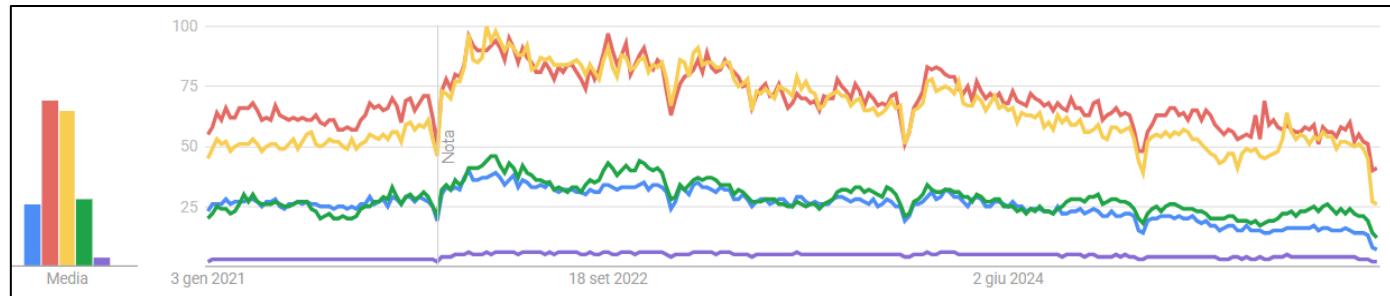
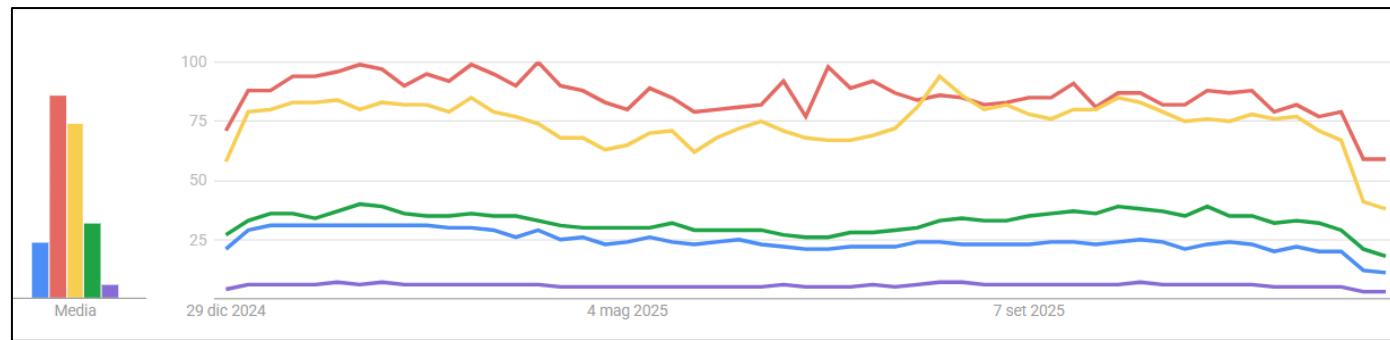
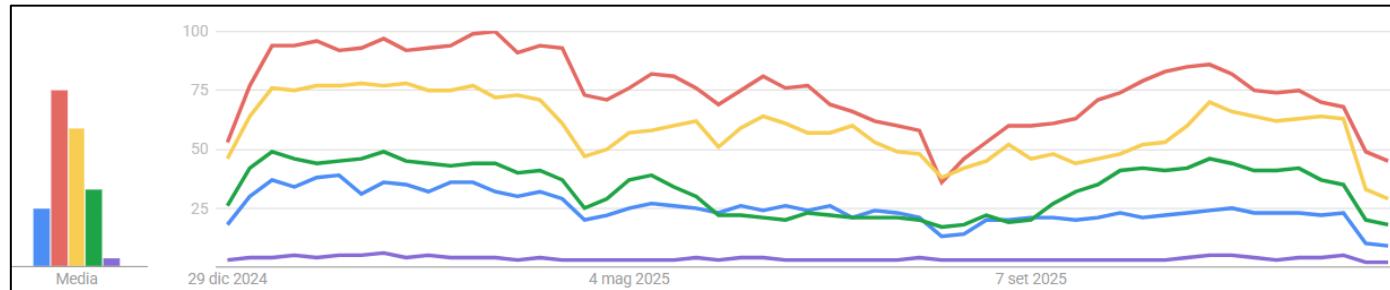


VERSO NUOVI LINGUAGGI





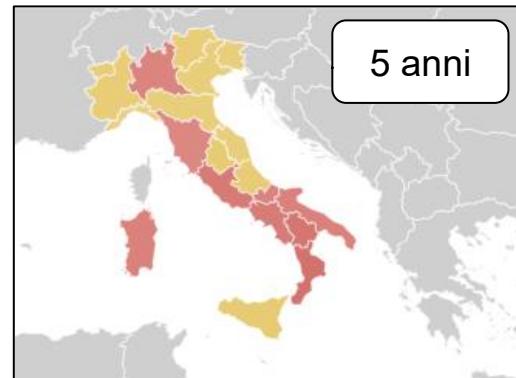
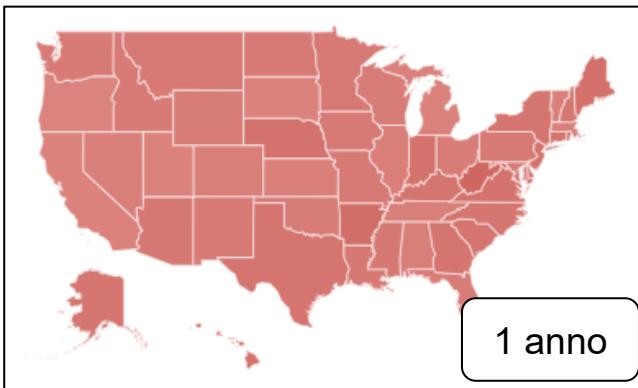
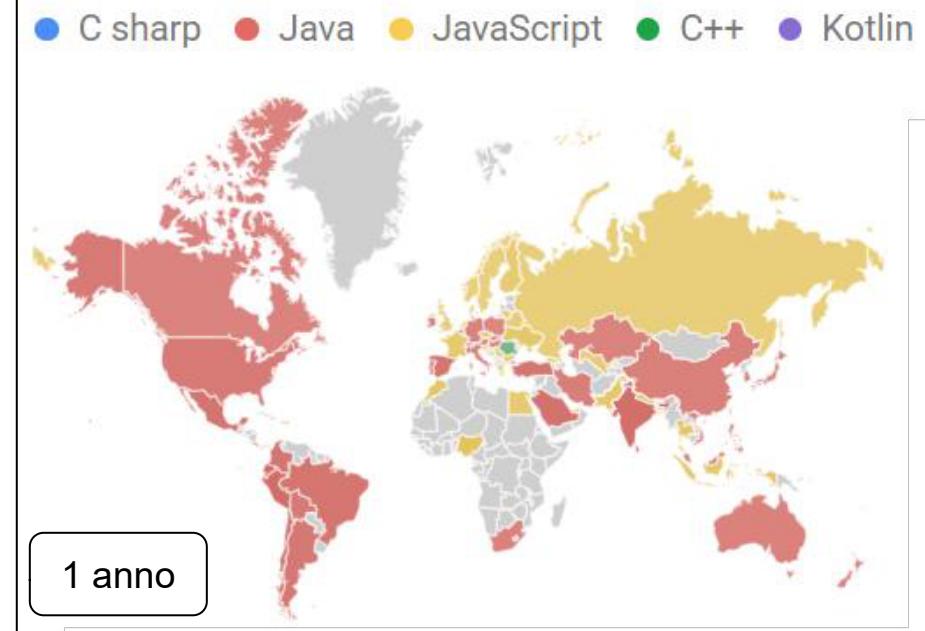
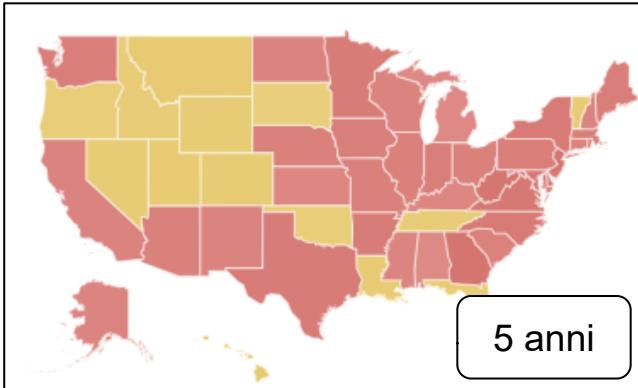
VERSO NUOVI LINGUAGGI





VERSO NUOVI LINGUAGGI

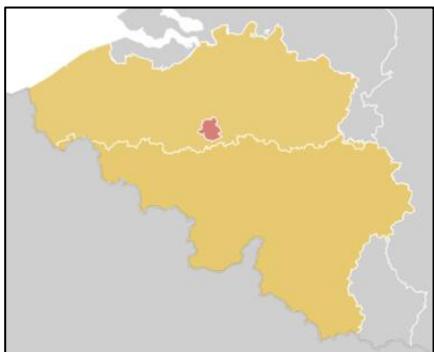
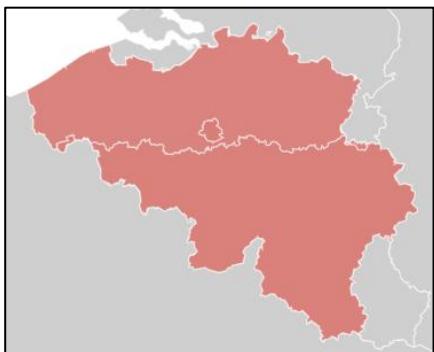
Popolarità dei linguaggi in base
alle ricerche Google
(fonte: *Google Trends*)





VERSO NUOVI LINGUAGGI

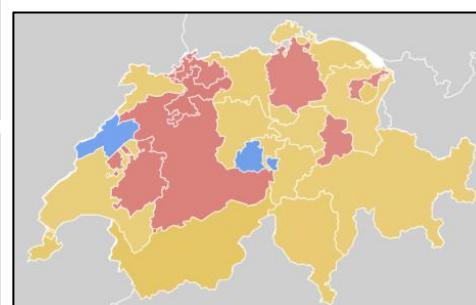
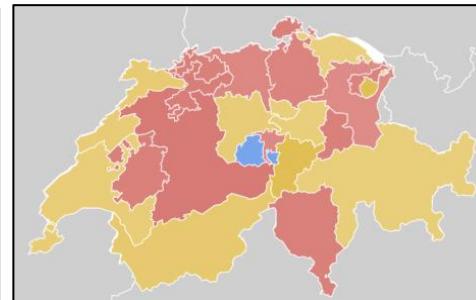
Belgio



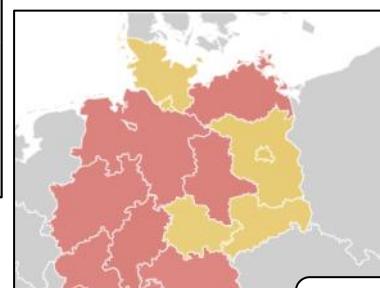
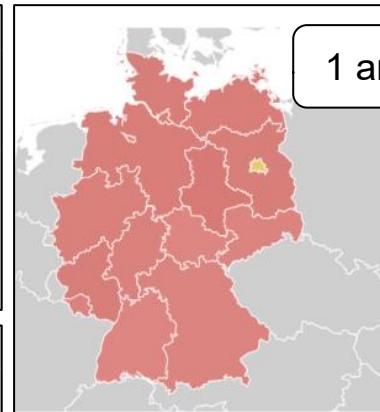
Paesi bassi



