



Progetti di Informatica III A

Università degli Studi di Bergamo
A.A. 2019/2020

PIFFARI MICHELE - 1040658

February 29, 2020

Contents

| | | |
|------------|--|-----------|
| I | Primo progettino | 1 |
| 1 | Cyclone | 3 |
| II | Secondo progettino | 5 |
| 2 | C++ | 7 |
| III | Terzo progettino | 9 |
| 3 | Energy drink vending machine con Scala | 11 |
| 3.1 | Descrizione del progetto | 11 |
| 3.2 | Gerarchia delle classi | 12 |
| IV | Quarto progettino | 15 |
| 4 | Coffe Machine con ASM | 17 |
| 4.1 | Descrizione del progetto | 17 |
| 4.2 | Macchina a stati | 17 |
| 4.3 | Standard Library | 17 |

List of Figures

| | | |
|-----|--------------------------------|----|
| 3.1 | Prodotti disponibili | 12 |
| 3.2 | Energy drink | 12 |
| 3.3 | Vending machine | 13 |
| 4.1 | Macchina a stati | 18 |

List of Tables

Part I

Primo progettino

1

Cyclone

Part II

Secondo progettino

2

C++

s

Part III

Terzo progettino

3

Energy drink vending machine con Scala

3.1 Descrizione del progetto

Il progetto scritto in Scala prevede di andare ad affiancare, alla macchinetta del caffè gestita in ASMETA, un distributore automatico di bevande energetiche. In particolar modo si è progettato un sistema con queste specifiche:

- Ogni distributore automatico può essere impostato per funzionare in una lingua piuttosto che in un’altro: nel codice è gestitata solamente la possibilità di introdurre distributori automatici in lingua italiana e in lingua inglese.
- Questi distributori possono gestire le seguenti bevande energetiche (*energy drink*):
 - RedBull
 - Monster
 - Gatorade
 - Italian

Ognuno di questi prodotti sarà caratterizzato dai seguenti campi descrittivi:

- prezzo
- volume (espresso in cl)
- data di scadenza
- insieme di tags, che permettono di esprimere le caratteristiche salienti di ognuno degli energy drink

Ogni distributore andrà ad offrire le seguenti funzionalità:

- Acquisto dei prodotti disponibili, regalando i prodotti scaduti: in particolare la macchinetta sarà in grado di fornire resto esatto al cliente (oppure tutta la somma di denaro inserita nel caso di prodotto scaduto).
- Mostrare l’elenco dei prodotti disponibili all’interno del distributore (identificato tramite ID), unitamente al numero di pezzi disponibili, come si vede in figura 3.1.
- Possibilità di cercare un prodotto tramite tag, per poter così trovare l’energy drink più adatto ad ogni evenienza
- Aggiunta di energy drink all’interno del distributore: nello specifico, l’aggiunta di un nuovo prodotto, avviene all’interno di una struttura definita come un array di Queue (ovvero una matrice), che non fa altro che andare a riprodurre la fisionomia di un distributore reale.

Available drinks for machine with id 100:

- 3 RED BULL
- 1 MONSTER
- 1 GATORADE
- 1 ITALIAN

Figure 3.1: Prodotti disponibili

3.2 Gerarchia delle classi

Nella applicazione realizzata sono stati realizzati due gerarchie facendo uso dei *trait*: i *trait* in Scala corrispondono alle interfacce in Java, ovvero permettono di definire la firma di ogni classe che ne implementa la struttura. Nel nostro caso abbiamo due strutture gerarchiche, gestite tramite traits, mostrate con un grafo ad "albero", nelle seguenti immagini (figura 3.2 3.3).

