

# HOTEL: simulazione di un sistema di prenotazioni di un hotel.

Alessandro Bocci, Susanna Pelagatti

Progetto 622AA  
AA 2024-25

## Contents

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Materiale in linea . . . . .	2
1.2	Struttura del progetto e prova orale . . . . .	2
1.3	Consegna del progetto . . . . .	2
1.4	Valutazione del progetto . . . . .	2
1.5	Istruzioni . . . . .	3
<b>2</b>	<b>Il progetto HOTEL: parte Scratch</b>	<b>3</b>
<b>3</b>	<b>Il progetto HOTEL: parte Python</b>	<b>3</b>
3.1	Classi di supporto . . . . .	4
3.2	Stanze dell'hotel . . . . .	4
3.3	Hotel . . . . .	4
3.4	Interfaccia grafica . . . . .	4
<b>4</b>	<b>Conclusioni</b>	<b>5</b>

## 1 Introduzione

L'esame del primo modulo (9 crediti) del corso di Programmazione e Analisi Dati (622AA) prevede lo svolgimento di un progetto didattico e di una prova orale come dettagliato nel programma ufficiale del corso. Questo documento descrive le specifiche del progetto didattico.

Il progetto è costituito da due parti: una parte relativa a Scratch (Sez. 2) e una parte relativa a Python (Sez. 3).

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

Deve inoltre essere consegnata una Relazione che contiene

- la spiegazione della parte Scratch (spirite, logica, interazione),
- le motivazioni delle scelte implementative e
- una guida utente che spiega come utilizzare il software.

## 1.1 Materiale in linea

Tutto il materiale relativo al corso è reperibile dal Moodle ufficiale del corso.

Eventuali chiarimenti possono essere richiesti ai docenti per posta elettronica o a ricevimento.

## 1.2 Struttura del progetto e prova orale

Il progetto può esser svolto individualmente o in gruppi di 2 studenti e deve essere consegnato entro la data e l'ora dell'appello ordinario in cui si desidera svolgere la prova orale. Lo studente è ammesso alla prova orale se la valutazione del progetto è sufficiente. La prova orale va svolta obbligatoriamente nell'appello nel quale si consegna il progetto. La data dell'orale verrà comunicata in seguito alla chiusura delle iscrizioni dell'appello scelto.

La prova orale tenderà a stabilire se lo studente è realmente l'autore del software consegnato, verterà su tutto il programma del corso e su tutto quanto usato nel codice anche se non fa parte del programma del corso. In particolare, l'orale comprenderà:

- una discussione delle scelte implementative;
- l'impostazione e la scrittura di semplici programmi Scratch e Python;
- domande legate al programma presentato durante il corso.

## 1.3 Consegna del progetto

La consegna del progetto avviene sul Moodle del corso, sotto forma di archivio che comprende:

- i file Python realizzati,
- il file `hotel.sb3` del progetto Scratch realizzato, e
- la Relazione in formato PDF.

## 1.4 Valutazione del progetto

Al progetto viene assegnata una fascia di valutazione da 0 a 30 e lode che tiene conto dei seguenti fattori:

- aderenza alle specifiche,
- strutturazione e riuso del codice (fattorizzazione del codice in funzioni/blocchi, uso di strutture dati adeguate etc),
- efficienza e robustezza (numero di iterazioni, numero di operazioni eseguite, fallimenti in caso di input inadeguati etc),
- qualità del codice (Scratch e Python) e dei commenti.

È possibile utilizzare funzioni e pacchetti non introdotti a lezione. In sede di orale verrà valutata approfonditamente la conoscenza dei candidati di tali strumenti.

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

## 1.5 Istruzioni

Leggere attentamente le specifiche in questo file, i file di **README** 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, documentate da commenti chiari e descritte accuratamente nella Relazione (solo parte Scratch).

La parte Python è corredata da una serie di test automatici eseguibili dal file *main.py* supportato dal file *testMy.py* che deve essere nella stessa cartella. Leggere il file *main.py* per capire quali sono i test effettuati e l'indice dei test falliti.

La consegna potrà avvenire **solo dopo aver superato tutti i test forniti**. Passare tutti i test non garantisce che il progetto sia totalmente corretto. Non passare tutti i test implica degli errori che pregiudicano un voto positivo al progetto.

## 2 Il progetto HOTEL: parte Scratch

Partendo dal cartoon “Hotel Transilvania”<sup>1</sup> realizzare una storia interattiva Scratch nella quale sono presenti almeno 2 punti di scelta da parte dell'utente (un totale di 4 finali diversi). E' possibile semplificare la storia originale mantenendo alcuni dei personaggi. Consegnare il file (formato .sb3) che realizza la storia e una relazione di massimo 10 pagine che contenga la struttura della storia (evidenziando i punti di scelta) e la descrizione dell'organizzazione del progetto con le motivazioni delle scelte implementative.

La relazione e il file .sb3 devono contenere nome, cognome e matricola (o telefono) dei componenti del gruppo.