Svizzera



Germania



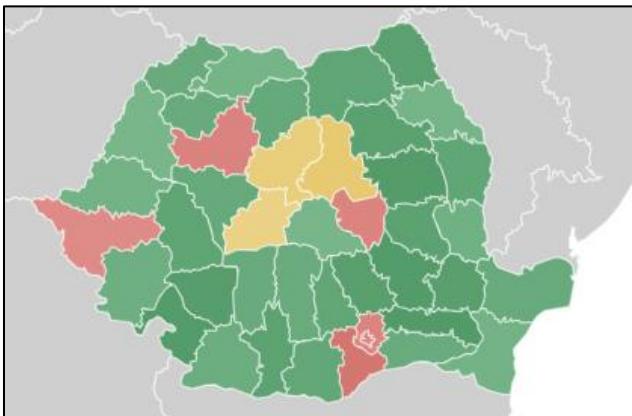
1 anno

5 anni

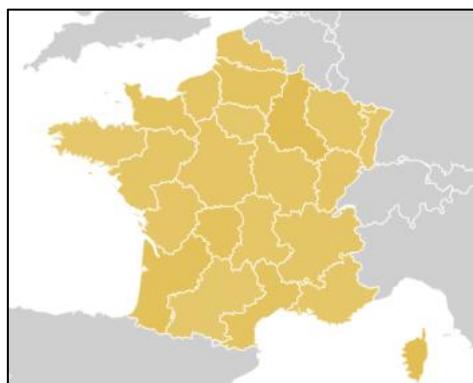


VERSO NUOVI LINGUAGGI

Romania



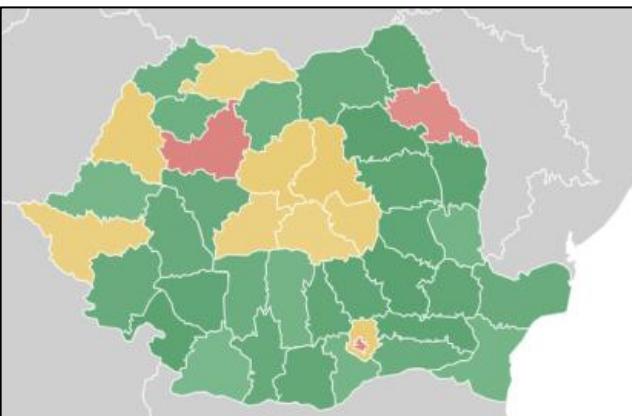
Francia



Russia



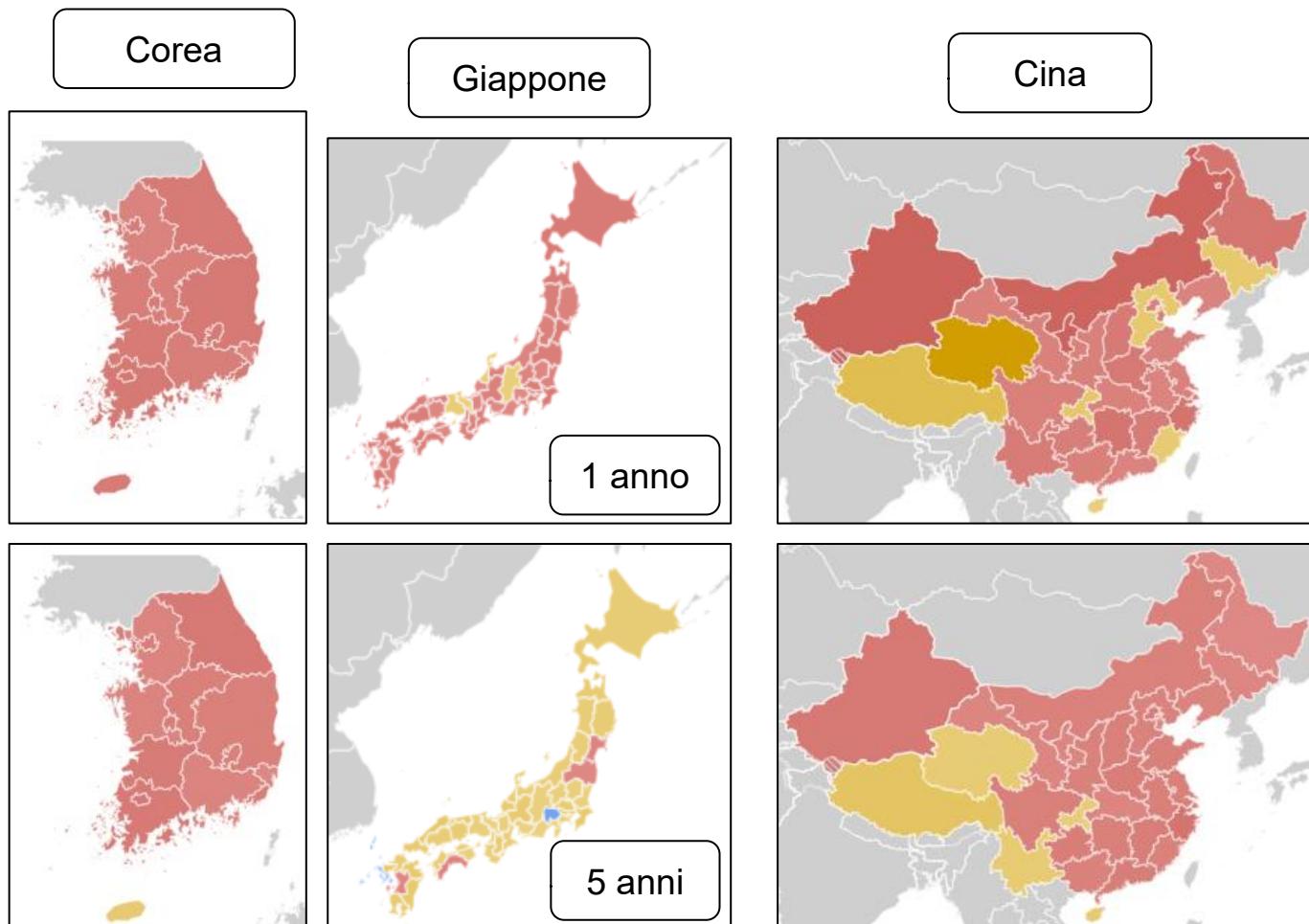
1 anno



5 anni



VERSO NUOVI LINGUAGGI





VERSO NUOVI LINGUAGGI

PYPL PopularitY of Programming Language

The PYPL PopularitY of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

Worldwide, Jan 2026 :

Rank	Change	Language	Share	1-year trend
1		Python	24.61 %	-5.0 %
2	↑↑	C/C++	14.13 %	+6.9 %
3	↑↑↑↑↑↑↑	Objective-C	13.35 %	+10.5 %
4	↓↓	Java	10.45 %	-4.8 %
5	↑	R	6.18 %	+1.6 %
6	↓↓↓	JavaScript	4.68 %	-3.3 %
7	↑↑↑↑↑	Swift	3.68 %	+1.1 %
8	↓	PHP	2.95 %	-0.9 %
9	↓↓↓↓↓	C#	2.79 %	-3.4 %

Worldwide, Jan 2025 :

Rank	Change	Language	Share	1-year trend
1		Python	29.8 %	+1.7 %
2		Java	15.35 %	-0.5 %
3		JavaScript	7.96 %	-1.0 %
4		C/C++	7.17 %	+0.3 %
5		C#	6.36 %	-0.3 %
6		R	4.63 %	+0.0 %
7		PHP	3.87 %	-0.6 %
8		TypeScript	2.84 %	-0.0 %
9	↑↑	Rust	2.8 %	+0.5 %
10		Objective-C	2.65 %	+0.2 %
11	↓↓	Swift	2.48 %	-0.2 %
12		Go	2.11 %	-0.0 %
13		Kotlin	1.9 %	+0.1 %



VERSO NUOVI LINGUAGGI

Jan 2017

Jan 2022

Jan 2023

Worldwide, Jan 2017 compared to a year ago:				
Rank	Change	Language	Share	Trend
1		Java	23.1 %	-0.9 %
2		Python	14.4 %	+2.7 %
3		PHP	9.7 %	-1.0 %
4		C#	8.4 %	-0.1 %
5	↑↑	Javascript	7.7 %	+0.5 %
6		C	7.1 %	-0.3 %
7	↓↓	C++	7.0 %	-0.6 %
8		Objective-C	4.4 %	-0.6 %
9	↑	R	3.4 %	+0.5 %
10	↓	Swift	3.0 %	+0.1 %
11		Matlab	2.6 %	-0.3 %
12		Ruby	2.1 %	-0.1 %
13		Visual Basic	1.5 %	-0.2 %
14		VBA	1.5 %	+0.0 %
15	↑	Scala	1.2 %	+0.4 %
16	↓	Perl	0.9 %	-0.1 %
17		lua	0.5 %	+0.0 %

Worldwide, Jan 2022 compared to a year ago:				
Rank	Change	Language	Share	Trend
1		Python	28.74 %	-1.8 %
2		Java	18.01 %	+1.2 %
3		JavaScript	9.07 %	+0.6 %
4	↑	C/C++	7.4 %	+1.1 %
5	↓	C#	7.27 %	+0.7 %
6		PHP	6.06 %	+0.0 %
7		R	4.19 %	+0.3 %
8		Objective-C	2.27 %	-1.4 %
9		Swift	1.91 %	-0.2 %
10		TypeScript	1.74 %	-0.0 %
11		Matlab	1.74 %	+0.0 %
12		Kotlin	1.71 %	+0.0 %
13		Go	1.19 %	-0.1 %
14		VBA	1.14 %	-0.0 %
15		Ruby	1.07 %	-0.0 %
16		Rust	0.98 %	-0.0 %
17	↑↑↑	Ada	0.75 %	+0.1 %

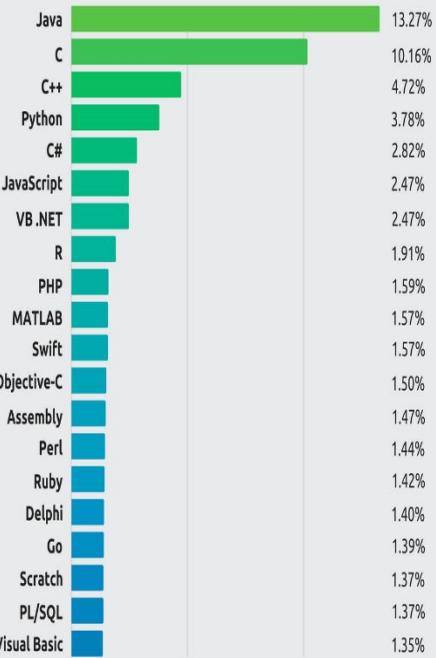
Worldwide, Jan 2023 compared to a year ago:				
Rank	Change	Language	Share	Trend
1		Python	27.93 %	-0.9 %
2		Java	16.78 %	-1.3 %
3		JavaScript	9.63 %	+0.5 %
4	↑	C#	6.99 %	-0.3 %
5	↓	C/C++	6.9 %	-0.5 %
6		PHP	5.29 %	-0.8 %
7		R	4.03 %	-0.2 %
8	↑↑↑	TypeScript	2.79 %	+1.0 %
9		Swift	2.23 %	+0.3 %
10	↓↓	Objective-C	2.2 %	-0.1 %
11	↑↑	Go	1.94 %	+0.7 %
12	↑↑↑	Rust	1.9 %	+0.9 %
13	↓	Kotlin	1.81 %	+0.1 %
14	↓↓↓	Matlab	1.63 %	-0.1 %
15	↑	Ruby	1.13 %	+0.3 %
16	↓↓	VBA	1.03 %	-0.0 %



VERSO NUOVI LINGUAGGI – 2025

Top Programming Languages

Tiobe Index - December 2017



Jan 2024	Jan 2023	Change	Programming Language
1	1		Python
2	2		C
3	3		C++
4	4		Java
5	5		C#
6	7	▲	JavaScript
7	10	▲	PHP
8	6	▼	Visual Basic
9	8	▼	SQL
10	20	▲	Scratch
11	12	▲	Go
12	27	▲	Fortran
13	17	▲	Delphi/Object Pascal
14	15	▲	MATLAB
15	9	▼	Assembly language
16	11	▼	Swift
17	25	▲	Kotlin

Jan 2025	Jan 2024	Change	Programming Language
1	1		Python
2	3	▲	C++
3	4	▲	Java
4	2	▼	C
5	5		C#
6	6		JavaScript
7	11	▲	Go
8	9	▲	SQL
9	8	▼	Visual Basic
10	12	▲	Fortran
11	13	▲	Delphi/Object Pascal
12	10	▼	Scratch
13	7	▼	PHP
14	19	▲	Rust

January Headline: C# is programming language of the year 2023!

January Headline: Python is TIOBE's programming language of the year 2024!



VERSO NUOVI LINGUAGGI – 2025



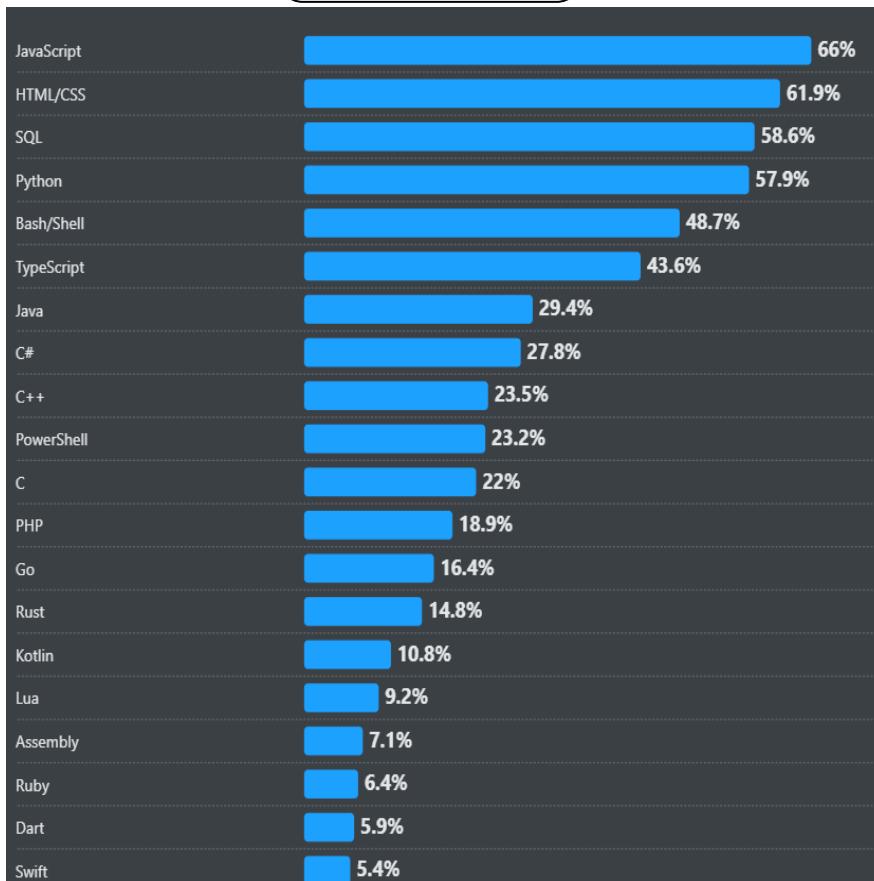
Very Long Term History

Programming Language	2025	2020	2015	2010	2005	2000	1995	1990	1985
Python	1	3	7	7	7	25	23	-	-
C++	2	4	4	3	3	2	1	2	13
C	3	2	1	2	1	1	2	1	1
Java	4	1	2	1	2	3	-	-	-
C#	5	5	5	6	9	9	-	-	-
JavaScript	6	7	8	9	10	7	-	-	-
Go	7	17	37	184	-	-	-	-	-
Visual Basic	8	19	234	-	-	-	-	-	-
SQL	9	9	-	-	100	-	-	-	-
Fortran	10	29	31	25	15	18	5	9	11
Ada	25	35	30	27	16	17	7	4	3
Lisp	29	31	18	17	13	16	6	3	2
Objective-C	35	10	3	24	40	-	-	-	-
(Visual) Basic	-	-	54	5	4	4	3	5	4



VERSO NUOVI LINGUAGGI – 2025

StackOverflow
Survey





VERSO NUOVI LINGUAGGI – 2025

According to [Indeed in 2025](#), Java is the third most in-demand programming language, after Python and JavaScript [with a median salary of \\$116,000 in](#) the United States. Java developer salary has increased by [11%](#) over the last five years.

