

ARCHIVIO: una collezione di funzioni per la gestione della carriera degli studenti

Alessandro Bocci, Susanna Pelagatti

Terzo assegnamento in itinere 622AA
AA 2023-24

Indice

1	Introduzione	1
1.1	Materiale in linea	1
1.2	Struttura degli assegnamenti, bonus, tempi di consegna e prova orale	1
1.3	Consegna degli assegnamenti	2
1.4	Valutazione dell'assegnamento	2
2	L'assegnamento ARCHIVIO	2
2.1	L'interfaccia grafica	3
2.2	Istruzioni	3

1 Introduzione

Il primo modulo (9 crediti) del corso di Programmazione e Analisi Dati (622AA) per la parte Python prevede lo svolgimento di due assegnamenti in itinere che esonerano dallo svolgimento del progetto finale. Questo documento descrive il primo assegnamento che prevede la realizzazione di un insieme di classi, metodi e funzioni Python.

Il software viene sviluppato e documentato utilizzando gli strumenti, le tecniche e le convenzioni presentati durante il corso.

1.1 Materiale in linea

Tutto il materiale relativo al corso può essere reperito dal Moodle ufficiale del corso. Eventuali chiarimenti possono essere richiesti ai docenti per posta elettronica.

1.2 Struttura degli assegnamenti, bonus, tempi di consegna e prova orale

I due assegnamenti Python possono essere svolti individualmente o in gruppi di 2 studenti e possono essere consegnati entro la data e l'ora dell'appello di Febbraio 2024. Agli studenti che consegnano una realizzazione sufficiente entro

una certa data pubblicata su Moodle vengono assegnati 2 punti di Bonus nella valutazione complessiva di ciascun assegnamento. La valutazione complessiva viene data dalla media delle valutazioni dei due assegnamenti e costituisce la base per la prova orale. La prova orale sarà composta di due parti. La prima parte sarà una discussione sugli assegnamenti presentati e tenderà a stabilire se lo studente è realmente l'autore di quanto consegnato (in caso di dubbi la valutazione verrà opportunamente aggiustata) in particolare verrà chiesto di leggere e modificare il codice e di spiegare quanto usato (in particolare costrutti e moduli non facenti parte del programma del corso). La seconda parte verterà su tutto il programma del corso. In particolare, l'orale comprenderà:

- una discussione delle scelte implementative
- l'impostazione e la scrittura di semplici programmi Scratch e Python (sequenziali e GUI) di difficoltà medio bassa rispetto a quelli visti nelle esercitazioni in classe
- domande su tutto il programma presentato durante il corso.

Il voto finale sarà la media (0-30L) fra la valutazione effettiva del progetto (emendata in caso di dubbi) e la prova orale.

1.3 Consegna degli assegnamenti

La consegna degli assegnamenti avviene secondo le istruzioni presenti nel `README.txt` di ciascun assegnamento.

1.4 Valutazione dell'assegnamento

All'assegnamento viene assegnata una fascia di valutazione da 0 a 30 che tiene conto dei seguenti fattori:

- motivazioni, originalità ed economicità delle scelte implementative
- strutturazione del codice (fattorizzazione del codice in funzioni, uso di strutture dati adeguate etc)
- efficienza e robustezza (numero di operazioni eseguite, fallimenti in caso di input inadeguati etc)
- aderenza alle specifiche
- qualità del codice Python e dei commenti

Tutti gli assegnamenti verranno confrontati automaticamente per verificare situazioni di plagio. Nel caso di elaborati uguali verranno presi provvedimenti per tutti i gruppi coinvolti.

2 L'assegnamento ARCHIVIO

L'assegnamento prevede la realizzazione di un archivio studenti simile (ma non uguale!) a quello del secondo assegnamento, attraverso la definizione di due classi (Archivio e Studente) e di una interfaccia grafica per attivare alcuni metodi del catalogo e per visualizzare i libri in esso contenuti.

Ogni studente è costituito dai seguenti attributi (o variabili di istanza):

- `cognome`, di tipo stringa;
- `nome`, di tipo stringa;
- `matricola`, di tipo intero positivo;
- `listaesami`, di tipo lista contenente una coppia (`codice`, `voto`), dove il codice è di tipo stringa mentre il voto è di tipo intero compreso fra 18 e 33 (inclusi).

La classe `Studente`, contiene almeno i metodi specificati nel file `archivio.py`.

La classe `Archivio` implementa il catalogo attraverso un dizionario, usa il campo `matricola` come chiave e come valore una coppia (`studente`, `note`) dove `studente` è un oggetto di tipo `Studente` e `note` è una stringa. La classe `archivio` deve contenere almeno i metodi specificati nel file `archivio.py`.

2.1 L'interfaccia grafica

Deve essere realizzata un'interfaccia grafica usando il modulo `tkinter` che permetta almeno di

- visualizzare il contenuto del archivio,
- inserire un nuovo studente nell'archivio,
- modificare i dati di uno studente,
- cancellare uno studente dall'archivio,
- calcolare la media dei voti di uno studente,
- caricare l'archivio da file,
- salvare l'archivio su file,
- uscire dall'applicazione.

Utilizzare il file `gui.py` per implementare l'interfaccia grafica, con gli `import` inseriti è possibile sfruttare le classi definite nel file `archivio.py` (se contenuto nella stessa cartella).

L'aspetto dell'interfaccia viene deciso dallo studente.

2.2 Istruzioni

Leggere attentamente il file `README.txt` ed analizzare i file forniti prima di iniziare ad implementare. Gestire accuratamente le eccezioni ovunque se ne possano generare (ricordate le tre regole viste a lezione!).

Ricordiamo l'importanza di analizzare i vari casi prima di iniziare a scrivere codice e di effettuare test incrementali sul codice durante lo sviluppo dei vari metodi.

Le parti opzionali devono essere corredate da test appropriati e documentate da commenti chiari o (in caso sia necessario) da un breve documento descrittivo che può essere consegnato insieme al codice e che spiega le motivazioni e la struttura di quanto realizzato.