## 3 Il progetto HOTEL: parte Python

La parte Python prevede la realizzazione di un software che simula la gestione delle stanze di un hotel e delle loro prenotazioni da parte dei clienti dell'hotel.

In particolare, il progetto è diviso in quattro parti, ognuna da implementare in file .py diversi contenenti le specifiche dettagliate da implementare:

- classi di supporto, file *classi.py*
- gerarchia di classi per le stanze, file *stanze.py*;
- implementazione dell'hotel, file *hotel.py*;
- interfaccia grafica, file *gui.py*.

Leggere attentamente le specifiche inserite nei file forniti per l'implementazione delle classi e dei metodi richiesti.

---

<sup>1</sup>[https://it.wikipedia.org/wiki/Hotel\\_Transylvania](https://it.wikipedia.org/wiki/Hotel_Transylvania)

### 3.1 Classi di supporto

Per realizzare il progetto dovranno essere create due classi di supporto utilizzate nel resto del codice: *Data* e *Prenotazione*.

La classe *Data* rappresenta una data specifica con giorno e mese. Dovrà esserci particolare cura nel rappresentare correttamente le date (es: evitare 30 di febbraio), il confronto fra date diverse (sia di uguaglianza che di maggiore e minore), e soprattutto il calcolo del numero di notti che separano due date (utilizzando l'operatore di differenza come per i numeri). La gestione degli anni bisestili non è richiesta.

La classe *Prenotazione* rappresenta una prenotazione per l'hotel, essa contiene e gestisce tutte le informazioni necessarie per una prenotazione: identificatore di prenotazione, numero della stanza prenotata, date di arrivo e di partenza, nome del cliente e numero di persone da ospitare. La data di arrivo non deve essere successiva alla data di partenza. Non si considerano date a cavallo della fine dell'anno, ad esempio: arrivo 31/12 partenza 1/1.

### 3.2 Stanze dell'hotel

L'hotel ha tre tipi di stanze: *Singole*, *Doppie* e *Suite*. Esse sono tre classi che estendono la classe *Stanza* che mette a comune lo stato per le stanze: numero di stanza, posti della stanza e prezzo per notte della stanza; inoltre, mette a comune i metodi principali delle stanze: getter, setter, rappresentazione a stringa, uguaglianza profonda e calcolo del prezzo di una stanza dato il numero di notti prenotate.

Le stanze singole hanno la peculiarità di avere un solo posto.

Le stanze doppie hanno la peculiarità di avere due posti letto e un prezzo per notte il 20% più alto del prezzo base.

Le suite hanno la peculiarità di avere almeno 4 posti letto, degli extra e un prezzo per notte il 50% più alto del prezzo base con un aggiunta di 10 euro per ogni extra.

### 3.3 Hotel

Un hotel memorizza in due dizionari le stanze e le prenotazioni dell'hotel. La classe *Hotel* fornisce metodi per la creazione, la modifica, la lettura e il salvataggio/caricamento su file di un hotel. Una stanza non può avere prenotazioni che si sovrappongono per lo stesso intervallo di date, comprese le date di arrivo e di partenza. Il prezzo di una prenotazione viene calcolato moltiplicando il numero di persone da ospitare per il prezzo della stanza dato il numero di notti prenotate. I dettagli dei metodi da implementare sono descritti nel file *hotel.py*.

### 3.4 Interfaccia grafica

L'interfaccia grafica deve essere implementata nel file *gui.py*. All'avvio dell'interfaccia grafica devono essere caricate le stanze dal file *hotel\_base.txt* generato da un'esecuzione senza errori o test falliti di *main.py*.

L'interfaccia grafica dovrà mostrare sin da subito le stanze dell'hotel indicando tipo e numero della stanza (con extra in caso di suite). Attraverso l'interfaccia grafica devono essere gestite le prenotazioni dell'hotel fornendo le seguenti funzionalità:

- mostrare la lista di prenotazioni dell'hotel;
- effettuare la prenotazione di una stanza;

- disdire la prenotazione di una stanza;
- ottenere il prezzo di una prenotazione dato l'indice;
- mostrare l'hotel ad una certa data inserendo i nomi dei clienti nelle stanze a quella data;
- ottenere le stanze libere ad una certa data;
- ottenere le prenotazioni di uno specifico cliente inserendo il nome;
- ottenere il numero di persone nell'albergo ad una certa data;
- salvare o caricare da file lo stato dell'hotel permettendo di inserire il nome del file;
- uscire dall'applicazione (per uscire dall'applicazione deve essere possibile usare sia mouse che tastiera).

L'aspetto generale dell'interfaccia grafica è lasciato alla creatività dello studente. La valutazione dell'interfaccia grafica è principalmente sull'usabilità dal punto di vista di un utente, sulla strutturazione del codice e sull'corretto utilizzo di eccezioni e metodi implementati nella classe Hotel.

## 4 Conclusioni

Riassumendo, il progetto prevede la realizzazione e la consegna dei seguenti file:

- il file *hotel.sb3* contenente il progetto Scratch;
- i file *classi.py*, *stanze.py*, *hotel.py* e *gui.py*;
- la Relazione formato PDF.