Worldwide Popularity of Programming Languages, 2024

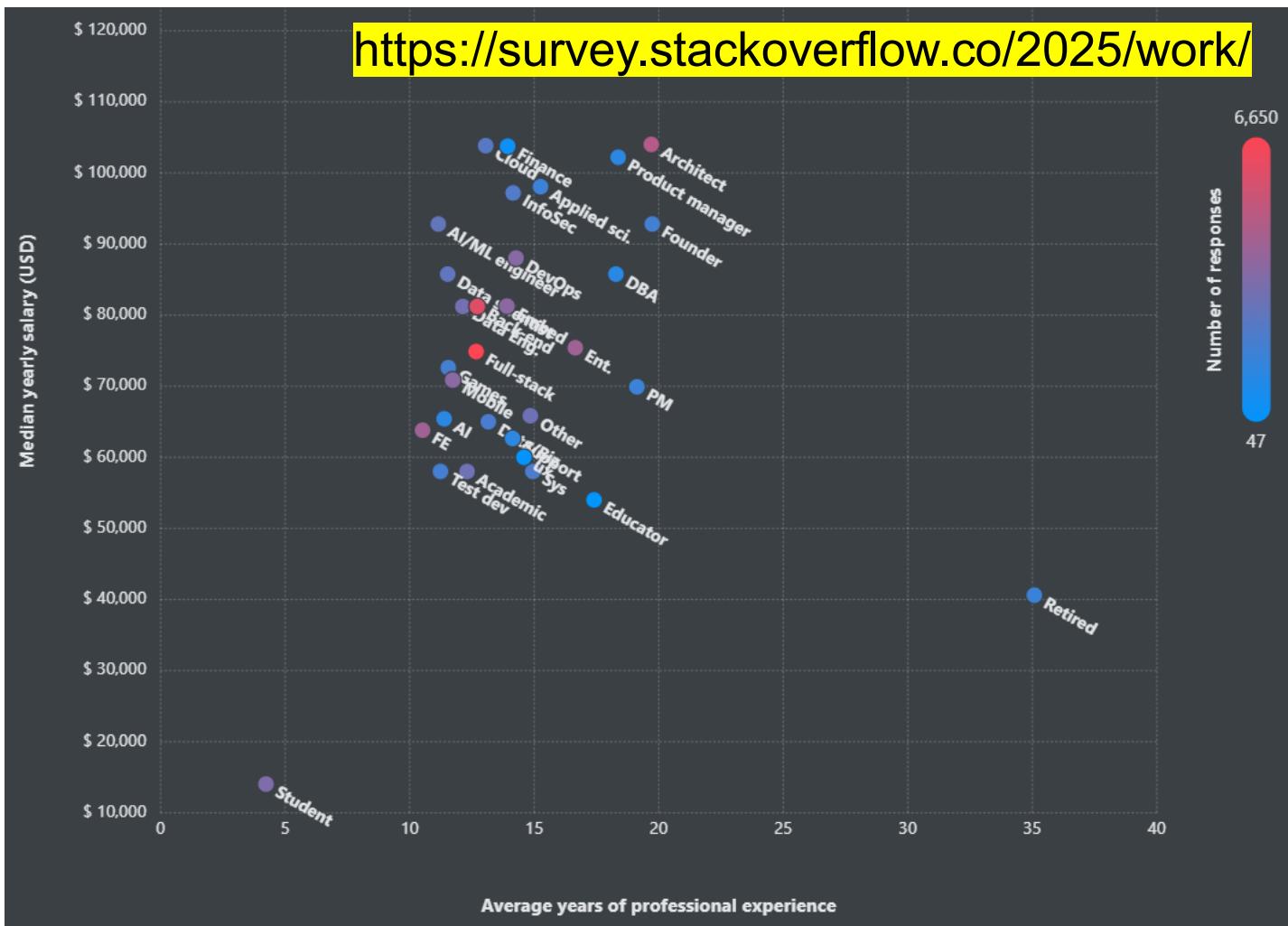
Rank	Language	Share	1-year trend
1	Python	29.71 %	+1.5 %
2	Java	15.43 %	-0.3 %
3	JavaScript	7.99 %	-0.9 %
4	C/C++	7.06 %	+0.3 %
5	C#	6.42 %	-0.2 %
6	R	4.67 %	+0.1 %

Worldwide, Jan 2026 :

Rank	Change	IDE	Share	1-year trend
1		Visual Studio	27.75 %	-0.4 %
2		Visual Studio Code	15.06 %	-0.1 %
3	↑↑	Eclipse	12.74 %	+2.6 %
4		Android Studio	10.67 %	+0.1 %
5	↓↓	pyCharm	8.05 %	-2.5 %
6		IntelliJ	6.31 %	-1.1 %
7	↑	Xcode	3.55 %	+0.5 %
8	↓	NetBeans	3.42 %	-0.4 %

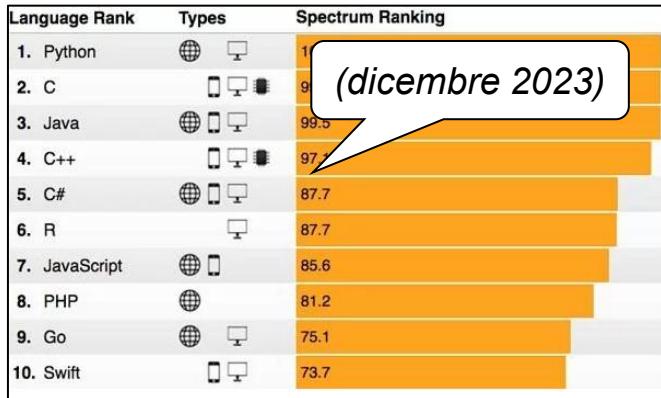


VERSO NUOVI LINGUAGGI – 2025

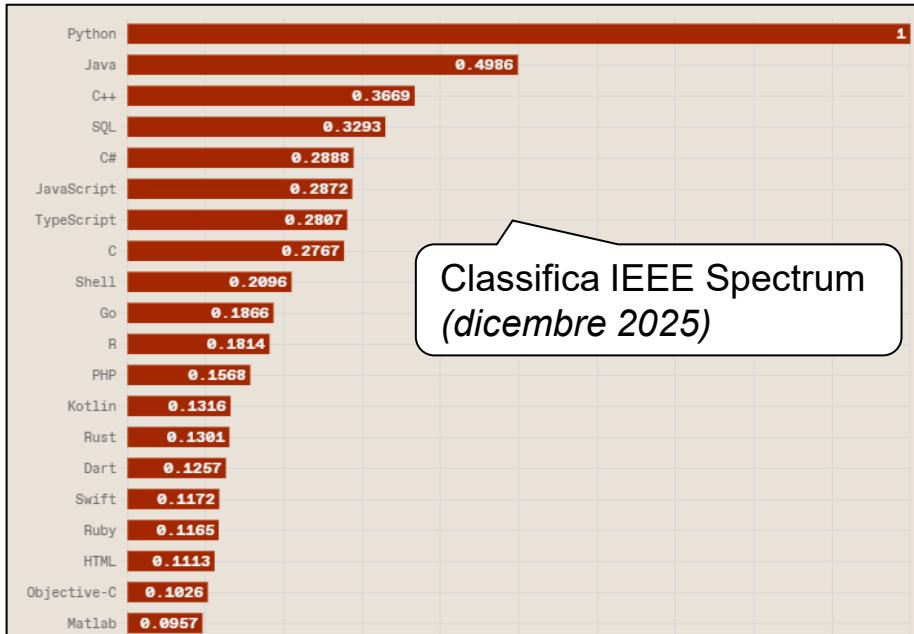




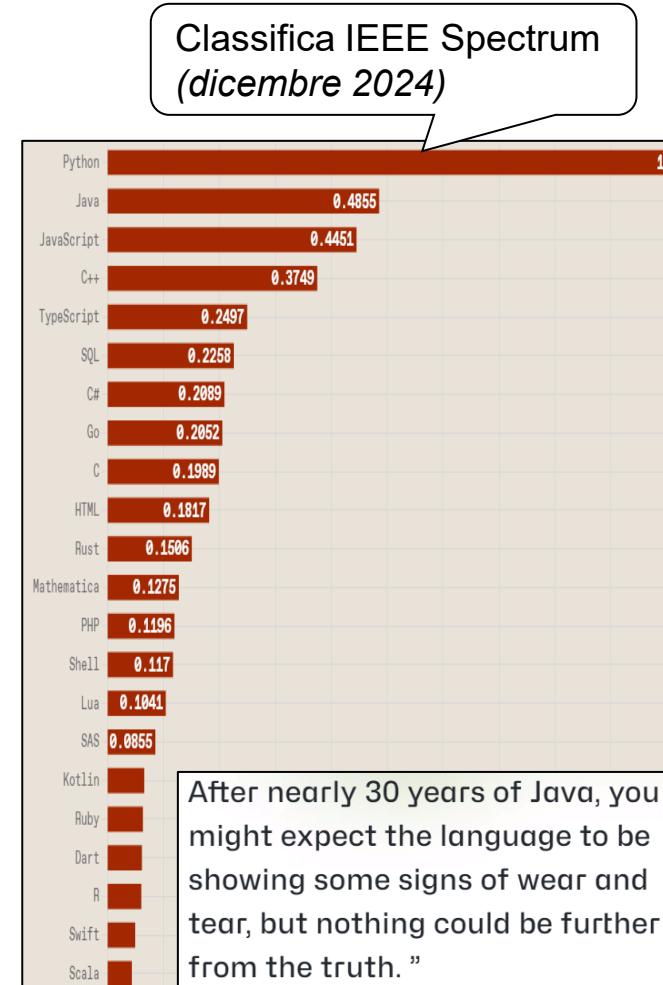
VERSO NUOVI LINGUAGGI – 2025



(dicembre 2023)

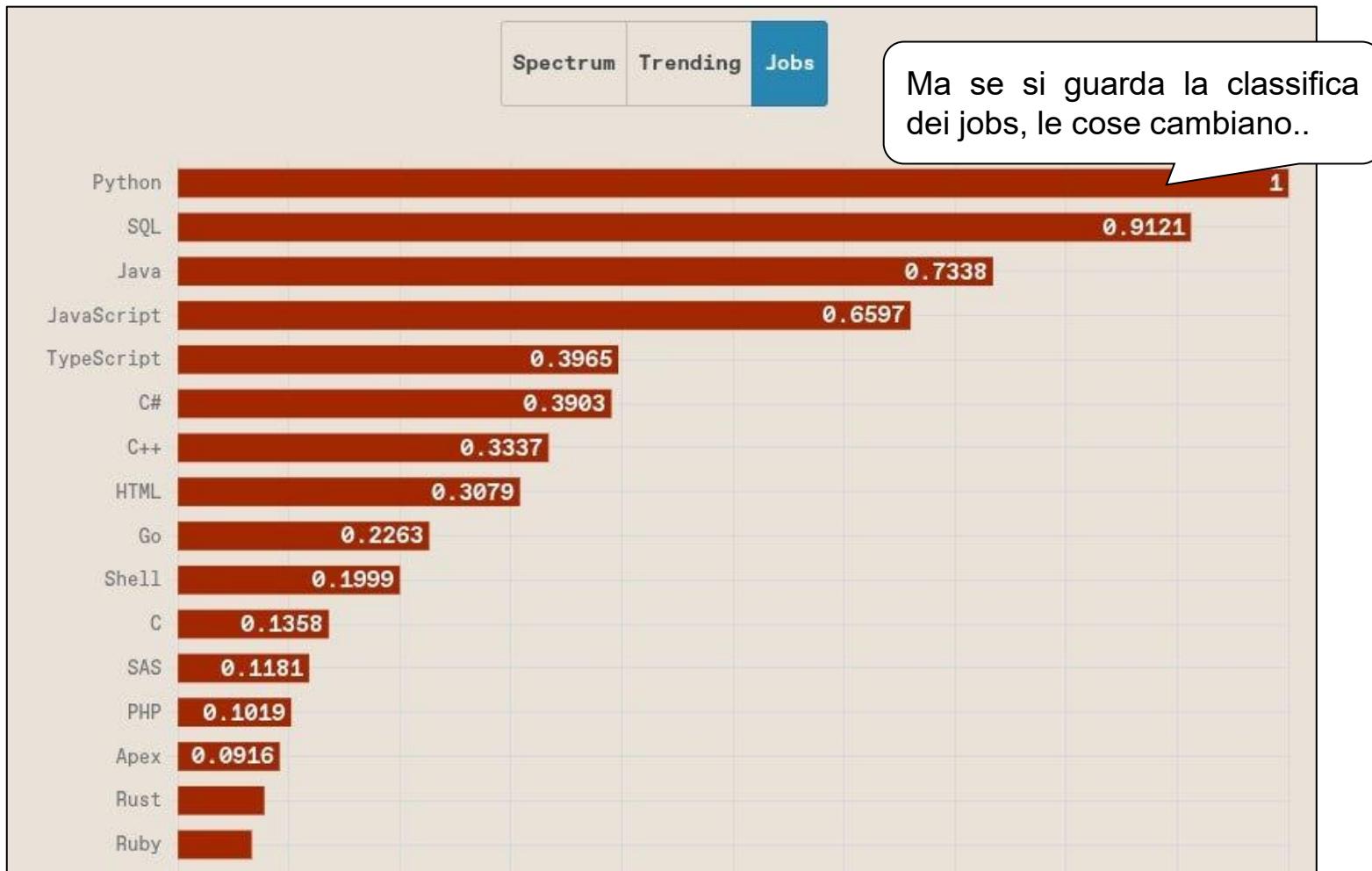


Classifica IEEE Spectrum
(dicembre 2025)





VERSO NUOVI LINGUAGGI – 2025





VERSO NUOVI LINGUAGGI – 2024

MA le classifiche vanno interpretate...

