

Titolo

Autori

12 luglio 2024

Indice

1	Specifiche Informali	3
2	Analisi e Specifica dei Requisiti	4
2.1	Analisi Nomi-Verbi	4
2.2	Revisione dei Requisiti	4
2.2.1	Requisiti Aggiuntivi	5
2.3	Glossario dei Termini	5
2.4	Classificazione dei Requisiti	6
2.4.1	Requisiti Funzionali	6
2.4.2	Requisiti sui Dati	6
2.4.3	Vincoli/Altri Requisiti	6
3	Modellazione dei Casi d'Uso	7
3.1	Attori e Casi d'Uso	7
3.2	Diagramma dei Casi d'Uso	8
3.3	Scenari	9
3.4	Diagramma delle Classi	14
3.5	Diagrammi di Sequenza	14
3.6	Verifica Completezza dei Requisiti	20
4	Piano di Test Funzionale	21
4.1	RegistraCliente	21
4.2	LoginUtente	23
4.3	CercaFarmaco	24
4.4	ModificaFarmaco	24
4.5	AggiungiFarmaco	25
4.6	EliminaFarmaco	26
4.7	RegistraConsegnaOrdineAcquisto	27
4.8	RitiraOrdine	28
4.9	GeneraOrdineAcquistoFarmacista	28
4.10	CreaOrdine	29
5	Progettazione	29
5.1	Diagramma delle classi	29
5.2	Modello Entity-Relationship del Database	30
5.3	Diagrammi di sequenza	30
6	Implementazione	31
7	Testing	32
7.1	Test strutturale	32
7.1.1	creaOrdine()	32

Capitolo 7

Testing

7.1 Test strutturale

7.1.1 creaOrdine()

Codice Java

```
1 public String creaOrdine(Map<Integer, Integer> farmaciQuantita) throws
  ↳ OrderCreationFailedException {
2     if (farmaciQuantita.isEmpty())
3         throw new OrderCreationFailedException("Ordine vuoto");
4
5     EntityCatalogo catalogo = EntityCatalogo.getInstance();
6     if (!catalogo.checkScorte(farmaciQuantita)) {
7         throw new OrderCreationFailedException("Ordine non creato per mancanza scorte");
8     }
9     EntityOrdine ordine = new EntityOrdine(this.getId());
10    try {
11        EntityOrdineAcquisto ordineAcquisto = new EntityOrdineAcquisto();
12        for (Map.Entry<Integer, Integer> entry : farmaciQuantita.entrySet()) {
13            int id = entry.getKey();
14            int quantita = entry.getValue();
15            EntityFarmaco farmaco = catalogo.cercaFarmacoById(id);
16            ordine.aggiungiOrdineFarmaco(farmaco, quantita);
17            int scorteResidue = catalogo.decrementaScorte(id, quantita);
18            if (scorteResidue == 0) {
19                ordineAcquisto.aggiungiOrdineAcquistoFarmaco(farmaco,
20                    ↳ EntityOrdineAcquisto.QUANTITA_ORDINE_DEFAULT);
21            }
22        }
23        if (!ordineAcquisto.getQuantitaFarmaci().isEmpty()) {
24            ordineAcquisto.salvaInDB();
25        }
26        ordine.salvaInDB();
27        storicoOrdini.add(ordine);
28        return ordine.getId();
29    } catch (FarmacoNotFoundException e) {
30        throw new OrderCreationFailedException("Errore creazione ordine, farmaco non
31            ↳ trovato");
32    } catch (DBException e) {
33        throw new OrderCreationFailedException(e.getMessage());
34    }
```

Control Flow Graph

Il numero di cammini linearmente indipendenti è detto *numero cicломatico* di McCabe, e può essere calcolato equivalentemente in uno dei modi seguenti. Sia G il grafo della funzione, allora risulta:

