

Alma Mater Studiorum-Università di Bologna Scuola di Ingegneria

Java e C#: deployment ESERCITAZIONE AUTONOMA

Corso di Laurea in Ingegneria Informatica

Anno accademico 2025/2026

Prof. ENRICO DENTI

Dipartimento di Informatica – Scienza e Ingegneria (DISI)



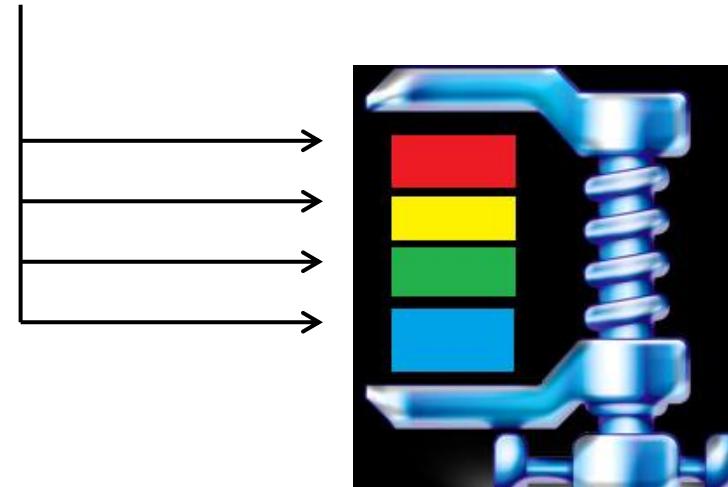
RIASSUNTO: IL DEPLOYMENT DEL SOFTWARE

In Java e C#, non esiste più l'eseguibile monolitico.

- Per distribuire un'applicazione o libreria, si **impacchettano le relative in un "archivio compattato"**
 - un solo file, ma *ogni classe mantiene la propria individualità*
- In Java, formato JAR (Java ARchive)
 - in C#, formato DLL / EXE

Per distribuire

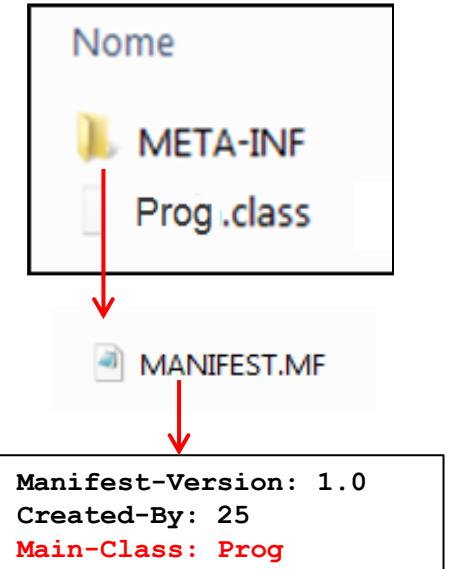
- **librerie:** bastano le classi
- **applicazioni:** bisogna anche dire *dove si trova il main*





RIASSUNTO: IL FORMATO JAR

- **JAR (Java ARchive)** è un normale file ZIP
 - si apre con Unzip, 7-Zip, etc.
- che però **contiene al suo interno una speciale cartella META-INF** con un file **MANIFEST.MF**, che mantiene una serie di *informazioni extra*
 - per le applicazioni: *la posizione del main*
- Ciò rende possibile eseguire un'applicazione Java semplicemente **facendo doppio clic sul file JAR**, come in un eseguibile tradizionale.



Gli archivi JAR si creano con lo **strumento jar** incluso nel JDK.

- È uno strumento a riga di comando: occorre apprenderne l'uso
- Analoghe funzionalità sono offerte dagli ambienti integrati (Eclipse)



RIASSUNTO: CREARE UN FILE JAR

Per creare un file JAR con una libreria:

jar cf nome.jar classi

Per creare un file JAR con un'applicazione eseguibile:

jar cmf info.txt nome.jar classi

dove **info.txt** è un *file di testo* che contiene la riga:

Main-Class: NomeClasseMain

In alternativa, si può evitare il file **info.txt** usando il comando **e**:

jarcef NomeClasseMain nome.jar classi

Come primo esercizio, incapsuleremo come JAR l'applicazione "*Equazioni di secondo grado*" precedentemente sviluppata.