You can find a full breakdown of [our methodology here](#), but let's jump into our results. [At the top, Python continues to cement its overall dominance](#), buoyed by things like popular libraries for hot fields such as AI as well as its [pedagogical prominence](#). (For most students today, if they learn one programming language in school, it's [Python](#).) Python's pretty popular with employers too, although there its lead over other general purpose languages is not as large and, like last year, it plays second fiddle to the database query language [SQL](#), which employers like to see paired with another language. SQL popularity with employers is a natural extension of today's emphasis on networked and cloud-based system architectures, where databases become the natural repository for all the bytes a program's logic is chewing on.

[Stalwarts like Java, Javascript, and C++](#) also retain high rankings, but it's what's going on a little further down that's particularly interesting. [Typescript](#)—a superset of Javascript—moves up several places on all the rankings, especially for Jobs, where it climbs to fourth place, versus 11th last year. Typescript's

Another climber is [Rust](#), a language aimed at creating system software, like C or C++. But unlike those two languages, Rust is “memory safe”, meaning it uses a variety of techniques to ensure programs can't write to locations in memory that they are not supposed to. Such errors are a major source of security vulnerabilities. Rust's profile has been rising sharply, boosted by things like a February [cybersecurity report from the White House](#) calling for memory safe languages to replace C and C++. Indeed, [C's popularity appears to be on the wane](#), falling from fourth to ninth place on the Spectrum ranking and from 7th to 13th on the Jobs ranking.



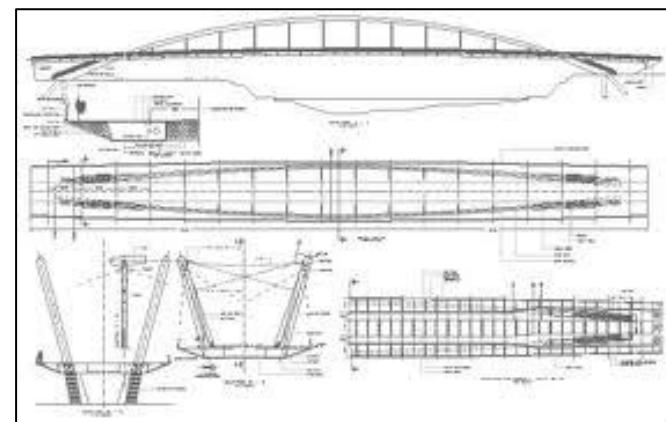
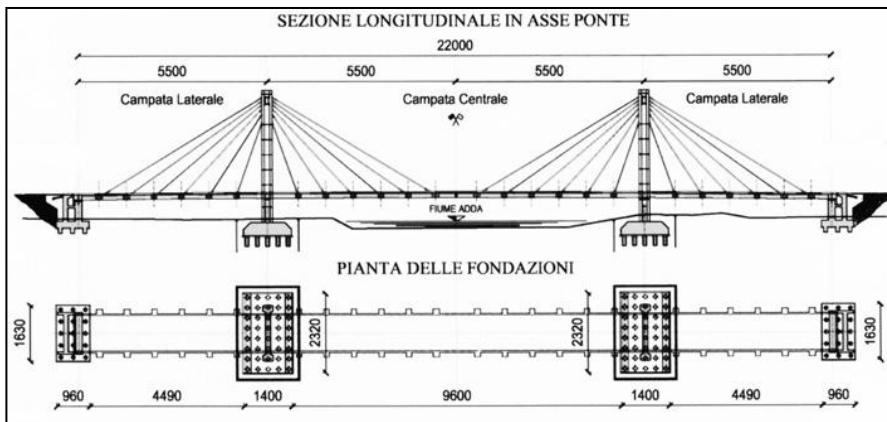
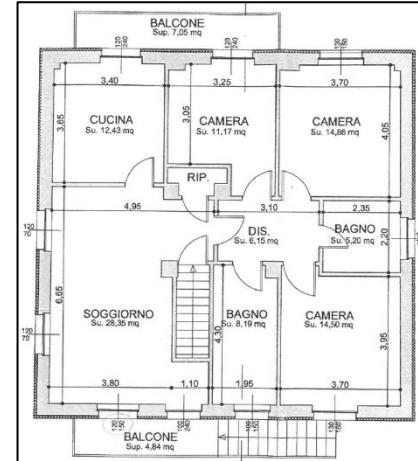
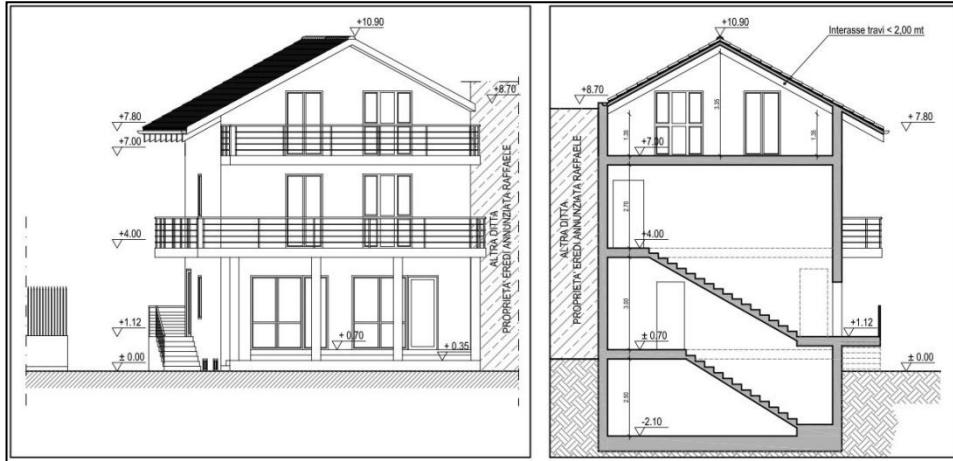
IL PROGETTO NON È IL CODICE

L'attività di progetto dev'essere riconosciuta come tale

- Il suo risultato – il *progetto* – dev'essere *un artefatto esistente e documentato in quanto tale*, come avviene da sempre in tutte le branche dell'ingegneria
 - il progetto di un sistema software non è il codice
 - esattamente come il *progetto di una casa* non è *la casa*, il *progetto di un ponte* non è *il ponte stesso!*
- Solitamente, nell'ingegneria il progetto è espresso tramite *disegni, tavole, diagrammi* che mirano a trasmettere:
 - la struttura d'insieme
 - le parti coinvolte e le loro relazioni



IL PROGETTO NON È IL CODICE





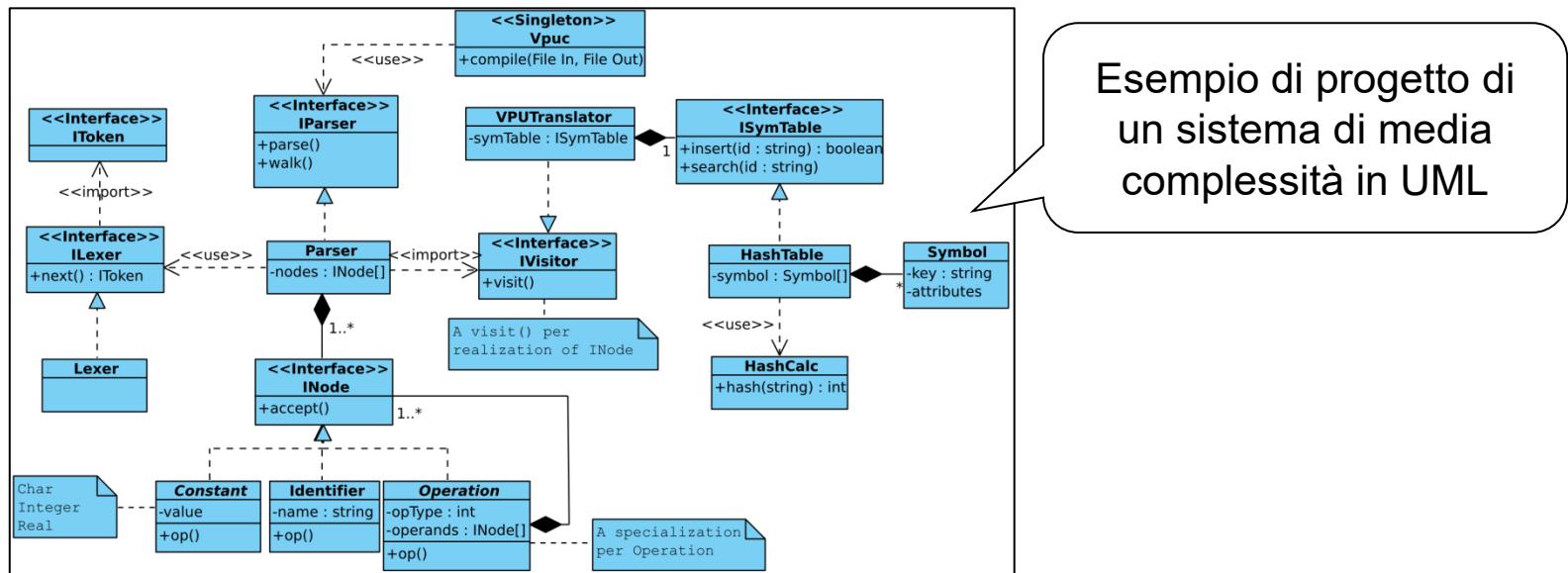
IL PROGETTO NON È IL CODICE

L'attività di progetto dev'essere riconosciuta come tale *anche nell'ingegneria informatica*

- L'attività di progetto dev'essere precedente e totalmente disaccoppiata dall'implementazione
- La relativa "giovinezza" dell'ingegneria informatica ha portato molti "artigiani del fai da te" e pensare erroneamente di poter far senza questi strumenti..
- ..ma non si gestisce la complessità senza poter operare a *un adeguato livello di astrazione – che non è il codice*
- *Servono metodologie, standard e linguaggi grafici per esprimere il progetto di un sistema software*

ESEMPIO: PROGETTO IN UML

- UML (Unified Modelling Language) è uno standard internazionale per esprimere modelli di sistemi (non necessariamente informatici)
- Fra i molti diagrammi che UML introduce, il *diagramma di struttura* esprime la *struttura di un sistema a oggetti*

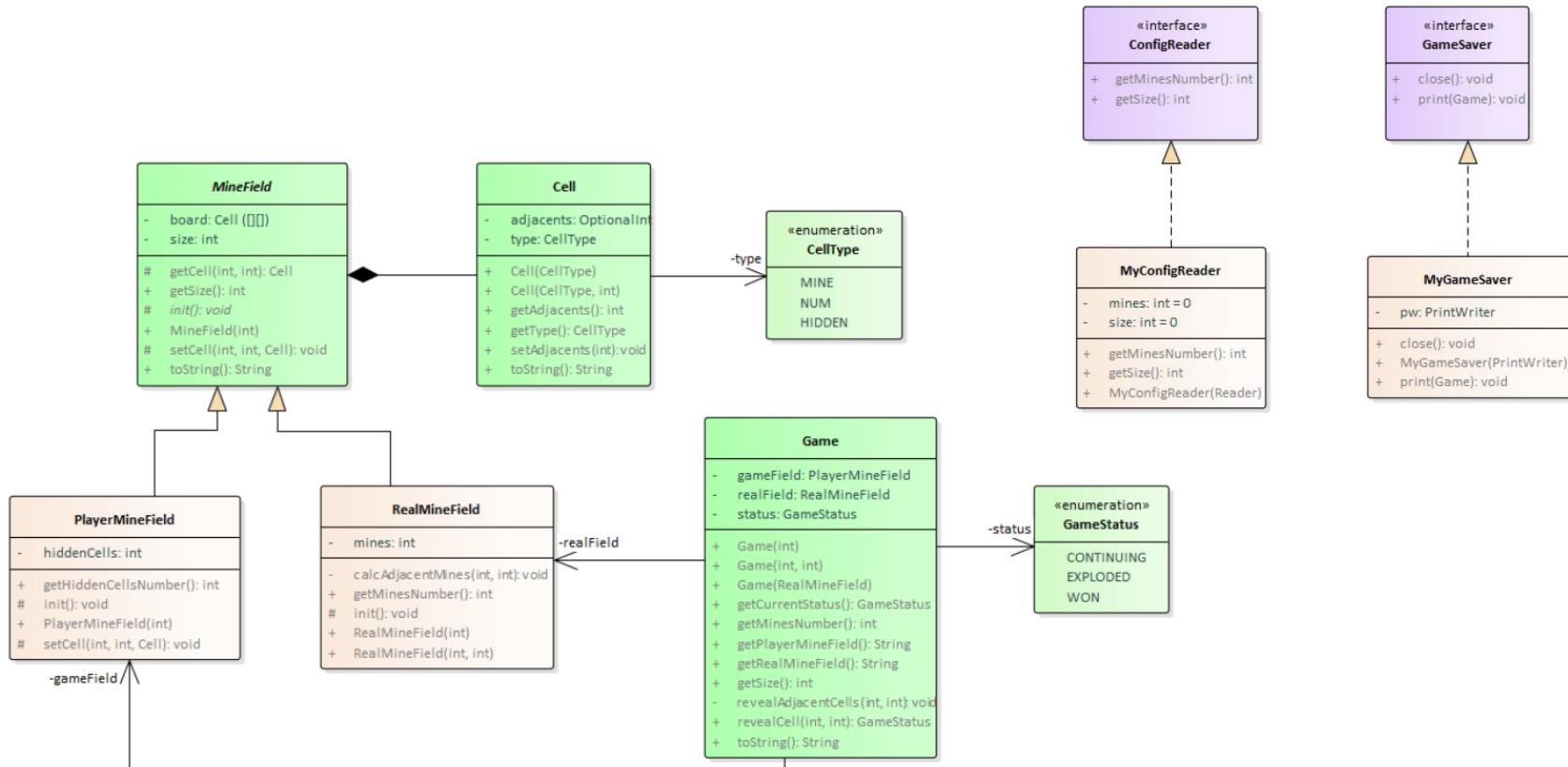


Esempio di progetto di un sistema di media complessità in UML



ESEMPIO: PROGETTO IN UML

- ESEMPIO: parte di un diagramma da un compito d'esame

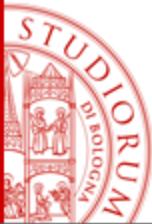




SOLO NUOVI LINGUAGGI..?

