**!!!!!!!!!!!!FARE COPERTINAAAAAAAAAAAA!!!!!!!!!!!!**

Indice

Indice …1

Introduzione …2

Perché fare testing

Il nostro codice

Homework 1

convertRomanToDecimal

convertDecimalToRoman

isValidRomanNumber

areInterdipendent

isPalindrome

Homework 2

…

…

Capitolo 1 – Homework 1

Specification based test - – Approccio a 7 step

Tutti le classi di test vanno inserite e spiegate all’interno di “Specification based test”

Structural Test e Code Coverage

Analisi Code Coverage – Stato iniziale

Soluzioni trovate

Analisi Code Coverage – Esito finale

Capitolo 2 – Homework 2

Property based test

…

Conclusioni

DATI PER HOMEWORK

Sì, i test che hai scritto coprono diversi aspetti dell'implementazione interna della classe `RomanToDecimalConverter`, fornendo quindi una buona base per eseguire structural testing. Ecco come i tuoi test corrispondono alle definizioni di structural testing:

1. \*\*Code Coverage\*\*: I test che hai scritto per ogni metodo della classe `RomanToDecimalConverter` contribuiscono alla copertura del codice. Verificano il comportamento atteso dei metodi e assicurano che tutte le istruzioni e i rami del codice siano eseguiti.

2. \*\*Branch Coverage\*\*: I test coprono i vari rami di esecuzione all'interno dei metodi, come il caso in cui `value < prevValue` o `value >= prevValue` nel metodo `convertRomanToDecimal`, garantendo così una buona copertura dei rami.

3. \*\*Boundary Testing\*\*: I test includono input che coprono i limiti delle condizioni, come l'input vuoto, input negativo, input massimo consentito, ecc., garantendo che la classe gestisca correttamente questi casi particolari.

4. \*\*Edge Cases\*\*: Alcuni test affrontano casi particolari come l'input con caratteri non validi, input che genera numeri romani non validi, ecc., che sono importanti per garantire che l'implementazione sia robusta anche in situazioni anomale.

5. \*\*Path Testing\*\*: Sebbene i tuoi test non coprano esplicitamente tutti i percorsi di esecuzione all'interno dei metodi, attraverso la combinazione dei vari test è probabile che molti percorsi siano esplorati, soprattutto considerando le diverse combinazioni di input.

In conclusione, i tuoi test forniscono una buona base per eseguire structural testing sulla classe `RomanToDecimalConverter`, ma è sempre consigliabile aggiungere ulteriori test per coprire ancora più scenari e migliorare la solidità del codice.