1. $V(G) = E - N + 2$ in cui $E = \# \text{archi in } G$, $N = \# \text{nodi in } G$
2. $V(G) = P + 1$ con $P = \# \text{predicati in } G$
3. $V(G) = R + 1$ con $R = \# \text{regioni chiuse in } G$

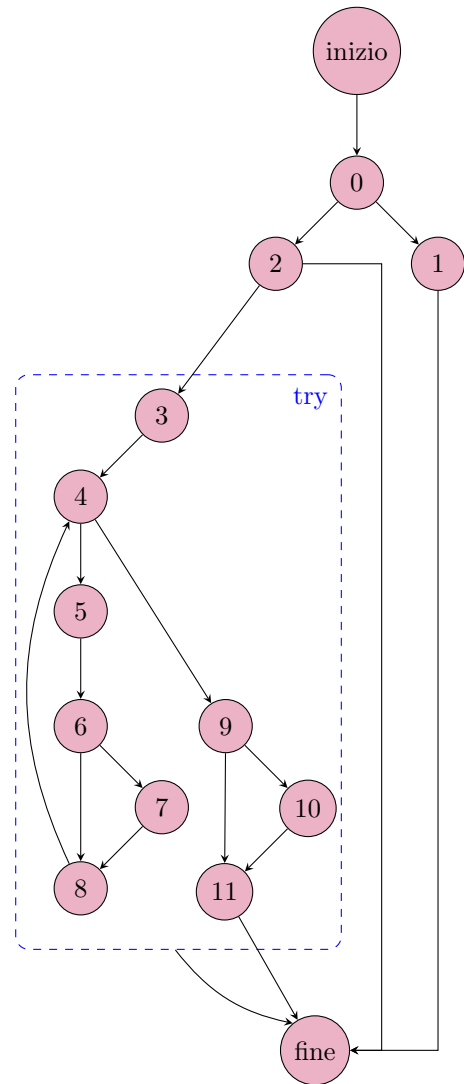
Nel nostro caso:

- $E = 16$
- $N = 12$
- $P = 5$
- $R = 5$

1. $V(G) = E - N + 2 = 16 - 12 + 2 = 6$
2. $P + 1 = 5 + 1 = 6$
3. $R + 1 = 5 + 1 = 6$

I cammini di base sono:

1. 0-1
2. 0-2
3. 0-2-3-4-9-11
4. 0-2-3-4-9-10-11
5. 0-2-3-4-5-6-8-4-9-11
6. 0-2-3-4-5-6-7-8-4-9-10-11



Test suite strutturale

Test Case ID	Descrizione	Cammino Coperto	Pre-condizioni	Input	Esito
1	Ordine vuoto	0-1	–	[]	Viene lanciata l'eccezione <code>OrderCreationFailedException("Ordine vuoto")</code>
2	Scorte insufficienti	0-2	Sono presenti 10 scorte del farmaco 'Tachipirina'	[(⟨idTachipirina⟩, 15)]	Viene lanciata l'eccezione <code>OrderCreationFailedException("Ordine non creato per mancanza scorte")</code>
3	Ordine contemporaneamente vuoto e non vuoto	0-2-3-4-9-11	–	[]	Cammino non percorribile
4	Ordine vuoto, viene generato un ordine d'acquisto	0-2-3-4-9-10-11	–	[]	Cammino non percorribile
5	Ordine creato, senza generare un ordine d'acquisto	0-2-3-4-5-6-8-4-9-11	Sono presenti 10 scorte del farmaco 'Tachipirina'	[(⟨idTachipirina⟩, 5)]	Viene creato un ordine
6	Ordine creato, si genera un ordine d'acquisto	0-2-3-4-5-6-7-8-4-9-10-11	Sono presenti 10 scorte del farmaco 'Tachipirina'	[(⟨idTachipirina⟩, 10)]	Ordine creato, parte una richiesta di fornitura per 50 'Tachipirina'

L'id dei farmaci viene scelto dal DB all'atto dell'aggiunta (indice autoincrementale). Pertanto, nel test l'id del farmaco viene ricavato con una funzione esterna.