MA un buon linguaggio non basta..

- Per avere successo occorre *non lasciarlo solo*
- Intorno a esso deve essere previsto un *ecosistema* pensato secondo una *precisa visione infrastrutturale*
 - l'esperienza (C++..) insegna che poter creare propri componenti e oggetti è importante, ma *standardizzare il linguaggio non basta*
- Non si può lasciare che ogni azienda (o strumento) realizzi "a proprio gusto" *tutto quello che ci sta intorno*
 - se ognuno realizza le proprie librerie (circa simili, ma non identiche) si creano *entità e comunità incomunicabili*
 - se ognuno copre solo alcune piattaforme, si finisce per *rifare le stesse cose N volte, con spreco di tempo e denaro*



DAI LINGUAGGI ALLE INFRASTRUTTURE

Occorre una *infrastruttura standard*

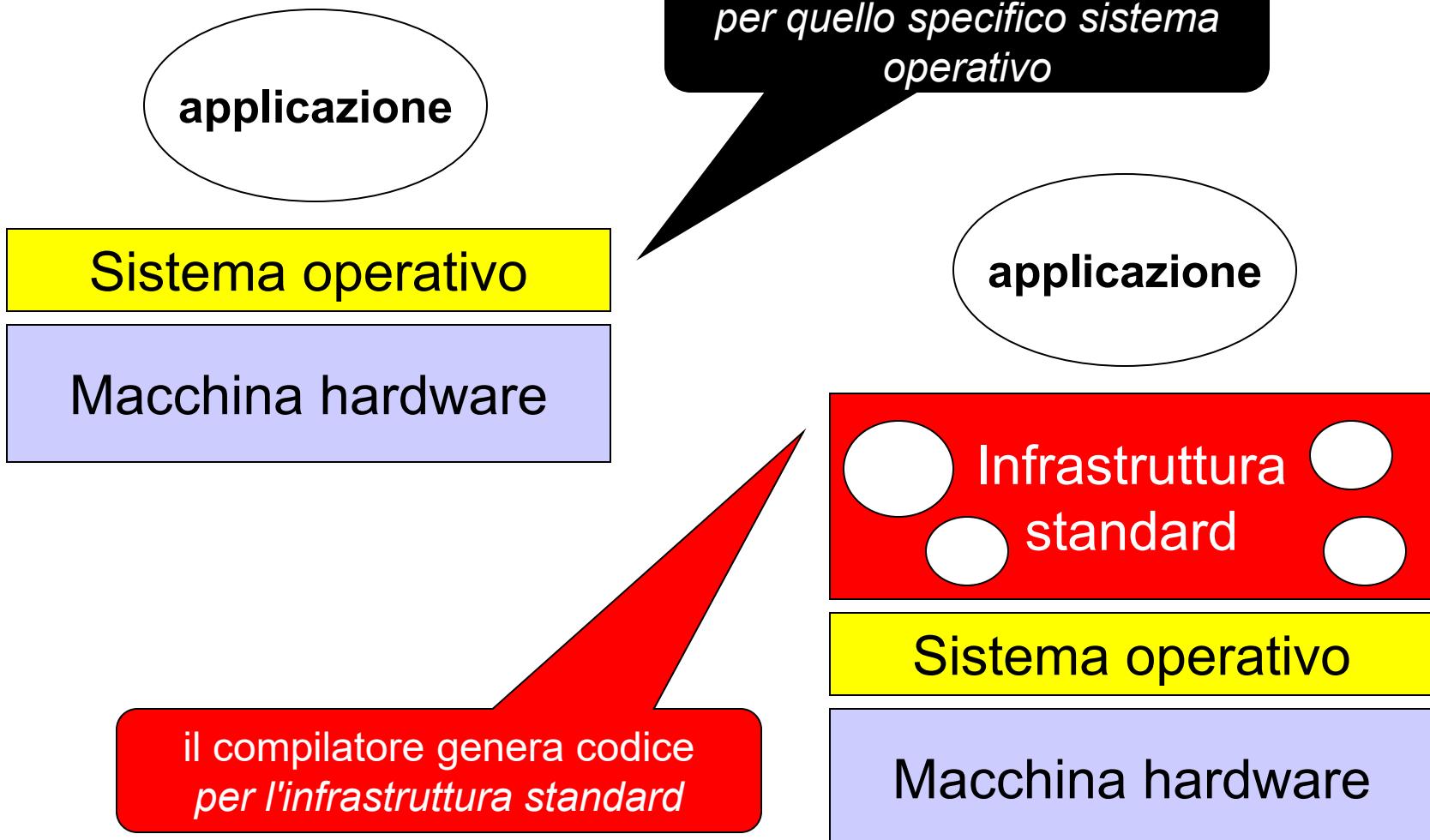
- un ambiente che offre *componenti pronti standardizzati, certamente presenti su ogni installazione*
- uno *strato di sicura affidabilità* su cui contare e costruire
- disponibile per *più sistemi operativi* e *diversi hardware*
- uno strato che *renda indipendente chi sta sopra dalla specifica configurazione hardware/software sottostante*

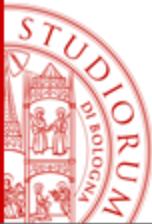
Se ci si riesce, *cambia totalmente lo scenario*

- cambia il processo di compilazione ed esecuzione
- *cambia il modo stesso di vedere "il software"*
- il sogno di ogni softwarista: *write once, run everywhere*



DAI LINGUAGGI ALLE INFRASTRUTTURE



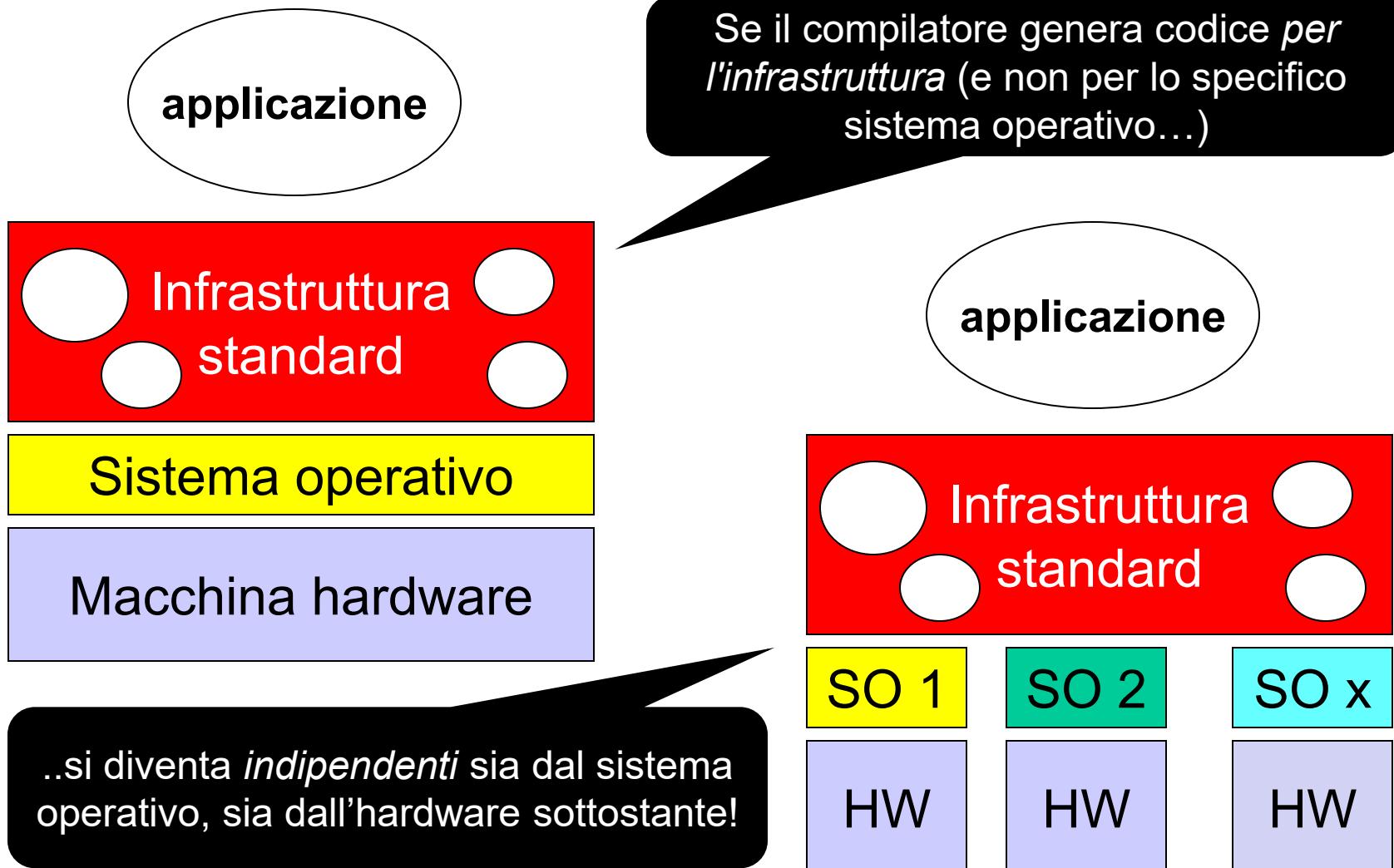


DAI LINGUAGGI ALLE INFRASTRUTTURE

- Poiché costruire software richiede tempo e risorse, si cerca di *massimizzare la resa* di tale investimento
- Perciò il software dovrebbe essere il più possibile *portabile e inter-piattaforma*
- Per questo interessano ***INFRASTRUTTURE PORTABILI*** e ***MULTI-PIATTAFORMA***
 - installabili su diversi sistemi hardware
 - con diversi sistemi operativi e configurazioni software
 - in grado di offrire alle applicazioni costruite sopra di esse un *ambiente noto, ricco e dal funzionamento uniforme*



INFRASTRUTTURE MULTI-PIATTAFORMA





COMPILAZIONE ED ESECUZIONE

applicazione

PRIMA il compilatore generava codice oggetto, lo *collegava* (*link*) **STATICAMENTE** alle librerie e produceva un *eseguibile AUTOCONTENUTO*

Sistema operativo

Macchina hardware

applicazione

ORA il compilatore genera codice *nel formato intermedio dell'infrastruttura* e non collega librerie staticamente:
si caricano e collegano DINAMICAMENTE, a run-time, i componenti che servono

