

Laboratorio di Programmazione Internet: progetto a.a. 2023/2024

Angelo Di Iorio
Roberto Amadini
Sara Zuppiroli

Tempi regole e vincoli

Attività e tempi del progetto in breve

- Creazione dei gruppi (2 o 3 persone a gruppo)
Il progetto può essere implementato utilizzando github
- **Consegna su Virtuale della [proposta degli esercizi](#)** (dettagli nelle prossime slide)
- **Consegna della proposta almeno una trentina di giorni prima dell'invio del progetto** (le date esatte nelle prossime slide)
- La proposta è valida per tutti gli appelli successivi, ma può essere aggiornata, inviando una nuova proposta entro le date indicate per la validazione (vedi prossime slide)
- Valutazione degli esercizi proposti e possibili correzioni da apportare, **dopo massimo 5 giorni** dalla presentazione della proposta, vale la regola del silenzio assenso.
- Consegna del progetto **almeno 7 giorni** prima delle date di discussione. I progetti possono essere discussi singolarmente

Discussione e consegna

- Il progetto si discute **SOLO dopo aver superato** la prova di laboratorio. **Non è obbligatorio**, ma consigliato, che tutti i membri del gruppo siano presenti*
- Il **progetto si consegna prima della data di discussione** (~7 giorni prima) su virtuale.unibo.it
- **Queste specifiche sono valide solo entro il 28/5/2025 (ultima data utile per discutere questo progetto).**
- Discussioni in presenza con indicazioni su virtuale e prenotazione su almaesami

* il vincolo per cui tutti i membri del gruppo avrebbero dovuto superare la prova di laboratorio prima della discussione è stato rimosso

Consegna

- Il progetto si consegna su Virtuale, ogni componente del gruppo che vuole discutere deve indicare nome del gruppo e consegnarlo personalmente, anche se si discute tutti insieme. La consegna è uguale per ogni membro del gruppo.
- Saranno pubblicate apposite sezioni su Virtuale
- Regole di consegna:
 - Consegnare una cartella compressa contenente:
 - link di accesso al repository github (se utilizzato, scelta consigliata)
 - Il link accessibile al codice sorgente
 - **Breve** relazione: max 3 pagine, con qualche screenshot dell'applicazione e una sequenza step-by-step di istruzioni per avviarla (in formato txt/md/pdf, no .odt o .docx)
 - Video illustrativo di 2 minuti: mostra l'applicazione in funzione ed è usato solo come backup nel caso non si riesca ad eseguire il codice consegnato

Linea del tempo

Cons. num.	Data per proposta	Data consegna progetto	Esame di laboratorio superato entro	Data discussione progetto
1	20/05/24	21/06/24	09/06/24	28/06/24
2	17/06/24	19/07/24	10/07/24	26/07/24
3	12/08/24	16/09/24	10/09/24	23/09/24
4	02/01/25	02/02/25	Verificare le date d'esame	10/02/25
5	18/04/25	21/05/25	Verificare le date d'esame	28/05/25

Valutazione progetto

- L'applicazione deve seguire le seguenti regole di implementazione:
 - Uso corretto di definizione delle classi
 - Utilizzo di Ereditarietà, Polimorfismo
 - Commenti al codice
 - Gestione degli errori
- Deve essere frutto del vostro lavoro
- La valutazione non è di gruppo ma singola.

Valutazione progetto

VOTO FINALE \approx VOTO LABORATORIO ± 3

- Il voto di laboratorio è la **base** con cui presentarsi alla discussione del progetto
- **Il progetto completa l'esame di laboratorio dando un bonus**
 - Salvo **eccezioni**, a seconda del progetto e della sua discussione orale il voto di laboratorio può essere:
 - **confermato**
 - **decrementato** da 1 a 3 punti
 - **incrementato** da 1 a 3 punti
- Una volta discusso il progetto non è possibile rifare l'esame di laboratorio



Il progetto



PLAY (Play, Learn, Assess Yourself)

- è un'applicazione stand-alone in Java FX per **fare esercizi divertenti imparare a programmare**:
 - si svolgono diversi esercizi che allenano diversi aspetti della conoscenza di un linguaggio di programmazione (abilità):
 - lettura del codice, individuazione di errori, ordine corretto delle operazioni, confronto tra codici simili, correzione degli errori, ecc...
 - ogni esercizio ha 3 gradi di difficoltà (principiante, intermedio, avanzato)

Architettura PLAY

- Play è un'applicazione JavaFX stand-alone:
 - Si avvia l'applicazione
 - l'utente deve far accesso all'applicazione con una semplice autenticazione.
 - Si accede a un'interfaccia in cui si vedono i diversi tipi di esercizi, volti all'insegnamento della programmazione, raggruppati per tipologia e grado.
 - l'applicazione visualizza una barra di avanzamento di ogni esercizio a seconda del livello raggiunto.
 - l'utente attiva un esercizio e gioca utilizzando il mouse o la tastiera a seconda dell'esercizio scelto.

