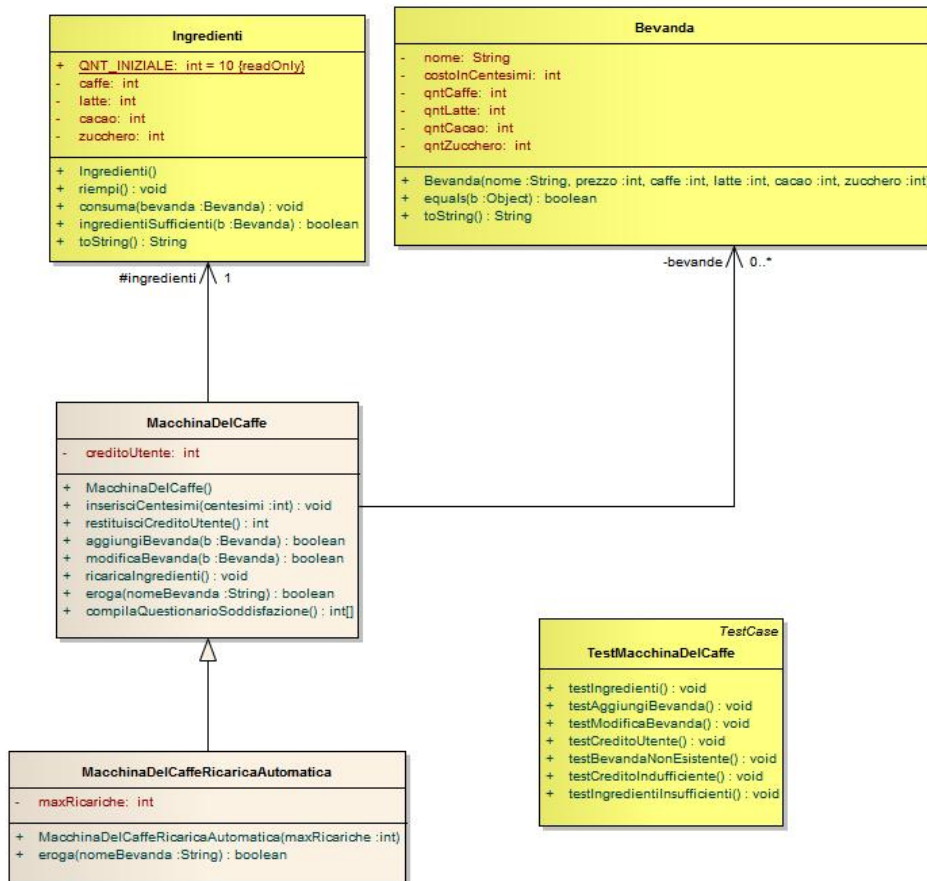


## Programmazione – Appello del 27 settembre 2012

### Testo parte pratica

La nota ditta produttrice del caffè “Javazza” sta finalizzando il progetto di un distributore automatico di bevande chiamato (su proposta di uno strapagato responsabile del Marketing) “macchina del caffè”. Il progetto del software di controllo è rappresentato dal seguente diagramma UML:



Lo studente completi l’implementazione delle classi descritte nel diagramma UML. Le classi rappresentate in giallo sono parzialmente implementate, le rimanenti sono da implementare per intero. Il diagramma dato è completo e non richiede ulteriori metodi ed attributi **ad eccezione dei metodi di incapsulamento** (set e/o get o nessuno, a seconda dei vari casi, da valutarsi con attenzione a cura dello studente).

Lo studente tenga conto delle seguenti specifiche.

La classe **Bevanda** codifica la quantità di caffè, latte, cacao e zucchero per preparare una bevanda erogabile dal distributore, e ne indica nome e prezzo (in centesimi di euro). La classe **Ingredienti** rappresenta la disponibilità di risorse (caffè, latte, cacao e zucchero) di un distributore automatico. La classe **MacchinaDelCaffe** rappresenta il controllore software del distributore automatico che permette di erogare un insieme di possibili bevande in considerazione della disponibilità di ingredienti e del credito dell’utente. Inoltre: la classe **TestMacchinaDelCaffe** permette di verificare il corretto funzionamento del sistema in alcuni casi interessanti; tale classe deve essere completata come descritto più avanti.