Infrastruttura
standard

SO 1

SO 2

SO x

HW

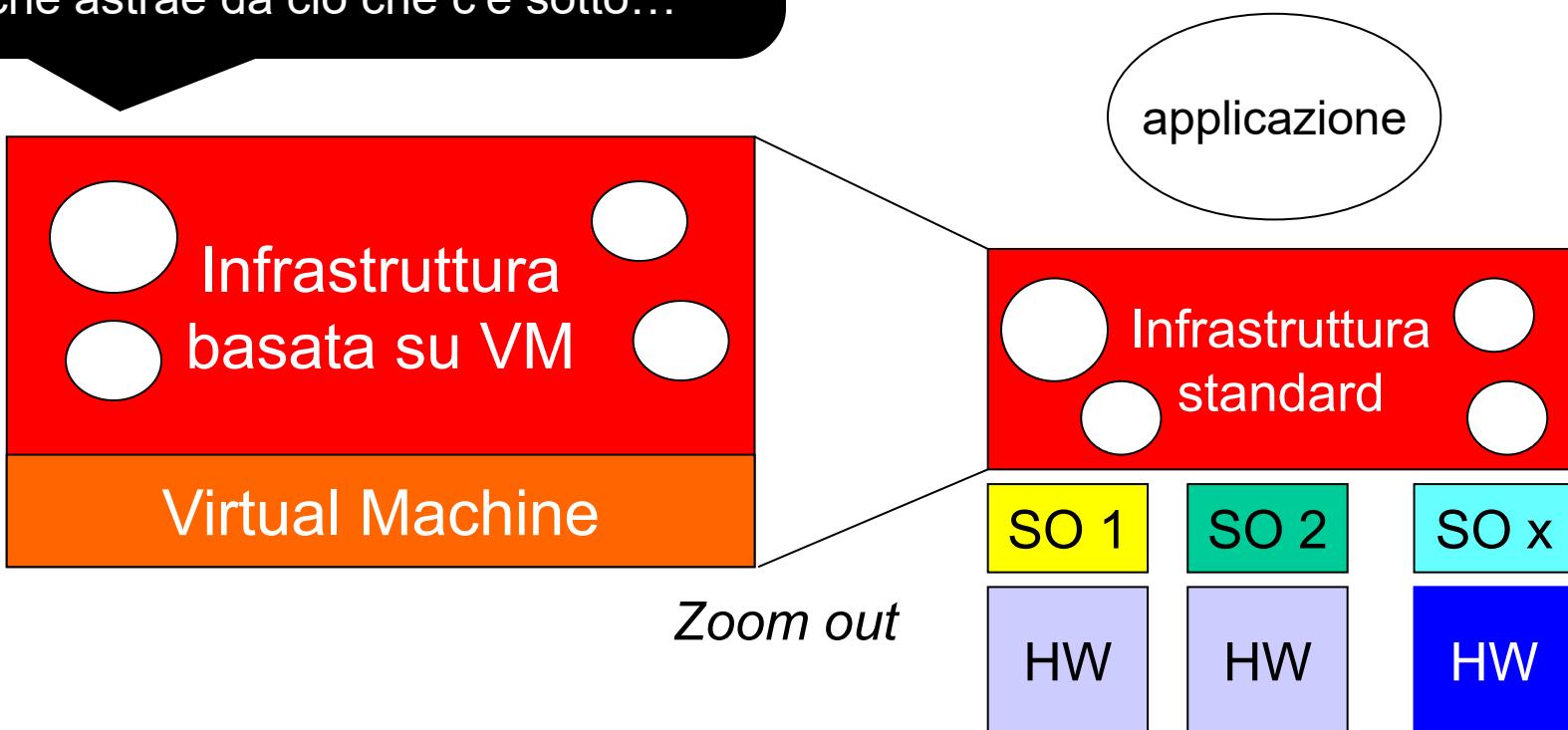
HW

HW



INFRASTRUTTURE MULTIPLE MULTI-PIATTAFORMA

Se poi l'infrastruttura a sua volta si appoggia su uno strato-base di MACCHINA VIRTUALE, che astrae da ciò che c'è sotto...



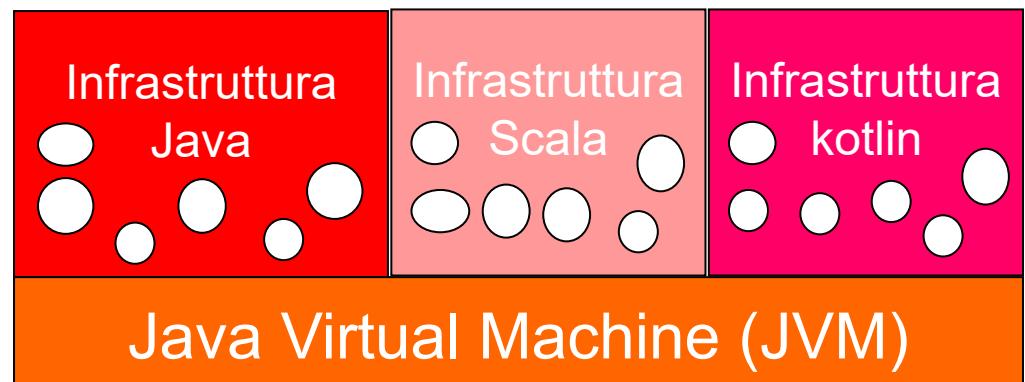


INFRASTRUTTURE MULTIPLE MULTI-PIATTAFORMA

Se poi l'infrastruttura a sua volta si appoggia su uno strato-base di MACCHINA VIRTUALE, che astrae da ciò che c'è sotto...



...si possono avere anche *DIVERSE INFRASTRUTTURE* sulla stessa VM, in grado di *interagire fra loro!*



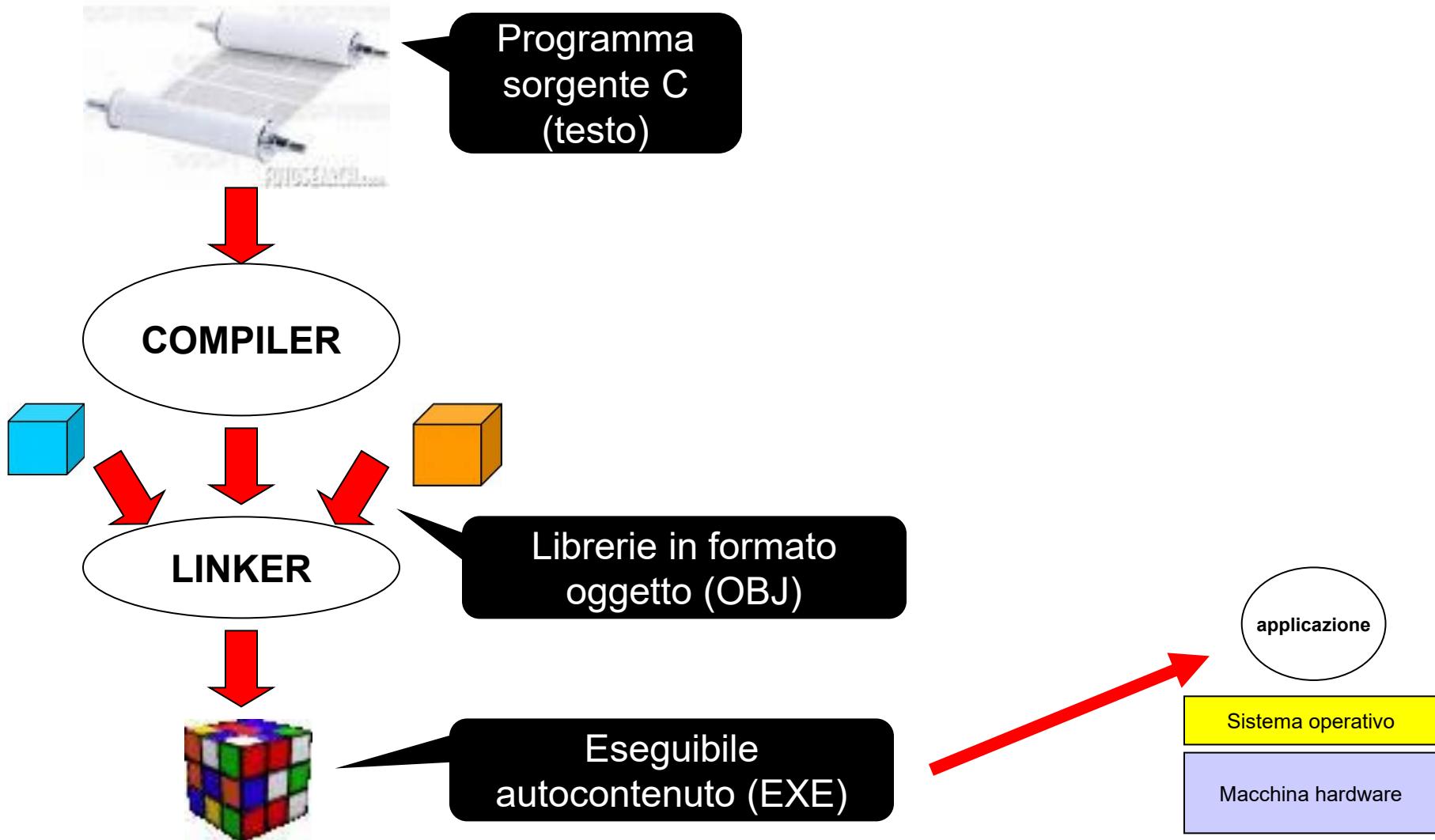


RIASSUMENDO

- Cambia il *processo di costruzione del software*
 - l'obiettivo non è più generare un programma per un certo processore e versione di sistema operativo..
 - ..ma generare qualcosa di *massimamente riusabile*
- Cambia il *processo di esecuzione del software*
 - non si esegue più un eseguibile (EXE) prodotto apposta per un certo processore e sistema operativo..
 - ..ma si fa eseguire all'*infrastruttura* un "prodotto" che non sarebbe eseguibile dal sistema operativo stand alone, poiché non è "autocontenuto": *un eseguibile portatile, inter-piattaforma*
- Si paga un (piccolo) prezzo per questo passo extra, ma i *vantaggi sono molto superiori al costo*
 - processo di sviluppo adatto a un *mondo distribuito e in divenire*

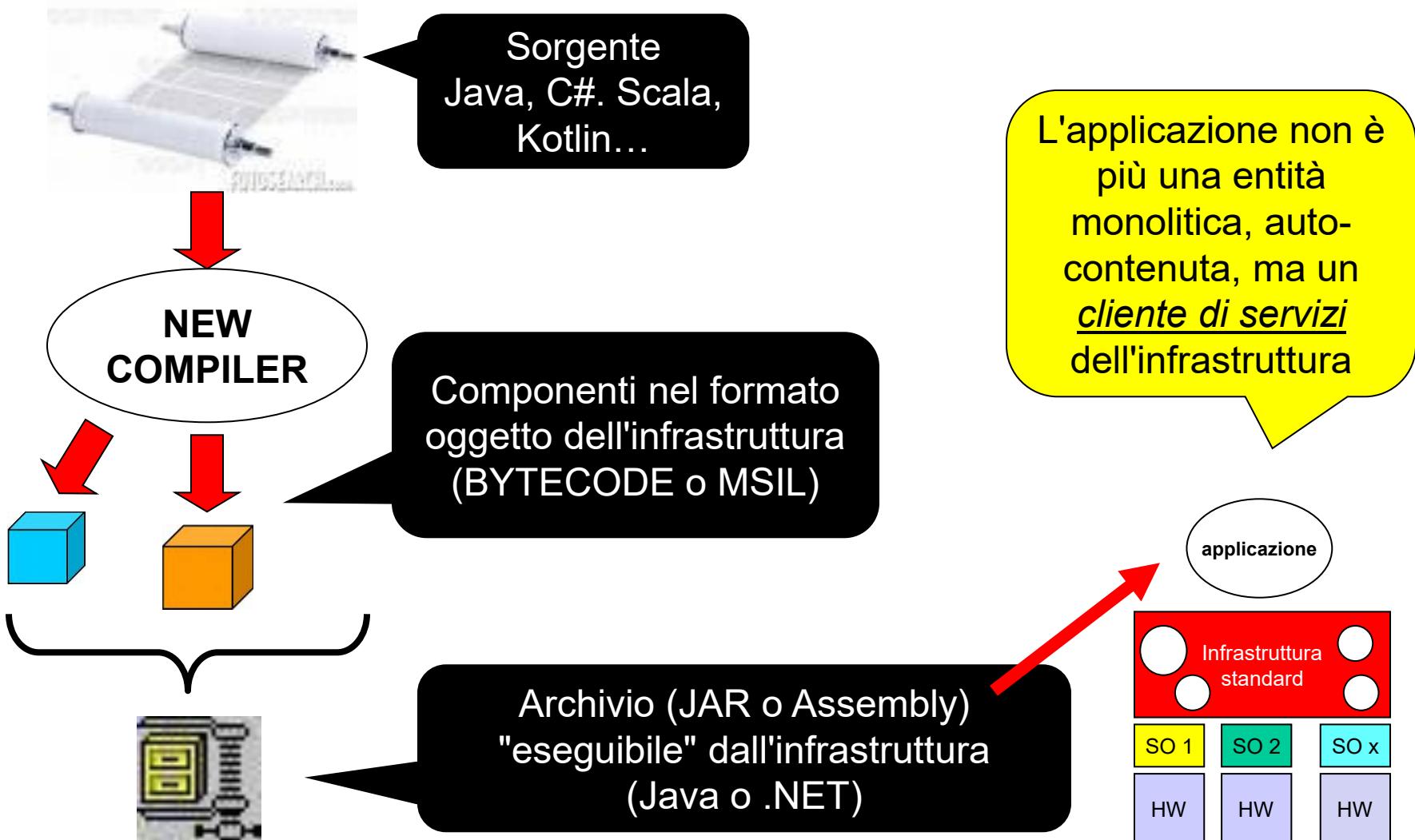


DAL "MONDO C" ...