Workflow PLAY

1. L'utente si autentica (tramite username e password si legge il file se ne è in possesso) o crea l'utenza con il pulsante dedicato ([schermata 1a](#)), si scrive il dato su file
2. L'utente accede alla videata iniziale, griglia divisa per tipologia di esercizio con le immagini degli esercizi e indicazione sul grado di apprendimento raggiunto.
3. L'utente clicca sull'icona o la descrizione e visualizza una breve descrizione dell'esercizio, Quando ha letto se preme ok accede all'esercizio, altrimenti torna al punto 2 con annulla.
4. L'utente preme il pulsante inizia e svolge l'esercizio. Alla fine se supera l'esercizio, se ne mostra uno simile fino al raggiungimento di **n** successi. Se supera gli n esercizi si passa al grado di difficoltà successivo se esiste.

Workflow PLAY

5. In ogni momento l'utente può sospendere l'esercizio in esecuzione con il pulsante esci dall'esercizio, se l'esercizio non è ancora stato concluso sarà considerato come un fallimento, altrimenti sarà valutato. Si torna a 2.
6. A fine partita il risultato viene visualizzato su una maschera e contestualmente salvato (su file). Sul file si salvano nome utente, numero e nome esercizio, se superato o fallito il tentativo, e grado raggiunto
7. L'utente può fare logout o chiudere l'applicazione in ogni momento e il sistema dovrà salvare i dati relativi al login su di un file.

Videata iniziale (1)

Login

Username	<input type="text"/>
Password	<input type="password"/>

ACCEDI

Crea Nuovo Utente

1a

Crea nuovo utente (1a)

Crea Nuovo Utente

Username	<input type="text"/>	<input type="button" value="X"/>	7
Password	<input type="password"/>		
Nome	<input type="text"/>		
Cognome	<input type="text"/>		
.....	<input type="button" value="Salva"/>	<input type="button" value="Annulla"/>	

Videata iniziale (2)

CIAO
Mario Rossi

<h2>Leggi il codice</h2> <div>Cosa stampa?(Esperto)</div> <div><div></div><div></div></div>	<h2>Ordine</h2> <div>Ordina i passi (Intemedio)</div> <div><div></div><div></div></div>	<div>X</div> <div>7</div>
<h2>Il debug</h2> <div>Trova l'errore (Esperto)</div> <div><div></div></div>	<h2>Studia i concetti</h2> <div>Il quiz (Principiante)</div> <div><div></div></div>	

Videata Regole (3)

Leggi il codice

Cosa stampa?

COSA DEVI FARE

bla bla bla...

OK Annulla

7

Esercizio (4)

Leggi il codice (1/6) X 7

testo....

a) stampa 1
b) stampa 2
c) non stampa nulla

Risposta

Domanda Precedente Prossima domanda Esci dall'esercizio

5 si salva il fallimento

Esercizio (4)

Leggi il codice (6/6) X 7

testo....

a) stampa 1
b) stampa 2
c) non stampa nulla

Risposta

Risultato

Esci dall'esercizio

5 si salva il fallimento

Risultato (6)

The screenshot shows a quiz result screen with a green background. At the top left, the text 'Leggi il codice Totale' is displayed. In the top right corner, there is a red square containing a white 'X' and a score of '7'. Below this, a light blue box contains the message: 'Bene hai superato l'esercizio e hai risposto correttamente a 4 domande su 6. Prova con un nuovo esercizio.' At the bottom, there are two buttons: a blue button labeled 'Nuovo Esercizio' and a red button labeled 'Esci dall'esercizio'. To the right of the red button, a blue arrow points to it, accompanied by the text '5 In questo caso si salva il fallimento o il successo'.

Leggi il codice Totale

X 7

Bene hai superato l'esercizio e hai risposto correttamente a 4 domande su 6.
Prova con un nuovo esercizio.

