



POLITECNICO
MILANO 1863

Introduzione alla Prova Finale

19 Marzo 2019

Scaglione San Pietro

RESPONSABILI

Giovanni Meroni
Amarildo Likmeta

TUTOR

Valentina Deda
Marco Bacis

Obiettivi della Prova Finale

- Mettere in pratica quanto imparato durante il corso di Ingegneria del Software...
- ... ed in generale durante il percorso di studi, per gli aspetti legati allo sviluppo del software
- Progettare, verificare, documentare ed implementare, secondo un approccio di tipo "object-oriented" una applicazione reale multiutente, distribuita e con interfaccia grafica
- Organizzare e gestire il processo di sviluppo del software e i relativi prodotti

La vostra prova

Nello specifico, vi viene richiesto di sviluppare un software che emula un gioco da tavolo



Valutazione della Prova

- Basata su una demo del progetto con presentazione, analisi del codice e della documentazione
 - Arricchita da una valutazione "in itinere" da parte della squadra
 - Il progetto si presenta con almeno due PC "in rete"
- Voto in 30-esimi, come per ogni altro corso
- Voto registrato alla fine del laboratorio (entro la prima metà di Luglio)
- Contribuisce, con gli altri corsi di tipo "prova finale", all'incremento con cui si "costruisce" il voto di laurea ($I = 1 + V/6$)
- **Il progetto DEVE essere sviluppato durante questo semestre**
- **La prova NON può essere recuperata**
- NON esistono appelli durante l'anno: se non viene superato occorre rifrequentare l'anno successivo

Valutazione della Prova

Oltre alle funzionalità implementate, il voto finale terrà conto di:

- **Qualità della *progettazione*** con particolare riferimento ad un uso appropriato di interfacce, ereditarietà, composizione tra classi, uso di pattern (statici, di comunicazione e architetturali) e divisione delle responsabilità.
- **La stabilità dell'implementazione e la conformità alle specifiche**
- La **leggibilità** del codice scritto, con particolare riferimento a nomi di variabili/metodi/classi/package, all'inserimento di commenti e documentazione JavaDoc (preferibilmente in inglese), la mancanza di codice ripetuto e metodi di eccessiva lunghezza.
- L'**efficacia** e la **copertura dei casi di test**, il nome e i commenti di ogni test dovranno chiaramente specificare le funzionalità testate e i componenti coinvolti.
- L'**utilizzo degli strumenti** (IntelliJ, Sonar, Git, Maven ..)
- **Conoscenza di tutto il codice da parte di tutti i componenti del gruppo**
- L'**autonomia**, l'**impegno** e la **comunicazione** (con i responsabili e all'interno del gruppo)

Warnings

Qualunque gruppo che venga trovato a copiare verrà **"bocciato"** e dovrà ripetere il corso l'anno prossimo

1

- Non copiare da altri, da altre sezioni, o da interi pezzi di codice trovati in rete...
- ... fidatevi, i controlli funzionano! (<https://theory.stanford.edu/~aiken/moss/>)

2

L'uso di librerie esterne deve essere approvato dai responsabili.

3

È vietato l'uso di tool che facilitano la creazione della GUI

Potete usare Swing o JavaFX

piazza

**Forum ufficiale dei 3 scaglioni
per discutere con colleghi e responsabili**

piazza.com/polimi.it/spring2019/085923

Se non l'avete già fatto REGISTRATEVI!

**La partecipazione al forum verrà tenuta in
considerazione per la valutazione finale.**

**L'interazione con tutor/responsabili al di fuori dell'orario
di laboratorio deve avvenire esclusivamente attraverso
questo strumento.**

Formazione Gruppi

GRUPPI DA 3 PERSONE



TUTTE APPARTENENTI ALLA STESSA 'SEZIONE'