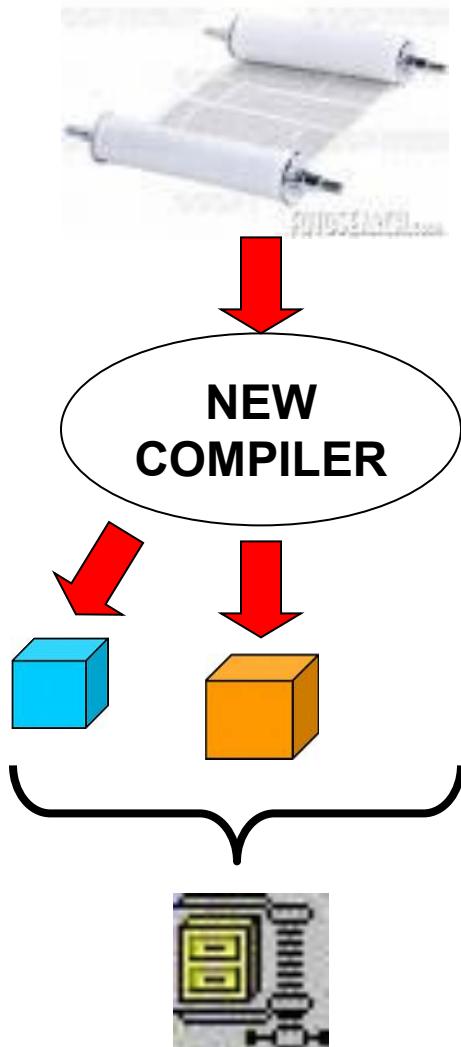




... AL "NUOVO MONDO"



IL MONDO DI JAVA e C#

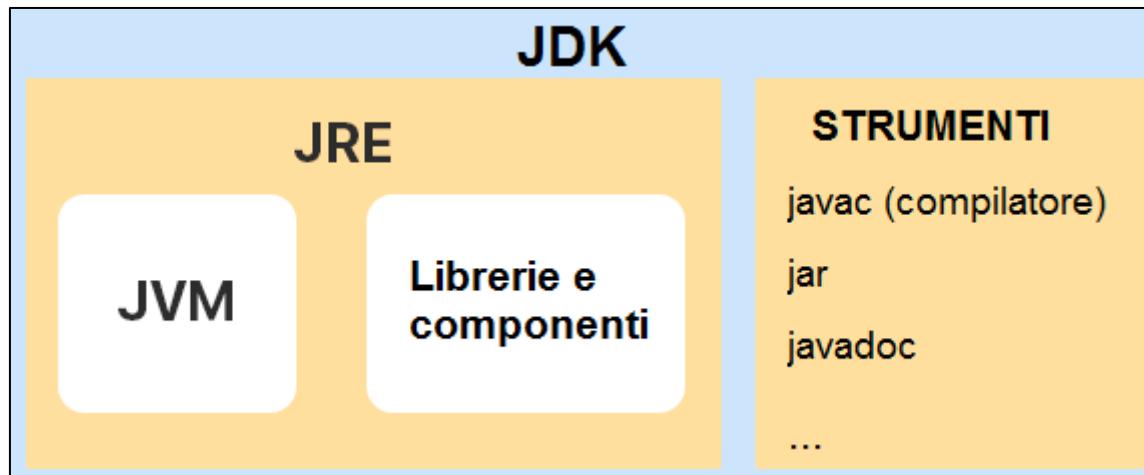


- Il sorgente è strutturato in una serie di componenti, i più importanti dei quali si chiamano **classi**
 - in Scala e kotlin, anche *oggetti* (*singleton*)
- Ogni componente viene compilato in un **formato portatile**, *indipendente da hw e sistema operativo*
- L'eseguibile può essere compilato su una sistema operativo **e funzionare su un altro**, purché la macchina sia *dotata della stessa infrastruttura*
- Per consentire ciò, cambia il senso di "eseguibile"
 - non è più un "eseguibile" in senso classico
 - un EXE di ".NET" o un JAR di Java non funziona su una macchina "nuda", col solo sistema operativo
- Occorre uno **STRATO-PONTE nell'infrastruttura**, che **interpreti** il formato portatile adattandolo alla specifica macchina
 - È l'unico strato *dipendente dalla piattaforma*.



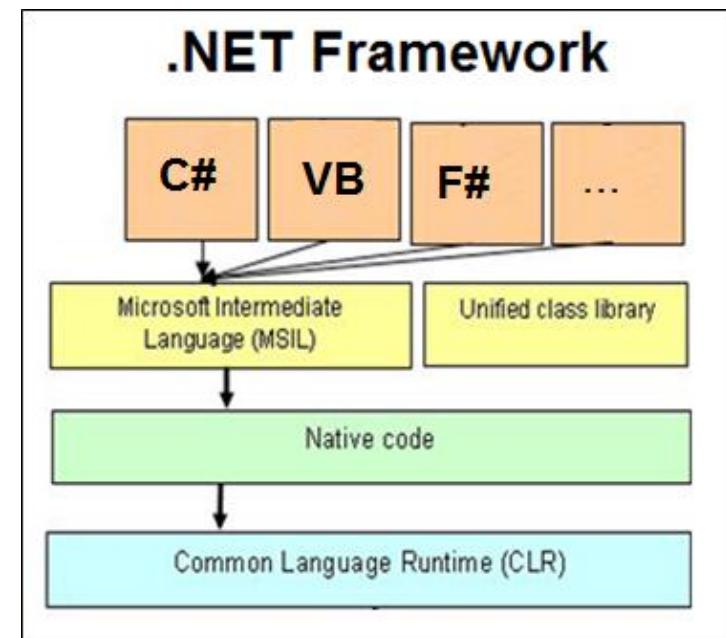
IL RUNTIME NELL'INFRASTRUTTURA

- Lo strato infrastrutturale di **Java** si chiama **JRE** (*Java Runtime Environment*)
 - due livelli: Java Virtual Machine (JVM) + JRE vero e proprio
 - per **eseguire** applicazioni Java dev'essere installato sul computer
 - leggero, gratuito, multi-piattaforma (Windows, Mac, Linux, Android*)
 - per **sviluppare** applicazioni serve un set di strumenti più ampio, che include compilatore e altri strumenti: il **JDK** (*Java Development Kit*)



IL RUNTIME NELL'INFRASTRUTTURA

- Lo strato infrastrutturale di C# si chiama .NET
 - articolazione simile: Common Language Runtime (CLR) + altro
 - pre-installato come .NET framework nei sistemi Windows incluso (*fino alla versione 4.0*) il compilatore a riga di comando (csc)
 - versione multiplataforma anche per MacOS, Linux, Docker
<https://dotnet.microsoft.com>
 - progetto open source, MONO (<https://www.mono-project.com/> ultima release 2019, patch 2024)
 - molteplicità di linguaggi supportati
- Versione attuale .NET 10 (LTS)





LA PIATTAFORMA JAVA

Scaricamento e installazione (dettagli in slide *Strumenti*)

Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC
Mastodon
Bluesky

Bylaws · Census
Legal
Workshop
JEP Process
Source code
GitHub
Mercurial
Tools
Git
jtregr harness
Groups
(overview)
Adoption
Build
Client Libraries
Compatibility &
Specification
Review
Compiler
Conformance
Core Libraries
Governing Board
HotSpot
IDE Tooling & Support
Internationalization
JMX
Members
Networking
Porters
Quality
Security
Serviceability
Vulnerability
Web

OpenJDK



What is this? The place to download the open-source implementation of the Java Platform, Standard Edition, and related projects.



Download and install the latest Oracle's free, GPL-licensed, production-quality JDK 25 binaries for Linux, macOS, and Windows. Available at jdk.java.net/25/; commercial JDK 25 binaries, based on the Oracle JDK, can be found [here](#).



Learn about the key active projects in the OpenJDK, including Amber (high-productivity parallel streams), Loom (lightweight concurrency primitives for parallel execution, functions and foreign data), VarHandle (variable handles), and specialized generics), and Zulu (a fast, reliable, and low-cost Java and the JDK).

If you want to learn how to contribute to the OpenJDK, there are available today, head over to the [Contributing](#) page.

jdk.java.net

GA Releases
JDK 25
JavaFX 25
JMC 9.1.1

Early-Access Releases
JDK 27
JDK 26
JavaFX 26
JavaFX Direct3D 12
Jextract
Leyden
Loom
Valhalla

Reference Implementations
Java SE 25
Java SE 24
Java SE 23
Java SE 22
Java SE 21
Java SE 20
Java SE 19
Java SE 18
Java SE 17
Java SE 16
Java SE 15

Builds

Linux/AArch64	tar.gz	220194971 bytes	(sha256)
Linux/x64	tar.gz	222490107 bytes	(sha256)
macOS/AArch64	tar.gz	215408868 bytes	(sha256)
macOS/x64	tar.gz	217693528 bytes	(sha256)
Windows/x64	zip	221632591 bytes	(sha256)

OpenJDK JDK 25.0.1 General-Availability Release

This page provides production-ready open-source builds of the Java Development Kit, version 25, an implementation of the Java SE 25 Platform under the General Public License, version 2, with the Classpath Exception.

Commercial builds of JDK 25.0.1 from Oracle, under a non-open-source license, can be found [here](#).

Documentation

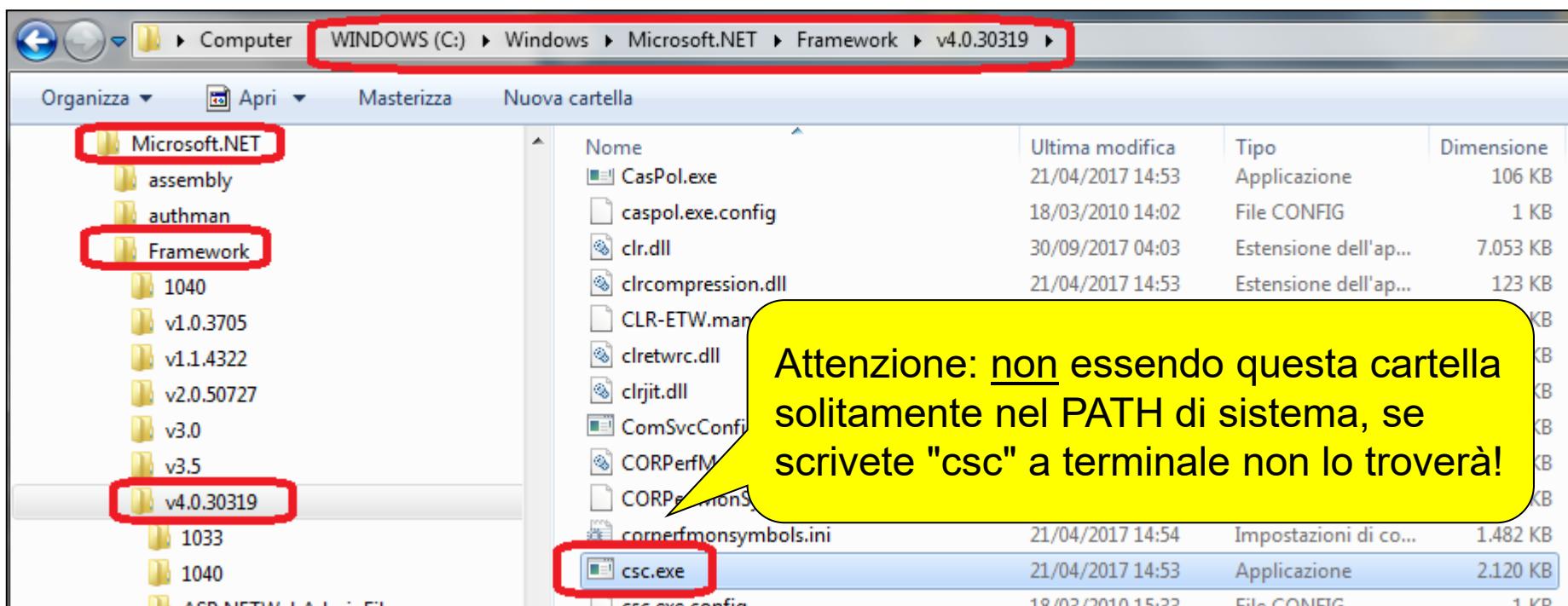
- Features
- Release notes
- API Javadoc

Scegliere la giusta piattaforma



LA PIATTAFORMA .NET

- Il compilatore (vecchio) da riga di comando è nella cartella del .NET Framework (una sottocartella di C:\Windows)
 - potreste averne anche più versioni: 3, 3.5, 4.0, ...
 - es. C:\Windows\Microsoft.NET\Framework\v4.0.30319 [e/o Framework64]





LA PIATTAFORMA .NET

- Per usarlo basta aggiungere la cartella al path corrente:

```
C:\Windows\System32\cmd.exe
C>csc
"csc" non è riconosciuto come comando interno o esterno,
un programma eseguibile o un file batch.

C>path=%PATH%;C:\Windows\Microsoft.NET\Framework\v4.0.30319
C>csc
Microsoft (R) Visual C# Compiler version 4.7.2053.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.

This compiler is provided as part of the Microsoft (R) .NET Framework, but only
supports language versions up to C# 5, which is no longer the latest version. Fo
r compilers that support newer versions of the C# programming language, see http
://go.microsoft.com/fwlink/?LinkId=533240

warning CS2008: Nessun file di origine specificato
error CS1562: Per gli output senza origine occorre specificare l'opzione /out.

C>_
```

Ora lo trova ☺



LA PIATTAFORMA .NET

- Dal 2019 il **.NET Framework** (solo per Windows) è stato sostituito dalla **piattaforma .NET** (ora versione 10)
 - allineato a C# 14.0 e Visual Studio 2025
 - sito: <https://dotnet.microsoft.com>
- Vedi anche progetto open source Mono

Microsoft | .NET | Perché .NET | Funzionalità | Learn | Documenti | Download | Community | TV IN DIRETTA | Tutti i siti Microsoft

Costruisci. Testa. Distribuisci.

.NET è il framework multipiattaforma open source gratuito per la creazione di app moderne e servizi cloud avanzati.

Scarica Attività Iniziali

Supportato in Windows, Linux e macOS

Costruisci con .NET

- Web**: Crea app Web e servizi per macOS, Windows, Linux e Docker.
- Per dispositivi mobili e desktop**: Usa una singola codebase per creare app native per Windows, macOS, iOS e Android.
- Cloud**: Consente di creare app native per il cloud scalabili e resistenti che vengono eseguite in tutti i principali provider di cloud.
- Intelligenza artificiale e Machine Learning**: Crea app intelligenti con C#, OpenAI e Azure.

Microservizi → Sviluppo di giochi → Internet delle cose → Dispositivo mobile → Desktop → Web front-end → API back-end → Dati →

Cross platform, open source .NET framework

Mono is a software platform designed to allow developers to easily create cross platform applications.

