

# Relazione per il progetto di Programmazione 2

Francesco Bocchi 550145 Corso B - secondo progetto intermedio

## 1 Introduzione

Il progetto richiede l'estensione dell'interprete didattico fornito dal docente, in modo tale da poter gestire una collezione di coppie  $\langle \text{chiave}, \text{valore} \rangle$  che rappresentano un dizionario di elementi non omogenei, dove la chiave è un identificatore.

## 2 Scelte progettuali

### 2.1 Sintassi concreta

Definiamo qui di seguito la sintassi concreta:

$$\begin{aligned} Dict &::= \varepsilon \mid (Ide, Exp); Dict \\ Select &::= (Dict, Ide) \\ Insert &::= (Dict, Ide, Exp) \\ Remove &::= (Dict, Ide) \\ Clear &::= Dict \\ ApplyOver &::= (Ide, Exp, Dict) \end{aligned}$$

### 2.2 Scelte progettuali

Gli operatori richiesti dal progetto sono stati implementati secondo il seguente criterio

**Dict** per la creazione di un nuovo dizionario. Dato che è stato specificato il fatto che la chiave dovesse essere un identificatore, un dizionario non è altro che una lista di coppie  $\langle ide, exp \rangle$ . La sua valutazione restituisce le coppie  $\langle ide, evT \rangle$ , con il significato che in un dizionario i tipi associati alle chiavi non sono omogenei.

**Select** che permette, attraverso un identificatore, di selezionare il valore di un elemento nel dizionario. La sua valutazione restituisce quindi un  $evT$ , ma nel caso in cui l'elemento corrispondente all'identificatore passato come parametro alla select non venga trovato, viene stampato un messaggio di errore.

**Insert** da cui si ottiene un nuovo dizionario (dato che questi sono immutabili) con una nuova coppia  $\langle ide, exp \rangle$  passata come parametro. Nel caso in cui in un dizionario sia già presente un elemento che corrisponda allo stesso identificatore passato come parametro, allora nel nuovo dizionario viene semplicemente sostituito il vecchio valore con il nuovo.

**Remove** attraverso la quale restituiamo un nuovo dizionario, eliminando l'associazione  $\langle ide, exp \rangle$  per cui l'identificatore corrisponda a quello passato come parametro. Nel caso in cui nel dizionario non venga trovato l'identificatore da rimuovere, allora viene restituito un nuovo dizionario identico.

**Clear** che applicata a qualunque dizionario, restituisce un nuovo dizionario vuoto.

**ApplyOver**<sup>1</sup> la quale applica un determinato tipo di funzione a "tutti" gli elementi del dizionario. A causa del fatto che un dizionario contiene elementi non omogenei, applicare una funzione a tutti gli elementi di un dizionario risulta complicato in quanto si deve gestire la coerenza dei tipi presi in ingresso dalla funzione e quelli presenti nel dizionario stesso: a questo scopo viene utilizzato il costrutto *try-with* che consente di catturare le varie eccezioni, tra le quali *type error*, ovvero l'eccezione dovuta ad errori di tipo nell'applicazione di funzioni. Dunque nel *try* si prova ad applicare la funzione passata come parametro al valore presente nel dizionario, se la valutazione va a buon fine si inserisce il valore calcolato dalla funzione nel dizionario, altrimenti il valore rimane inalterato.

## 2.3 Istruzioni per l'esecuzione

Nel file del progetto sono presenti una serie di dichiarazioni di espressioni seguite dalla loro valutazione, con un commento che riporta il risultato atteso da stampare. Si testano tutte le operazioni sia in casi semplici che in casi limite (elementi già presenti, rimozione da un dizionario vuoto, ...).

---

<sup>1</sup>Ogni operatore controlla che l'espressione passata come argomento sia un dizionario, ovvero un *Dict*. In caso contrario viene lanciato un messaggio di errore.