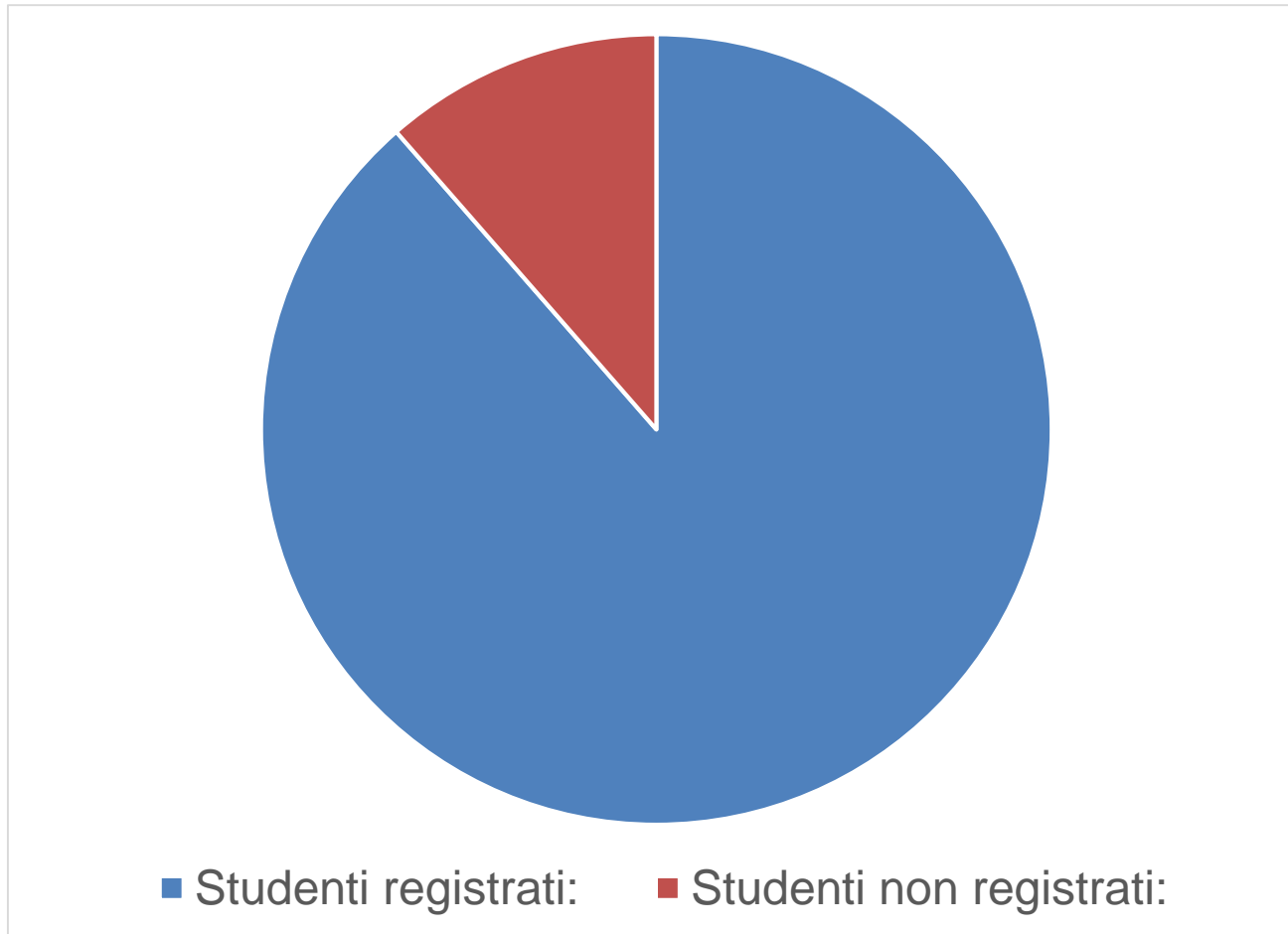
Form registrazione gruppi

<https://goo.gl/forms/8fSkoQfh9X7ePYIn1>

Entro la fine di questo laboratorio

Potete usare Piazza per trovare compagni

Stato studenti



Frequenza

A series of thin, light blue vertical lines of varying heights, creating a comb-like or barcode-like decorative element.

La frequenza è **obbligatoria**

MAX 2 ASSENZE

A thin, light blue horizontal line with a dotted pattern.

Ricordatevi di firmare il foglio presenze!

Calendario Laboratorio (32h)

NON DEFINITIVO

19 Marzo	Focus: Introduzione + Setup Ambiente di sviluppo Checkpoint: Download Strumenti (IntelliJ, Sonar, JDK 8, ...)
26 Marzo	Focus: UML design + Pattern MVC Checkpoint: Setup completo + gruppi formati + repository creato
2 Aprile	Focus: Testing, JUnit, JavaDoc Checkpoint: UML (Modello)
9 Aprile	Focus: Comunicazione, Multithreading e Protocol design Checkpoint: UML (Modello) consolidato e implementato + MVC
7 Maggio	Focus: Swing e GUI Design Checkpoint: Implementazione comunicazione, Test coverage e documentazione
14 Maggio	Focus: Discussione Generale Checkpoint: Implementazione funzionalità selezionate
21 Maggio	Focus: Discussione Generale Checkpoint: Implementazione funzionalità selezionate + GUI
28 Maggio	Focus: Discussione Generale Checkpoint: Implementazione funzionalità selezionate + GUI
-	SUPPORTO LIMITATO
? Luglio	Presentazione



- 1] Ogni studente deve creare un account personale GitHub
- 2] Una volta stabilito il gruppo di appartenenza, uno dei membri del gruppo crea un repository **PRIVATO** denominato
`ing-sw-2019-cognome1-cognome2-cognome3`
- 3] Invitate al repo tutti i membri del gruppo e i responsabili:
`@meronig` `@amarildolikmeta`

Repository su GitHub

Vi viene inoltre richiesto di creare la cartella '**deliveries**'
nella root del repository

Periodicamente dovrete caricare uno screenshot del **report Sonar**,
oltre ai **diagrammi UML** richiesti.

Solo commit significative

La qualità delle commit (titoli, commenti) verrà presa in considerazione nella
valutazione finale.

(evitare di comminare ad ogni singola modifica del codice)

<https://brigade.engineering/the-secrets-to-great-commit-messages-106fc0a92a25>



Ci aspettiamo
un progetto **compilabile** ed **eseguibile** con Maven
(senza errori!)

TODO per il prossimo Lab

- Ogni gruppo deve avere una repo (privata) su GitHub, rispettando le regole indicate, in particolare:
 - I responsabili devono essere invitati alla repo (usernames: amarildolikmeta, meronig)
 - Deve essere presente il file .gitignore (su BeeP) nella cartella radice
 - Deve essere stato fatto il primo commit del progetto, realizzato mediante Maven con il file pom.xml corretto (su BeeP)
- Tutti devono essere registrati a GitHub e far parte della repo associata al proprio gruppo
- Il nome del repo GitHub deve essere comunicato mediante apposita form
- Tutti devono aver compreso le regole del gioco Adrenalina