#### Classe Bevanda (fornita parzialmente)

- definisce la quantità di caffè, latte e cacao per preparare una bevanda erogabile, e ne indica nome e prezzo (in centesimi di euro)
- definisce un costruttore per inizializzare tutti i suoi attributi. NB: il costruttore solleva un’eccezione se invocato con valori impropri dei parametri
- effettua l’override del metodo `equals()` per realizzare la seguente definizione di uguaglianza fra bevande: “Due bevande sono uguali se e solo se hanno lo stesso nome”

- effettua l'override del metodo toString() restituendo una stringa che esprime il nome, il prezzo e la composizione (unità di caffè, latte e cacao) di una bevanda

### **Classe Ingredienti (fornita parzialmente)**

- definisce le quantità di ingredienti disponibili in un dato momento per preparare bevande;
- definisce un costruttore per inizializzare le quantità ad un valore iniziale di default definito dalla costante QNT\_INIZIALE;
- definisce il metodo riempi() che reinizializza tutte le quantità di ingredienti
- definisce il metodo ingredientiSufficienti(bevanda) per valutare se gli ingredienti disponibili sono sufficienti per preparare la detta bevanda
- definisce il metodo consuma(bevanda) che aggiorna le quantità di ingredienti dopo l'utilizzo degli stessi per preparare la detta
- effettua l'override del metodo toString() restituendo una stringa che esprime lo stato corrente degli ingredienti disponibili

### **Classe MacchinaDelCaffè**

- rappresenta un distributore automatico di bevande caratterizzato da una collezione di bevande erogabili (che non ammette duplicati, in base alla definizione di uguaglianza fra bevande), da una dotazione di ingredienti e da un credito (in centesimi di euro) inserito dall'utente. *[Suggerimento: per comodità potrebbe essere utile implementare un metodo con visibilità ristretta, cerca(bevanda): Bevanda, che permette di trovare una data bevanda nella collezione e se presente ne restituisce il reference.]*
- il costruttore senza parametri definisce una nuova macchina del caffè associata inizialmente ad una collezione vuota di bevande, ad una dotazione di ingredienti e con credito utente pari a zero
- il metodo inserisciCentesimi(centesimi) corrisponde all'inserimento di monete da parte dell'utente e incrementa il credito utente del numero di centesimi inseriti (se negativo, il metodo non fa nessuna modifica al credito utente)
- il metodo restituisciCreditoUtente() azzerà il credito utente, restituendo come valore di ritorno il numero di centesimi disponibili
- il metodo aggiungiBevanda(bevanda) inserisce un nuovo elemento nella collezione di bevande, a meno che l'elemento non sia già presente. Restituisce true se l'inserimento avviene con successo, false in caso contrario
- il metodo modificaBevanda(bevanda) modifica una bevanda esistente, che deve essere quindi già presente nella collezione. Restituisce true se la modifica avviene con successo, false in caso contrario
- il metodo ricaricaIngredienti() ricostituisce la disponibilità iniziale di ingredienti
- il metodo eroga(nomeBevanda) eroga la bevanda con il nome indicato. Se l'erogazione può avvenire (la bevanda prescelta è parte della collezione di bevande, il credito utente è sufficiente, e c'è sufficiente disponibilità di ingredienti) il metodo aggiorna la disponibilità di ingredienti e il credito utente in base a quanto erogato e restituisce true. Se il credito utente non è sufficiente o la bevanda indicata non esiste o la disponibilità di ingredienti non è sufficiente, il metodo restituisce false.
- Il metodo compilaQuestionarioSoddisfazione() funziona interattivamente e restituisce un array di interi che esprimono un voto dell'utente (da 1 a 5, o 0 nel caso di giudizio inespresso) per le bevande erogate dal distributore. Per ogni bevanda erogabile, il metodo chiede all'utente se vuole esprimere un giudizio. Se sì, viene chiesto il voto, reiterando la richiesta fino a che non viene inserito un voto valido (ovvero un valore fra 1 e 5). Alla fine il metodo stampa a video il valore del massimo, del minimo e della media aritmetica dei voti inseriti (senza contare i giudizi inespressi) e restituisce l'array con i voti.

### **Classe MacchinaDelCaffèRicaricaAutomatica**

- rappresenta uno speciale distributore automatico di bevande in grado di ricaricare automaticamente la propria disponibilità di ingredienti per un numero di volte prestabilito
- il costruttore definisce la macchina del caffè e inizializza il numero di volte che la ricarica automatica può essere effettuata
- il metodo eroga(nomeBevanda) effettua l'override del metodo corrispondente della superclasse, ed implementa il comportamento di ricarica automatica come segue. Il metodo eroga il prodotto richiesto. In caso di successo dell'erogazione, il metodo verifica se gli ingredienti disponibili sono ancora sufficienti per l'erogazione di un'altra bevanda uguale a quella appena erogata, e in caso di risultato negativo della verifica procede ad effettuare la ricarica, aggiornando il numero di ricariche che possono essere ulteriormente effettuate

Provare il sistema con i test di unità forniti (**classe TestMacchinaDelCaffè**) e implementare inoltre il seguente test aggiuntivo: Si crei una macchina del caffè e due bevande diverse, si aggiungano le bevande nella macchina del caffè, si inserisca credito sufficiente per entrambe e si provi ad erogarle entrambe, verificando il successo delle operazioni di erogazione.