Nuovo Esercizio

Esci dall'esercizio

5 In questo caso si salva il fallimento o il successo

Qualche libreria

- Esistono diversi formati per serializzare i dati: CSV (Comma Separated Values), JSON (Javascript Object Notation), XML (eXtensible Markup Language), *custom*
- Qualche libreria utile in Java (non vincolante!):
 - JSON serializer: <https://github.com/google/gson>
 - XML serializer: <https://www.edureka.co/blog/serialization-of-java-objects-to-xml-using-xmlencoder-decoder/>
 - CSV serializer: <http://opencsv.sourceforge.net/>

Requisiti obbligatori ed estensioni

- Queste specifiche sono poco vincolanti di proposito, siete invitati a personalizzare e aggiungere funzionalità
- E' richiesto tuttavia implementare **tutte le funzionalità indicate come obbligatorie** nelle slide precedenti

Qualche idea (non vincolante) per estensioni

- Comunicazione tramite mail/telegram/... dei risultati di una tipologia di esercizi → bisogna tenere traccia oltre che dello username anche di un recapito per gli utenti
- Extension pack: si prevede un utente amministratore che possa aggiungere esercizi (tipologia, gradi e i diversi esercizi)
- Multi-lingua: gli esercizi possono essere descritti in lingue diverse, configurabili dall'amministratore a seconda dell'utente
- l'applicativo usa dei timer per tener traccia del tempo di completamento
- si possono generare classifiche
-

Gli esercizi e le loro regole

- Gli esercizi da presentare sono scelti dai gruppi, ma ci sono alcuni vincoli OBBLIGATORI:
 - Almeno un esercizio per ogni componente del gruppo
 - Ogni esercizio deve considerare un ambito diverso di comprensione del linguaggio di programmazione (es. lettura del codice, l'ordine delle istruzioni, capacità di individuare gli errori, scrittura del codice di esercizi, uso e comprensione del polimorfismo e dell'ereditarietà, valutazione del codice)
 - In ogni esercizio è formato da almeno 3 esercizi diversi per grado di difficoltà (gli esercizi per lo stesso grado sono simili tra loro, es. stessa struttura ma valori diversi)

Proposta Esercizi

- Per ogni componente del gruppo sarà presente il nome seguito da:
 - titolo esercizio
 - [prompt di confronto con ChatGPT] (vedi prossime slide)
 - motivazione dell'importanza di svolgimento dell'esercizio
 - abilità esercitata
 - descrizione delle regole di esecuzione dell'esercizio
 - perché svolgendo l'esercizio si apprende tale abilità
 - cosa distingue i diversi gradi di difficoltà
 - come si creeranno i diversi esercizi dello stesso tipo di pari grado? Almeno 3.
- Obbligatorio includere almeno 3 gradi di difficoltà per ogni esercizio. (Possono essere anche di più)

CONSEGNA: Proposta Esercizi

NOME GRUPPO

COGNOME E NOME PRIMO COMPONENTE

ESERCIZIO “TROVA L'ERRORE”

[Prompt chatgpt o altro (link di riferimento)]

Motivazione formativa (perché ho scelto questo esercizio, come penso possa essere utile, ecc...)

ABILITA' esercitata (es. lettura del codice, ordine le istruzioni, conoscenza della semantica delle istruzioni, scrittura del codice, ecc...)

REGOLE dell'esercizio (es. durante l'esercizio si leggeranno 4 pezzi di codice errato, bisogna individuare i punti dove questo va in errore.).

MOTIVAZIONE: Svolgendo questo esercizio quale abilità si esercita? Perché?

Un esercizio è più difficile di un altro? qual è il criterio per valutare la difficoltà?

Come si creano i diversi esercizi di pari grado di difficoltà?

COGNOME E NOME SECONDO COMPONENTE

...

Uso di ChatGPT

Gli esercizi devono essere progettati dal gruppo anche con l'aiuto di ChatGPT o servizi simili

In questo caso, nella consegna va indicato il prompt usato per poter individuare tipi di esercizi, gli esercizi, le abilità da sviluppare, ecc..

Q) devo creare degli esercizi per migliorare le abilità del pensiero computazionale, hai qualche idea da suggerirmi?

A).....

Q)devo creare degli esercizi per migliorare le abilità di scrittura, lettura e comprensione del codice, hai qualche idea da suggerirmi?

A)....

...

Link:

<https://chat.openai.com/share/daed6b7c-28a4-421d-b519-f4d8a911feff>

Gestione base di dati e file

- Play memorizza esercizi, risultati parziali e utenti sul filesystem
- Ogni gruppo sceglie il formato dati da usare e le librerie per fare il parsing
- Necessari ovviamente controlli di consistenza sui dati
- Ogni gruppo sceglie la strategia che preferisce per gestire gli errori purché ben documentata e consistente
- NOTA: anche questa è una semplificazione, per un problema reale avremmo usato un database
- [Opzionale] E' ammesso usare un database

DOMANDE?

Contatti: Sara Zuppiroli <sara.zuppiroli@unibo.it>

