

# Corso di Laboratorio di Programmazione I

A.A. 2014/2015

## Progetto

### Regole per lo svolgimento:

- Il progetto deve essere svolto singolarmente o in gruppi di al massimo 2 persone.
- Il progetto deve essere consegnato **almeno 5 giorni prima** della prova scritta della parte di Programmazione I relativa all'appello in cui si desidera registrare l'esame.
- NON è permesso utilizzare nessuna libreria del linguaggio Java.
- Il progetto deve contenere i metodi richiesti **rispettandone esattamente il nome, il tipo e l'ordine dei parametri formali, ed il tipo di ritorno.**
- Si è naturalmente liberi di sviluppare metodi aggiuntivi di supporto ai metodi richiesti, laddove lo si ritenga utile.
- La consegna (del file Prenotazioni.java) va effettuata all'indirizzo [luca.moscardelli@unich.it](mailto:luca.moscardelli@unich.it) mettendo come oggetto **[jmm] consegna progetto laboratorio 2015** e nel testo dell'email il/i nome/i dello/degli studente/i che consegna/no.

### Testo del progetto:

Sviluppare in **Java** (non in Java--) una libreria per la gestione di prenotazioni di camere di un albergo in una stagione dell'anno.

Le prenotazioni devono essere memorizzate in un array bidimensionale di numeri interi, in cui la prima dimensione denota le camere dell'albergo, e la seconda i giorni della stagione di interesse.

Il valore **m[i][j]** contenuto nell'array bidimensionale m deve essere:

- **-1** nel caso in cui la stanza numero **i** ( $i=0, \dots, m.length-1$ ) sia libera nel giorno **j**;
- **Un intero  $n \geq 0$**  nel caso in cui la stanza **i** sia occupata nel giorno **j** dalla prenotazione **n**.

La libreria deve essere composta dai seguenti metodi, tutti inclusi nella stessa classe **Prenotazioni** salvata nel file **Prenotazioni.java**:

1. `static int[][] generaTabella (int nStanze, int nGiorni)`  
prende come parametri il numero di stanze ed il numero di giorni da gestire e restituisce un array bidimensionale di interi delle opportune dimensioni, che poi verrà utilizzato per gestire le prenotazioni.  
L'array bidimensionale deve essere inizializzato in modo che tutti i suoi elementi siano **-1**.

2. `static boolean libera (int[][] tabella, int nStanza, int giornoInizio, int giornoFine)`  
prende come parametri una tabella di prenotazioni, il numero della stanza, il giorno di inizio e il giorno di fine di un periodo di interesse e restituisce true se e solo se la stanza nStanza è sempre libera nel periodo di interesse (estremi inclusi).  
Se i giorni di interesse e/o la tabella non contiene dati validi, viene restituito false.
3. `static int stanzaLibera (int[][] tabella, int giornoInizio, int giornoFine)`  
prende come parametri una tabella di prenotazioni, il giorno di inizio e il giorno di fine di un periodo di interesse e restituisce l'indice di una qualsiasi stanza libera nel periodo di interesse (estremi inclusi).  
Se non esiste nessuna stanza libera nel periodo di interesse, viene restituito -1.  
Se i giorni di interesse e/o la tabella non contiene dati validi, viene restituito -1.
4. `static boolean stanzeLibere (int[][] tabella, int nStanze, int giornoInizio, int giornoFine)`  
prende come parametri una tabella di prenotazioni, il giorno di inizio e il giorno di fine di un periodo di interesse ed il numero di stanze richieste, e restituisce true se e solo se esistono nStanze libere nel periodo di interesse (estremi inclusi).  
Se i giorni di interesse e/o la tabella e/o nStanze non contiene dati validi, viene restituito **false**.
5. `static int prenota (int[][] tabella, int nStanza, int giornoInizio, int giornoFine)`  
prende come parametri una tabella di prenotazioni, il numero della stanza, il giorno di inizio e il giorno di fine di un periodo di interesse, effettua se la stanza richiesta risulta libera nel dato periodo (estremi inclusi) la prenotazione richiesta assegnandole un numero che sia univoco (non deve essere uguale a nessun altro numero presente in tabella). Il metodo restituisce il numero di prenotazione assegnato, oppure -1 nel caso in cui non sia stato possibile effettuare la prenotazione.  
Se i giorni di interesse e/o nStanza e/o la tabella non contiene dati validi, viene restituito -1.
6. `static int[] prenotaBlocco (int[][] tabella, int nStanze, int giornoInizio, int giornoFine)`  
prende come parametri una tabella di prenotazioni, numero di stanze da Prenotare, il giorno di inizio e il giorno di fine di un periodo di interesse, effettua se esistono nStanze libere nel dato periodo (estremi inclusi) la prenotazione richiesta assegnandole un numero che sia univoco (non deve essere uguale a nessun altro numero presente in tabella).  
Il metodo restituisce un array di lunghezza **nStanze + 1** in cui il primo elemento è il numero di prenotazione assegnato, ed i successivi elementi i numeri di stanza assegnati.  
Nel caso in cui non sia stato possibile effettuare la prenotazione, oppure se i giorni di interesse e/o nStanze e/o la tabella non contiene dati validi, viene restituito **null**.
7. `static void cancellaPrenotazione (int[][] tabella, int nPrenotazione)`  
prende come parametri una tabella di prenotazioni e il numero di una prenotazione, e cancella (se è presente) la prenotazione dalla tabella.
8. `static int inizio (int[][] tabella, int nPrenotazione)`  
prende come parametri una tabella di prenotazioni e il numero di una prenotazione, restituisce il giorno di inizio della prenotazione.  
Se la prenotazione non è presente, e/o la tabella non contiene dati validi, viene restituito -1.