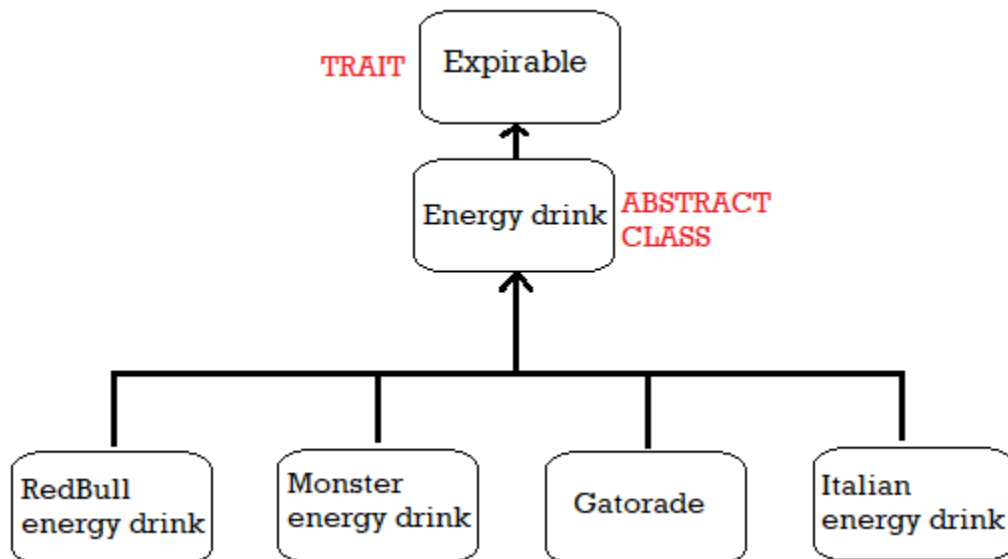


Figure 3.2: Energy drink

3.3 Filter

3.4 Match

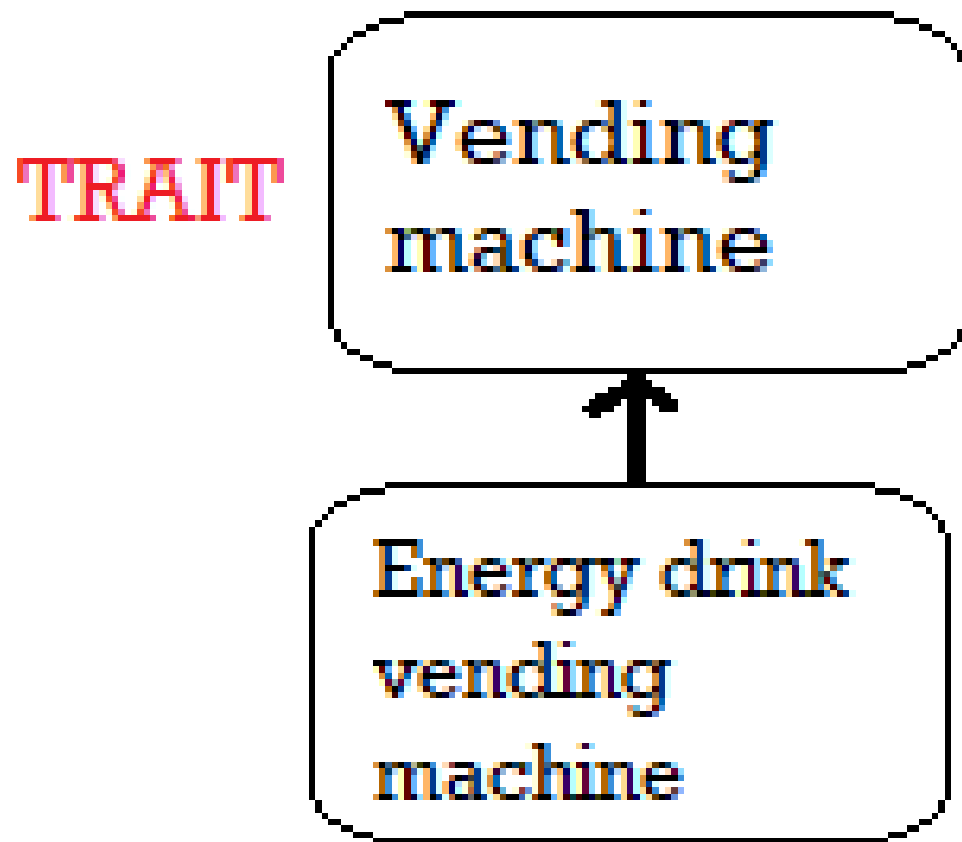


Figure 3.3: Vending machine

Part IV

Quarto progettino

4

Coffe Machine con ASM

4.1 Descrizione del progetto

Per quanto riguarda la parte di ASM *ASM* si è deciso di riprendere un esempio visto in classe (e durante le pause), ovvero quello relativo alla *Coffee machine*. Il distributore modellato può preparare diversi tipi di bevande (caffè, cappuccino etc), ognuna preparata con diverse quantità di ingredienti (acqua, caffè, latte etc) i quali vengono consumati e devono essere reintegrati dal manutentore.

Il distributore accetta pagamenti solamente in moneta tramite l’inserimento di denaro nell’apposita fessura.

Il distributore è in grado di fornire il resto (anche se non sempre in modo esatto).

