

# Ostello

Si vuole modellare un ostello. L'ostello ha 50 letti in totale, distribuiti in 10 stanze da 5 letti ciascuna: ogni letto (e stanza) può essere libero o occupato.

All'inizio tutti i letti sono liberi.

Un letto in una stanza si può occupare (dicendo quale stanza e quale letto) solo se il letto non è occupato. Si può occupare un qualsiasi letto libero di una stanza. Una stanza è occupata se non ha letti liberi.

## Compito

In un file word annota tutto quello che fai. Aggiungi anche gli screenshot per i punti principali (ad esempio copertura ...). Per effettuare gli screenshots usa lo strumento di cattura di windows. Anche la documentazione sarà valutata per ogni singolo progetto (circa del 15%).

Sviluppa un progetto in eclipse per ognuna delle domande. Ogni progetto dovrà avere come nome il tuo COGNOME\_ES (con ES il tipo dell'esercizio – vedi titoli degli esercizi). Puoi anche andare in parallelo (ad esempio modello NUSMV e codice Java). Metti i progetti e il documento word in una directory con nome COGNOME\_NOME (che puoi usare come workspace). Alla fine crea un file zip e consegna quello.

Commenta per bene tutto il codice che scrivi.

## Java

In Java scrivi una classe **Ostello** che implementa l'ostello descritto sopra. Lo stato dei letti nelle stanze deve essere modellato con una **matrice di booleani** (le righe sono le stanze e le colonne i letti). Ha tre metodi:

```
public boolean checkin(int stanza, int letto)
```

che occupa un letto specifico in una stanza. Restituisce true se la prenotazione ha successo.

```
public boolean checkin(int stanza)
```

che occupa un letto libero qualsiasi in una stanza. Restituisce true se la prenotazione ha successo.

Per vedere se una stanza è libera definisci un metodo:

```
boolean libera(int stanza)
```

# 1. ES = TESTING

Scrivere l'implementazione e i casi di test con Junit. Scrivi una classe di test `OstelloTest` con i seguenti test:

1. un caso di test che mostri che è possibile trovare un letto libero ed occuparlo
2. un caso di test che mostri che non è possibile occupare un letto già occupato
3. un caso di test che mostri che non è possibile occupare un letto in una stanza occupata

Esegui il controllo della copertura (istruzioni, branch, decisioni, MCDC). Se il caso sopra non è sufficiente ad avere una copertura soddisfacente, aggiungi i casi di test necessari (in una classe separata per ogni copertura).

Metti la directory test con i test separata (oltre src).

Prova con Randoop a generare casi di test (con pochi casi di test) e prova a valutarne la copertura.

# 2. ES = JML

Scrivi i contratti JML. Cerca di scrivere sia le precondizioni, che le postcondizioni dei metodi. Cerca di spostare nelle precondizioni alcuni controlli che facevi all'inizio dei metodi.

Cerca di scrivere anche invarianti.

Prova i contratti JML con una classe main in cui chiami i diversi metodi.

Prova anche a modificare il codice e controlla che i contratti siano violati. Documenta bene le violazioni e le loro cause in commenti e nel file di documento.

# 3. ES = KEY Verifica dei programmi

ATTENZIONE. Inizia da una classe con contratti **vuoti** (true) e **aggiungi** mano a mano i contratti però sempre dimostrandone la loro correttezza – cioè le prove devono essere sempre chiuse **vere**. La valutazione dipenderà dal massimo contratto che riuscite a dimostrare vero (contratti non provati non verranno valutati).

Se la tua classe contiene dei metodi delle librerie (tipo `system.out.println`) eliminali. Anche `Random` va eliminata se l'hai usata.

# 4. ES = NUMSV

Specificare l'ostello utilizzando NuSMV e le sue proprietà usando LTL e/o CTL.

Limitati a 2 letti per 2 stanze (usa gli array)

Introduci un array di letti. Introduci anche un array di 2 booleani per indicare se la stanza i-esima è occupata.

Ad ogni istante il sistema può fare due operazioni:

- occupa letto: per occupare un letto a caso ma che sia libero
- libera letto: per liberare un letto a caso ma che sia occupato

Le due azioni modificano i letti e l'occupazione delle stanze in modo corretto.

Tra le proprietà desiderate (alcune potrebbero essere false) c'è:

- P1: una stanza non può mai essere occupata
- P2: un letto non può mai venire occupato se è già occupato
- P3: un letto una volta occupato non può diventare più libero

Provare se sono vere oppure no. Se sono false presentare un contro esempio (se viene prodotto) e modificarle in modo che siano vere.

Aggiungi anche altre proprietà che ti sembrano importanti e commenta.

Per ogni proprietà aggiungi una descrizione in Italiano di quello che vuol dire.

## **5. ES = FSM (MBT con le FSM)**

Scrivi una FSM. Limitati a 3 stanze con 3 letti. Prova a generare il modello con modelJunit.

Salva il disegno della macchina come immagine.

## **6. ES = COMB (Combinatorial testing)**

Rappresenta in combinatorial testing il sistema con lo stato dei letti e delle stanze (anche in questo caso prendi solo 3 letti e 3 stanze). Introduci anche gli opportuni vincoli.

Applica il combinatorial testing.

Prova a generare la copertura pairwise.