Sponsored by Microsoft, Mono is an open source implementation of Microsoft's .NET Framework as part of the .NET Foundation and based on the ECMA standards for C# and the Common Language Runtime. A growing family of solutions and an active and enthusiastic contributing community is helping position Mono to become the leading choice for development of cross platform applications.

Get Mono

The latest Mono release is waiting for you!

Download

Read the docs

We cover everything you need to know, from configuring Mono to how the internals are implemented.

Our documentation is open source too, so you can help us improve it.

Community

As an open source project, we love getting contributions from the community. File a bug report, add new code or chat with the developers.

Contribute to Mono

Learn more



AMBIENTI ONLINE

Per sperimentare senza dover installare niente sul proprio computer (o usando un tablet o smartphone), si possono usare gli *ambienti online*

- disponibili per praticamente ogni linguaggio
- accessibili con un comune browser
- spesso chiamati «playground»
 - Java (Kotlin, Scala, Swift..): <https://code.labstack.com/java>
 - Java: <https://www.studytonight.com/code/playground/java/>
 - C# playground: <https://dotnetfiddle.net/srx9kM>
 - Kotlin playground: <https://play.kotlinlang.org/>
 - Scala: <https://scastie.scala-lang.org/>
<https://scalafiddle.io/>
<https://www.katacoda.com/courses/scala/playground>

...



AMBIENTI ONLINE

The image displays six screenshots of online environments for writing and running code:

- .NET Fiddle**: A C# playground interface with tabs for Options, Language (C#), Project Type (Console), Compiler (.NET 4.7.2), and a preview window showing a simple application that prints "Hello from .NET".
- Kotlin**: An online sandbox for Kotlin. It shows a code editor with a sample program that initializes a counter, increments it, and prints its value. The output window shows the result: 0.8660254.
- Scastie**: An online Scala editor. The code editor contains a sample program for a counter class. The output window shows the generated Java code for the counter.
- codingground**: An online Scala compiler. It shows a code editor with a sample program for a counter class. The output window shows the command \$scalac *.scala.
- sl**: A platform for learning programming. It features a Tutorials section with categories like NUMPY, TKINTER, KOTLIN, and JAVASCRIPT. Below it is a Java Compiler section with tabs for Practice Coding and Beacon.
- labstack**: An online Java editor. It shows a code editor with a sample Java program that prints "Hello from Java!". The output window shows the command java Main.



LA TIPICA OBIEZIONE (1/2)

MA.. così non ho un “vero eseguibile”!

- Occhio alle parole: cosa vuol dire “eseguibile”?
- Ogni file “eseguibile” lo è solo *nel contesto di una certa infrastruttura*, che gli fornisce il necessario contorno
 - un “eseguibile” Windows non funziona in MacOS
 - un “eseguibile” .NET non funziona in Windows senza .NET
 - un file Word non si apre senza Microsoft Word installato..
- Una entità è “eseguibile” (apribile) solo se è presente lo strato software che la interpreta, *non perché il suo nome finisce per “EXE”*
 - un JAR di Java è dunque “eseguibile” quanto un EXE di C#: ciò che conta è che sia presente l’infrastruttura di supporto



LA TIPICA OBIEZIONE (2/2)

Sarà anche vero, ma... *gli eseguibili “veri” si aprono facendoci sopra “doppio clic”!*

E INFATTI...

- con la giusta configurazione del JRE (vedere slide!) si rendono eseguibili con doppio clic le applicazioni Java (Scala, Kotlin) distribuite come archivio JAR
 - Android: archivio APK
- l'installazione del .NET framework rende eseguibili le applicazioni .NET distribuite come archivio EXE
 - Occhio al nome: gli "EXE di .NET non sono i classici eseguibili di Windows... tanto è vero che su Windows "puro" non girano!
 - con gli EXE "classici" hanno in comune solo la desinenza.

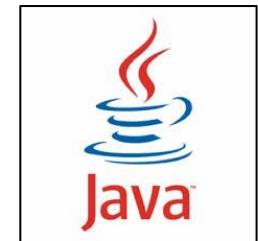


JAVA e C#: L'INFRASTRUTTURA

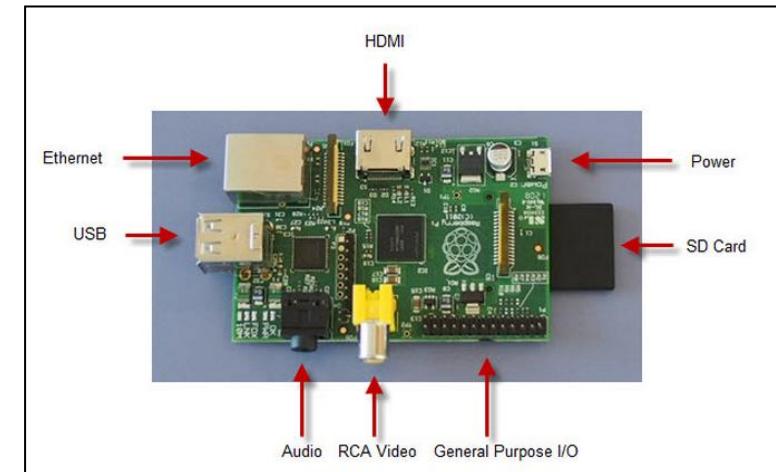
- A differenza di quanto accadeva in C (e C++), **Java e C# nascono dotati da una *infrastruttura standard portable***
 - indipendenza dalla piattaforma hardware / software
 - write once, run everywhere
- L'**infrastruttura offre servizi per coprire praticamente ogni esigenza del moderno sviluppo di software**
 - supporto per la grafica (JavaFX, Swing, Java2D, 3D..)
 - programmazione a eventi
 - supporto di rete: URL, Socket, connessioni...
 - supporto per il multi-threading e la concorrenza
 - interfacciamento con database (JDBC)
 - supporto per la sicurezza (cifratura, autenticazione..)
 - esecuzione remota (RMI)...

PORTABILITÀ: JAVA

- L' **infrastruttura Java** è disponibile per praticamente *tutte le piattaforme hardware/software*
- Spesso *embedded* in lettori DVD/BlueRay, TV,...
 - *guarda dietro se trovi il logo!*



- Disponibile anche su micro-computer low-cost e all-in-one come i Raspberry Pi





JAVA su Raspberry Pi

There's a **computer** for that

Whatever your application and whatever your scale, **Raspberry Pi** offers cost-effective, high-performance computing for businesses and the home.

Designed and manufactured in the UK.

[Browse our products](#)



The Pi4J Project
Java I/O library for Raspberry Pi

Search CTRL + K

[Home](#)

[About](#)

[Prepare a Raspberry Pi](#)

[Write OS to SD card](#)

[The First Boot](#)

[Install Java and Tools](#)

[Getting Started With Pi4J](#)

[Documentation](#)

Welcome to Pi4J > Prepare a Raspberry Pi > Install Java and Tools

[java](#) [javafx](#) [jbang](#) [maven](#) [pi4j os](#) [sdkman](#)

INSTALL JAVA AND TOOLS

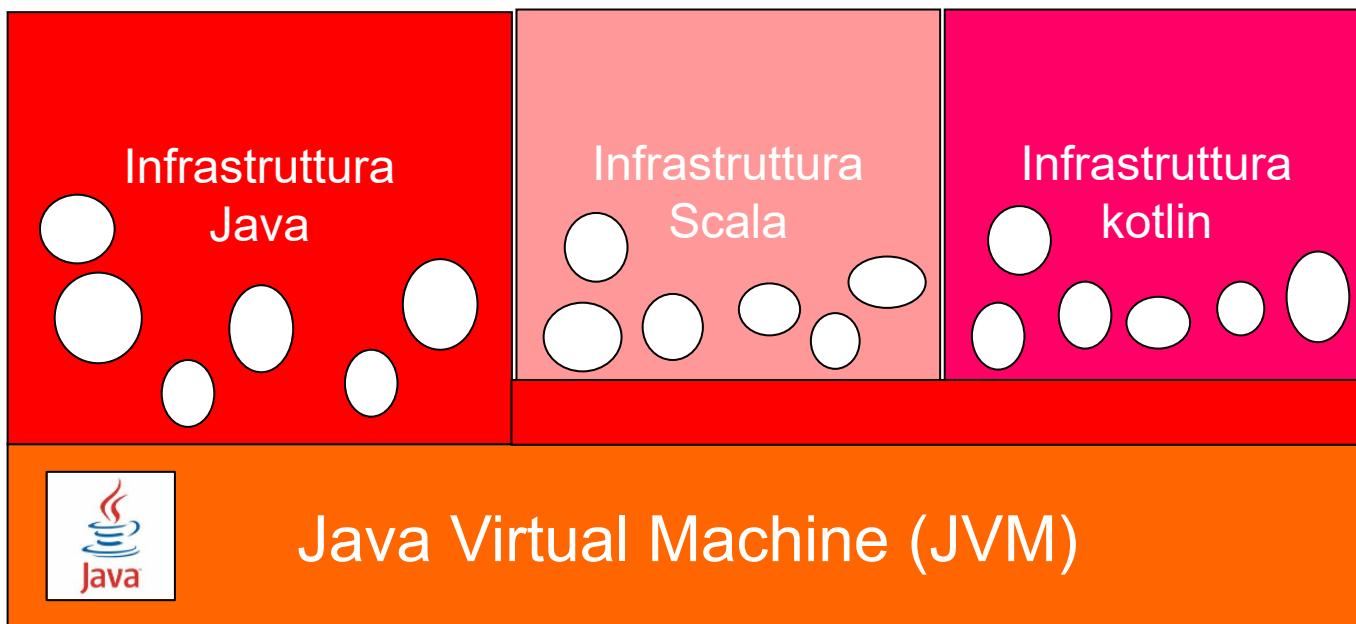
Our Raspberry Pi has started for the first time and we are now ready to add some Java to it!

Tip

As described on the previous page "The First Boot", a script can be executed that will do all this at once:
`curl -sL https://raw.githubusercontent.com/Pi4J/pi4j-os/main/script/prepare-for-java.sh | bash`.

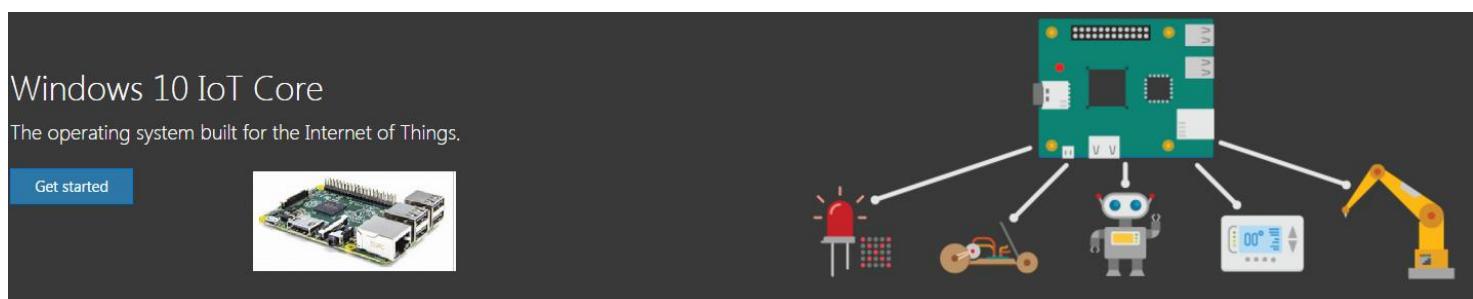
PORTABILITÀ: JAVA

- L' **infrastruttura Java** funge anche da base per molte altre infrastrutture e linguaggi, come *Scala* e *Kotlin*
 - di conseguenza, componenti Java, Scala, Kotlin sono interoperabili



PORTABILITÀ: .NET

- L' **infrastruttura .NET** è disponibile su più piattaforme
 - nativamente, sui sistemi Windows
 - tramite installazione separata per sistemi Linux/Mac
- Molti linguaggi supportati (ma non Java)
- Da Windows10, concetto di *Universal App*
 - unica app per desktop e mobile
- *Windows per IoT* – in particolare, *Windows IoT core* per singole applicazioni su schede, inclusi Raspberry PI





IL SUPPORTO

- Java e derivati, nonché C#, sono dotati da un *compilatore ottimizzato* che genera codice per l'infrastruttura
- A differenza di C e C++, però, qui **il compilatore è solo uno degli strumenti**: il kit include anche strumenti per:
 - produrre *adeguata documentazione* sempre aggiornata
→ *estrazione automatizzata di MANUALI HTML*
 - supportare attività di *collaudo sistematico*
→ *esecuzione automatizzata di SUITE DI TEST*
 - progettare *la distribuzione dell'applicazione* (o *libreria*), eventualmente anche con firma digitale (per garantirne autenticità)
→ *impatto sul processo produttivo*
→ *impatto sull'organizzazione dell'applicazione*



IL PROBLEMA DEL DEPLOYMENT

Deployment = distribuzione di un software

- Non può essere lasciata al "fai da te":
anche questa attività va organizzata e progettata
 - è intrinsecamente non scalabile: 10 installazioni sono una cosa, 1000 o 10000 sono tutt'altra faccenda!
- **Cosa distribuire?**
 - supportare N piattaforme hardware & software *non può significare dover distribuire N eseguibili diversi...* è ingestibile!
 - ideale: distribuire un solo file, per tutte le possibili piattaforme
- **Un archivio di *componenti compattati*, simile a uno ZIP**
 - magari usabile *senza bisogno neanche di scompattarlo...* ☺
 - per Java e derivati: **formato JAR** (Java ARchive)