9. `static int fine (int[][][] tabella, int nPrenotazione)`  
prende come parametri una tabella di prenotazioni e il numero di una prenotazione, restituisce il giorno di fine della prenotazione.  
Se la prenotazione non è presente, e/o la tabella non contiene dati validi, viene restituito **-1**.
10. `static int durata (int[][][] tabella, int nPrenotazione)`  
prende come parametri una tabella di prenotazioni e il numero di una prenotazione, restituisce la durata della prenotazione.  
Se la prenotazione non è presente, e/o la tabella non contiene dati validi, viene restituito **0**.
11. `static boolean allungaPrenotazione (int[][][] tabella, int nPrenotazione, int nGiorni)`  
prende come parametri una tabella di prenotazioni, il numero di una prenotazione ed il numero di giorni di cui allungare la prenotazione, e allunga **se possibile** la prenotazione.  
Se  $nGiorni < 0$ , la prenotazione viene accorciata di  $-nGiorni$  giorni. Se  $-nGiorni$  è maggiore o uguale della durata della prenotazione, la prenotazione viene cancellata.  
Nel caso in cui non sia stato possibile allungare la prenotazione viene restituito **false**, altrimenti **true**.  
Se la tabella e/o nPrenotazione non contiene dati validi, viene restituito **false**.
12. `static boolean anticipaPrenotazione (int[][][] tabella, int nPrenotazione, int nGiorni)`  
prende come parametri una tabella di prenotazioni, il numero di una prenotazione ed il numero di giorni di cui anticipare la prenotazione, e anticipa **se possibile** la prenotazione.  
Se  $nGiorni < 0$ , la prenotazione viene posticipata di  $-nGiorni$  giorni. Se  $-nGiorni$  è maggiore o uguale della durata della prenotazione, la prenotazione viene cancellata.  
Nel caso in cui non sia stato possibile anticipare la prenotazione viene restituito **false**, altrimenti **true**.  
Se la tabella e/o nPrenotazione non contiene dati validi, viene restituito **false**.
13. `static Blocco prenotaBlocco2 (int[][][] tabella, int nStanze, int giornoInizio, int giornoFine)`  
Si consideri il seguente tipo riferimento definito dall'utente:  

```
class Blocco{
    int nPrenotazione;
    int[] stanze;
}
```

  
Si scriva il metodo `prenotaBlocco2` che differisce da `prenotaBlocco` solo per il tipo di ritorno: `prenotaBlocco2` deve restituire un Blocco (e non un array di interi), in cui nPrenotazione è il numero assegnato alla prenotazione, e stanze è un array (di lunghezza **nStanze**) contenente le stanze della prenotazione.

#### **NOTE:**

- Attenzione alla differenza di significato tra “numero della stanza” e “numero di stanze”
- Quando una prenotazione si riferisce a più stanze, le operazioni che la riguardano devono essere applicate a TUTTE le stanze.