PROGRAMMAZIONE II (A,B) - a.a. 2013-14
Prima Valutazione Intermedia — 31 marzo 2014

1. Una `CoinsCollection` è una collezione di monete del valore di 50 centesimi, 1 o 2 euro. Un valore tipico è

   [*v1**50C, *v2**1E, *v3**2E]

   dove *v1*, *v2* e *v3* sono valori interi maggiori di zero.

2. Fornire (scrivendo un'interfaccia Java) la specifica del tipo di dati astratto modificabile `CoinsCollection` con le seguenti operazioni:

   - `createCC()` crea una `CoinsCollection` vuota;
   - `addCoin(Integer m)` inserisce la moneta di valore `m` nella collezione;
   - `balance()` restituisce il valore totale della collezione di monete;
   - `makeChange(Integer v)` effettua l'operazione di "cambio" del valore `v` in monete presenti nella collezione: restituisce, se possibile, una `CoinsCollection` con un valore totale uguale a `v`, togliendo le monete corrispondenti da `this`.

   Fornire, oltre all'intestazione dei metodi, l'overview (basata sul punto 1.), le clausole REQUIRES, MODIFIES e EFFECTS di ogni metodo, indicando anche le eccezioni eventualmente lanciate e se sono checked o unchecked.

3. Supponendo di utilizzare come struttura di rappresentazione del TDA `CoinsCollection` una lista di `Integer` della forma

   ```
   private List<Integer> monete;
   // lista della monete presenti nella collezione
   ```

   definire la funzione di astrazione e l'invariante di rappresentazione.

4. Fornire l'implementazione del metodo `balance`.
   Fornire l'implementazione del metodo `makeChange` e dimostrare che preserva l'invariante di rappresentazione.

5. Supponiamo di aggiungere al TDA `CoinsCollection` il metodo `getCoins` così definito:

   ```
   public List<Integer> getCoins() { return newArrayList<Integer>(coins); }
   ```

   Sapendo che il costruttore `ArrayList<Integer>(Collection <Integer>)` restituisce una nuova lista in cui vengono copiati tutti gli oggetti della collezione passata come parametro, dire se l'introduzione di questo metodo determina l'esposizione della rappresentazione della classe, cioè se questo metodo può essere usato da un programmatore per falsificare l'invariante di rappresentazione di `CoinsCollection`.

6. Un ragionevole test per il metodo `makeChange` può essere costituito dalla creazione di una `CoinsCollection`, seguita dall'aggiunta di un certo numero di monete con `addCoin`, e da una o più invocazioni di `makeChange` con opportuni argomenti, seguite da controlli sulla collezione restituita o sull'eccezione eventualmente lanciata.

   Definire una batteria di test per il metodo `makeChange`, cioè un insieme di test sufficiente a convincersi che il metodo implementato soddisfi la specifica.

7. Si definisca il TDA `FloppyCoinsCollections` esattamente come `CoinsCollections`, tranne che per il metodo `makeChange` ch viene definito come segue:

   - `makeChange(Integer v)` restituisce una `CoinsCollection` con un valore totale uguale a `k`, `k` è il più piccolo tra `v` e `this.balance()`.

   Dire, motivando la risposta, se `FloppyCoinsCollections` è un sottotipo di `CoinsCollections`, cioè se è soddisfatto il principio di sostituzione.

Draft soluzione (possibile)

1. Overview: La classe rappresenta una collezione di monete da 1 centesimo (1C), 2 centesimi (2C), 5 centesimi (5C), 10 centesimi (10C), 20 centesimi (20C) e 50 centesimi (50C) typical values: [v1*1C, v2*2C, v3*5C, v4*10C, v5*20C, v6*50C]

2. Invariante di rappresentazione I(c) = !coins = null e ogni elemento della lista ha un valore che appartiene all'insieme 1, 2, 5, 10, 20, 50.

3. funzione di astrazione v1*1c: v1 e' il numero degli elementi della lista che hanno valore 1, simile per gli altri casi.

4. CoinsCollection makeChange(Integer v) throws IllegalValue
   Effects: throws IllegalValue se v ha un valore negativo
   Modifies this e restituisce una collezione di mentine con il valore pari a v.

   Una possibile alternativa e' dire che contiene il minor numero possibile di monete da 1C o 2C

5. implementazione da fare

6. No perche Integer e' non modificabile.

7. Dipende dalla scelta della specifica ma si deve fare un test per uk valore negativo e poi per un valore minore di 100.