

00

# Programmazione ad Oggetti

## Introduzione al corso

Mirko Viroli  
`mirko.viroli@unibo.it`

C.D.L. Ingegneria e Scienze Informatiche  
ALMA MATER STUDIORUM—Università di Bologna, Cesena

a.a. 2022/2023

## Titolare del corso: Prof. Mirko Viroli

- e-mail — [mirko.viroli@unibo.it](mailto:mirko.viroli@unibo.it)
- homepage — <https://www.unibo.it/sitoweb/mirko.viroli>

## Modulo di laboratorio 1: Prof. Danilo Pianini

- e-mail — [danilo.pianini@unibo.it](mailto:danilo.pianini@unibo.it)
- homepage — <https://www.unibo.it/sitoweb/danilo.pianini>

## Modulo di laboratorio 2: Prof. Roberto Casadei

- e-mail — [roby.casadei@unibo.it](mailto:roby.casadei@unibo.it)
- homepage — <https://www.unibo.it/sitoweb/roby.casadei>

# Contatti con gli studenti

## Chi contattare

- Mirko Viroli: “teoria” + organizzazione del corso
- Pianini/Casadei: “pratica” + laboratorio

## Attraverso il forum

- Per domande la cui risposta è di interesse generale (e quindi tutte le domande tecniche)

## Via mail al docente

- Per questioni personali
- Per domande che consentono risposte concise

## Ricevimento

- (annunciato nelle Home Page dei docenti)

## Sito **virtuale** di Ateneo

- <https://virtuale.unibo.it/course/view.php?id=38482>
- sarà il luogo degli avvisi (e notifiche), forum di discussione, produzione di materiale
- tutti gli studenti che seguono il corso si iscrivano, e lo tengano d'occhio
- in particolare: avendo sempre le slide sottomano

# Organizzazione generale del corso

## Lezioni aula (due da 3 ore la settimana)

- Illustrano i concetti teorici, metodologici e pratici
- Basate su slide proiettate (ma non solo)

## Laboratorio (turni da 3-4 ore a settimana)

- A giorni alterni (il che vi lascia un giorno libero)
- Illustra ulteriori aspetti metodologici e pratici
- Con esercizi necessari alla comprensione e alla sperimentazione
- È parte integrante del corso

## Studio a casa (almeno 4 ore a settimana) – p.e., nel giorno libero

- Rilettura slide, esperimenti pre- e/o post-laboratorio
- È praticamente necessario se volete rimanere in pari...

# Programma (di massima) del corso

## Parti principali

- Elementi base di programmazione OO e Java
- Polimorfismo (ereditarietà, subtyping, genericità)
- Librerie (I/O, grafica, concorrenza)
- Integrazione col paradigma funzionale (lambda, streams)
- Pattern e buone pratiche di programmazione
- Elementi di programmazione C#

# Testi di riferimento (non necessario l'acquisto)

## Programmazione in Java

- B.Eckel. Thinking in Java, 4th edition.
- J.Block. Effective Java, 2nd edition.
- R.Warburton. Java 8 Lambdas.

## Programmazione in C#

- Jon Skeet. C# in depth, 3rd edition.

## Altri riferimenti

- E.Gamma et.al. Design Patterns Elements of Reusable Object-Oriented Software.
- R.Martin. Clean Code: A Handbook of Agile Software Craftsmanship
- Java e C# online documentation (tutorials, Language Specification, APIs)

## Java

- Framework Java: OpenJDK 17 (Open Java Development KIT)
  - ▶ <https://adoptium.net/>
- Integrated Development Environment: Visual Studio Code + Java plugin
  - ▶ <https://code.visualstudio.com/>
  - ▶ <https://code.visualstudio.com/docs/languages/java>
- Altri strumenti
  - ▶ Git
  - ▶ Gradle

## Altri framework – parte finale del corso

- Visual Studio .NET (Microsoft), oppure
- framework Mono e IDE Monodevelop/Rider (Linux e MacOS)

## Istruzioni sull'installazione (sul PC di casa)

- Già disponibili su “virtuale”
  - <https://apice-at-disi.github.io/software-installation-instructions/>
  - Molto importante rendersi operativi a casa nel giro di una settimana!
  - L'uso di VSCode sarà necessario solo fra 2/3 settimane
- ⇒ sarebbe consigliato l'uso del sistema operativo Linux



# Sul ruolo di questo corso

## Elementi essenziali

- Costruzione del software, e quindi di sistemi
- Analisi problemi, e organizzazione di soluzioni
- Tecniche base ed (alcune) avanzate di programmazione ad oggetti
- Introduzione a principi/tecniche di “programmazione moderna”
- Elementi di gestione di un progetto software
- Utilizzo di strumenti integrati di sviluppo

## Questo corso nel più ampio contesto dei vostri studi

- Competenze a curriculum
- Enfasi sull'approccio metodologico
- Target di qualità piuttosto elevato
- È cruciale dedicargli subito il tempo necessario

## Prova scritta

- Durata 1.5 ore circa, svolta in laboratorio
  - Verifica mirata di capacità tecniche, di problem-solving, buona progettazione OO
- ⇒ Forniremo esercizi-tipo in laboratorio
- ⇒ I temi d'esame del passato sono disponibili e abbastanza indicativi
- ⇒ Svolta in laboratorio

## Discussione progetto

- Progetto sviluppato in gruppo, 70-80 ore a testa (vedi i 12 CFU)
  - Concordato col docente prima di iniziare
  - Da relazionare con qualità, poi discusso su appuntamento
  - Consegne con deadline scelta da voi ma a quel punto stringente: orientativamente, fine Febbraio, Aprile, Giugno, Agosto, Ottobre
  - In discussione si porta anche un micro-task C# (< 10 ore)
- ⇒ I dettagli discussi a metà corso
- ⇒ Regole d'esame già disponibili (in versione draft) sul “virtuale”

# Prerequisiti

## Buona conoscenza

- tecniche di programmazione imperativa/strutturata
- costruzione e comprensione di semplici algoritmi e strutture dati

## Attenzione a chi è già “fluente” in linguaggi ad oggetti

- Java o C#
- disimparare le “bad practice” richiede umiltà e fatica...