Quando tutte le bevande sono esaurite, il distributore va fuori servizio, in attesa che gli ingredienti vengano aggiunti dal manutentore, il quale può inoltre prelevare o inserire monete dal distributore, sempre tenendo conto del vincolo di capacità del vano porta monete.

4.2 Macchina a stati

La ASM è basata su una sottostante macchina a stati finiti, mostrata in figura, che definisce i principali stati e transizioni del distributore. La ASM permette di estendere questa FSM introducendo un concetto aumentato di “stato”, che comprende anche funzioni dinamiche, modificando le quali si possono memorizzare informazioni aggiuntive.

In particolare, è stato possibile memorizzare informazioni su:

- Quantità di ingredienti residui
- Monete possedute dal distributore
- Credito dell’utente attuale

4.3 Standard Library

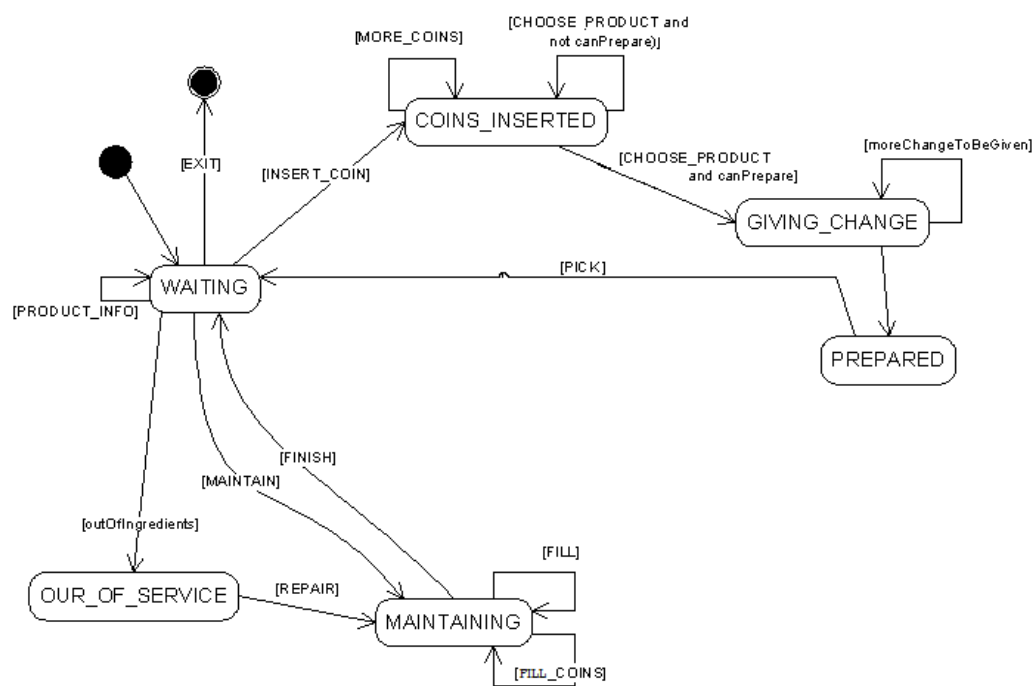


Figure 4.1: Macchina a stati