

# TESTING E VERIFICA DEL SOFTWARE

## ESERCITAZIONE 3 -

### Verifica di programmi con openJML

#### Esercizio 1

COPIA il codice JML dell'esercitazione precedente (ad esempio Account) e prova con ESC i contratti.

Rimuovi System.out e classi di prova con main.

#### Esercizio 2 (cartella esercizio 5)

Prova alcuni contratti semplici (come quelli visti a lezione)

Alcuni esempi:

```
//@ requires true;
/*@ ensures \result >= a && \result >=b &&
(\result == a || \result ==b); @*/
public int max(int a, int b) {
    if (a >= b)
        return a;
    else
        return b;
}
```

#### Esercizio 3 (cartella esercizio 6):

Considera il seguente problema.

Ho una classe che ha un campo array di 5 interi tutti inizializzati a 0.

Voglio provare che il costruttore inizializza correttamente il campo array.

Procedo per gradi:

- se metto anche semplicemente il contratto ensures array != null senza invarianti non riesco a provare nulla
- aggiungo un loop invariant che dice che array non è nullo
- aggiungo la post condizione che array abbia length == 5 e modifico il loop invariant in accordo
- aggiungo anche la condizione che gli elementi dell'array sono messi a 0 come desiderato
- se aggiungo normal\_behavior, devo anche dimostrare che l'indice è corretto

#### Esercizio 4:

Cerca di dimostrare i contratti non chiusi dell'esercizio 1

#### Esercizio 5 - te giugno 21

Dato il seguente programma in Java, che controlla se un array contiene un numero pari maggiore di b;

```

boolean check(int[] a, int b) {
    if (a != null && a.length > 0) {
        for (int x : a) {
            if (x > b && x % 2 == 0)
                return true;
        }
    }
    return false;
}

```

Scrivi i contratti e prova la correttezza. Puoi semplificare il codice se i contratti che metti rendono superflui alcuni controlli. Puoi riscrivere il codice a tuo piacimento per portare a termine la dimostrazione. Ricordati la sintassi dei quantificatori:

(\forall x <dominio>;<range\_valori>;<condizione> )

(\exists x <dominio>;<range\_valori>;<condizione> )

## Tema d'esame del 2025

Prendi il tema d'esame di JML e prova con ESC la correttezza dei contratti.

Io sono riuscito 3 su 4

▼ Silos

[VALID] getGrano() [1,059 z3\_4\_3]

[VALID] gru(int,int) [1,082 z3\_4\_3]

> [INVALID] nastro() [1,099 z3\_4\_3]

[VALID] Silos(int) [1,104 z3\_4\_3]