L'APPLICAZIONE EQUAZIONI DI 2° GRADO

- Soluzione corretta (con finezze di stampa)

```
public class Equation {
    public static void main(String[] args){
        if (args.length!=3) {
            System.out.println("Errore: fornire i tre coefficienti a, b, c separati da spazi");
        }
        else {
            double a = Double.parseDouble(args[0]);
            double b = Double.parseDouble(args[1]);
            double c = Double.parseDouble(args[2]);
            if (a==0) {
                System.out.println("Equazione di 1 grado: " + b + "x" + (c>=0?"+" :"") + c + " = 0");
                System.out.println("Soluzione: " + (-c/b));
            }
            else {
                System.out.println("Equazione di 2 grado: " + a + "x^2" + (b>=0?"+" :"") + b + "x" + (c>=0?"+" :"") + c + " = 0");
                double delta = b*b-4*a*c;
                if (delta<0) {
                    double q = Math.sqrt(-delta)/(2*a);
                    double p = -b/(2*a);
                    // System.out.println("Soluzioni immaginarie: " + p + '\u000B1' + q + "i"); // per prompt comandi (codice 177)
                    System.out.println("Soluzioni immaginarie: " + p + '\u000F1' + q + "i"); // per finestre grafiche windows (codice 241)
                }
                else if (delta>0) {
                    double q = Math.sqrt(delta)/(2*a);
                    double p = -b/(2*a);
                    // System.out.println("Soluzioni reali: " + p + '\u000B1' + q); // per prompt comandi (codice 177)
                    System.out.println("Soluzioni reali: " + p + '\u000F1' + q); // per finestre grafiche windows (codice 241)
                }
                else /* delta==0 */ {
                    double p = -b/(2*a);
                    System.out.println("Soluzioni coincidenti: " + p);
                }
            }
        }
    }
}
```



ESERCIZIO 1

- Situazione: applicazione costituita da una singola classe
 - l'applicazione è già stata compilata, c'è un unico file .class

Nome	Ultima modifica	Tipo
Equation.class	27/02/2020 16:22	File CLASS
Equation.java	27/02/2020 16:22	File JAVA

- Per generare un'app eseguibile occorre predisporre un file con la posizione del main: chiamiamolo posmain.txt

Equation.class	27/02/2020 16:22	File CLASS	2 KB
Equation.java	27/02/2020 16:22	File JAVA	2 KB
posmain.txt	27/02/2020 17:59	Documento di testo	0 KB

posmain.txt - Blocco note

File Modifica Formato Visualizza ?

Main-Class: Equation



ESERCIZIO 1

- Per creare il JAR eseguibile:

jar cmf posmain.txt Equation.jar *Equation.class*

- Per eseguire il JAR così ottenuto:

java -jar Equation.jar argomenti

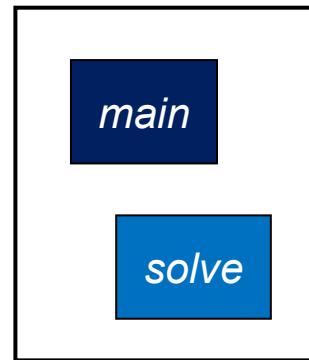
```
E>jar cmf posmain.txt Equation.jar Equation.class  
E>java -jar Equation.jar 1 -5 6  
Equazione di 2 grado: 1.0x^2-5.0x+6.0 = 0  
Soluzioni reali: 2.5±0.5
```



ESERCIZIO 2

REFACTORING SU DUE FILE

- Obiettivo: separare il main dalla "business logic"
 - DUE CLASSI: **Equation** e **LaunchEquation**
 - nella classe **Equation** non c'è più alcun main: al suo posto, la funzione **solve** riceve *numeri* e *restituisce una stringa*
→ *NON stampa nulla autonomamente*
 - la classe con il main, **LaunchEquation**, converte gli argomenti da stringa a numero e invoca la funzione **solve** di **Equation**
- Nuova struttura:





ESERCIZIO 2

REFACTORING SU DUE FILE

```
public class LaunchEquation {  
    public static void main(String[] args){  
        if (args.length!=3) {  
            System.out.println("Errore: fornire i tre coefficienti a, b, c separati da spazi");  
        }  
        else {  
            double a = Double.parseDouble(args[0]);  
            double b = Double.parseDouble(args[1]);  
            double c = Double.parseDouble(args[2]);  
            String result = Equation.solve(a,b,c);  
            System.out.println(result);  
        }  
    }  
  
    public class Equation {  
        public static String solve(double a, double b, double c) {  
            String result;  
            if (a==0) {  
                result = "Equazione di 1 grado: " + b + "x" + (c>=0?"":"") + c + " = 0" + System.lineSeparator()  
                    + "Soluzione: " + (-c/b);  
            }  
            else {  
                result = "Equazione di 2 grado: " + a + "x^2" + (b>=0?"":"") + b + "x" + (c>=0?"":"") + c + " = 0" + System.lineSeparator();  
                double delta = b*b-4*a*c;  
                if (delta<0) {  
                    double q = Math.sqrt(-delta)/(2*a);  
                    double p = -b/(2*a);  
                    result += "Soluzioni immaginarie: " + p + '\u00081' + q + 'i'; // per prompt comandi (codice 177)  
                    // result += Soluzioni immaginarie: + p + \u00081 + q + i; // per finestre grafiche windows (codice 241)  
                }  
                else if (delta>0) {  
                    double q = Math.sqrt(delta)/(2*a);  
                    double p = -b/(2*a);  
                    result += "Soluzioni reali: " + p + '\u00081' + q; // per prompt comandi (codice 177)  
                    // result += Soluzioni reali: + p + \u00081 + q; // per finestre grafiche windows (codice 241)  
                }  
                else /* delta==0 */ {  
                    double p = -b/(2*a);  
                    result += "Soluzioni coincidenti: " + p;  
                }  
            }  
            return result;  
        }  
    }  
}
```

Solve restituisce una stringa
in cui accumula il messaggio

System.lineSeparator()
restituisce il separatore di riga
della piattaforma corrente



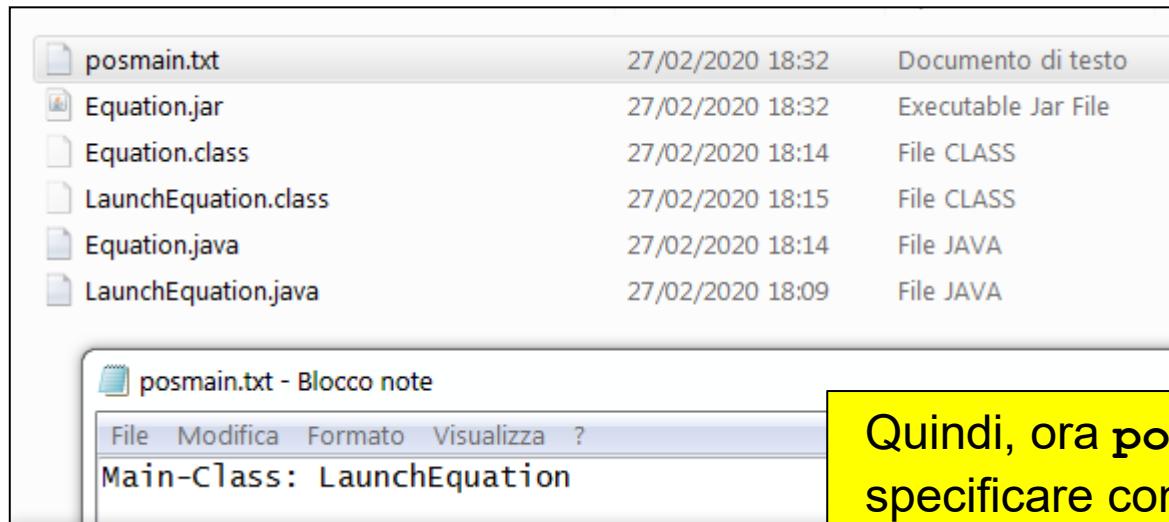
ESERCIZIO 2

REFACTORING SU DUE FILE

- Per creare il JAR occorre includere ora **entrambe** le classi:

jar cmf posmain.txt Equation.jar

Equation.class LaunchEquation.class



Quindi, ora **posmain.txt** deve specificare come entry point la classe **LaunchEquation**



ESERCIZIO 2

REFACTORING SU DUE FILE

- Per creare il JAR occorre includere ora **entrambe** le classi:

```
jar cmf posmain.txt Equation.jar
```

Equation.class LaunchEquation.class

- ora **posmain.txt** specifica come entry point **LaunchEquation**

```
E>jar cmf posmain.txt Equation.jar Equation.class LaunchEquation.class
```

- oppure, più brevemente

```
jar cmf posmain.txt Equation.jar *.class
```

- Per eseguire:

```
E>java -jar Equation.jar 1 -5 6  
Equazione di 2 grado: 1.0x^2-5.0x+6.0 = 0  
Soluzioni reali: 2.5±0.5
```



E SE LO VOLESSIMO RIFARE IN C#?

- A livello di codice le modifiche sono minime:
 - utile mettere la direttiva iniziale «`using System`» per comodità
 - `string` al posto di `String`
 - `Main, Sqrt, Length` al posto di `main, sqrt, length`
 - `Convert.ToDouble(...)` al posto di `Double.parseDouble(...)`
- Per compilare si può agire in due modi:
 1. compilare insieme i due file `Equation.cs` e `LaunchEquation.cs` ottenendo direttamente l'eseguibile `LaunchEquation.exe`

```
E>csc /nologo LaunchEquation.cs Equation.cs
E>dir LaunchEquation.*
```

02/03/2020 17:40	427	LaunchEquation.cs
02/03/2020 17:50	5.120	LaunchEquation.exe
2 File	5.547 byte	
0 Directory	730.664.136.704 byte disponibili	



E SE LO VOLESSIMO RIFARE IN C#?

- oppure:
 2. compilare separatamente prima **Equation.cs** → **Equation.dll**
e poi **LaunchEquation.cs** linkando la precedente DLL
Si ottiene così l'eseguibile **LaunchEquation.exe**

```
E>csc /nologo /target:library Equation.cs
E>dir *.dll
02/03/2020  17:54           4.096 Equation.dll
              1 File        4.096 byte
              0 Directory   730.664.136.704 byte disponibili
E>csc /nologo /reference:Equation.dll LaunchEquation.cs
E>dir *.exe
02/03/2020  17:55           4.096 LaunchEquation.exe
              1 File        4.096 byte
              0 Directory   730.664.140.800 byte disponibili
```

NB: l'opzione **/nologo** serve semplicemente per ridurre la verbosità dei messaggi nell'output del compilatore



E SE LO VOLESSIMO RIFARE IN C#?

- Per eseguire, nulla cambia:

```
E>LaunchEquation 1 5 6  
Equazione di 2 grado: 1x^2+5x+6 = 0  
Soluzioni reali: -2,5±0,5  
  
E>LaunchEquation 1 5 -6  
Equazione di 2 grado: 1x^2+5x-6 = 0  
Soluzioni reali: -2,5±3,5  
  
E>LaunchEquation 0 5 -6  
Equazione di 1 grado: 5x-6 = 0  
Soluzione: 1,2  
  
E>LaunchEquation 1 0 1  
Equazione di 2 grado: 1x^2+0x+1 = 0  
Soluzioni immaginarie: 0±1i  
  
E>LaunchEquation 1 0 -1  
Equazione di 2 grado: 1x^2+0x-1 = 0  
Soluzioni reali: 0±1
```