

**Relazione Giuseppe Giordano - N.M.: 598955**  
**Secondo Progetto-sessione invernale**  
**Anno Accademico 2019/2020**

Il seguente progetto si basa sulla progettazione e sviluppo di un interprete in Ocaml.

Risponde alla richiesta di estensione del linguaggio funzionale didattico e del relativo interprete al fine di permettere la manipolazione di dizionari.

Un dizionario è una collezione di coppie chiave-valore con chiave unica.

Per la creazione di tale interprete ho esteso la sintassi astratta, trattata a lezione, a cui ho aggiunto nuove Espressioni:

```
type exp=  
|Dizionario of (variable*exp) list (lista di coppie  
variabile-valore)  
|Insert of variable*exp*exp  
|Delete of exp*variable  
|Has key of variable*exp  
|Iterate of exp*exp  
|Fold of exp*exp  
|Filter of variable list*exp
```

alcune eccezioni:

```
exception ErroreGenerico of string;;  
exception ElementononTrovato;;  
exception ChiavePresente of string;;
```

ed un nuovo tipo:

```
type evT=  
|Dizionariovalues of (variable*evT)list
```

Per l'implementazione del dizionario ho utilizzato una lista di coppie variabile-valore ed ho utilizzato il tipo Dizionariovalues per convertire il tipo exp della coppia in evT; in questo modo l'eval può valutare l'espressione.

All'interno del dizionario non possono esistere elementi duplicati, infatti:

L'operazione insert inserisce gli elementi in coda nel dizionario e se l'elemento è già presente segnala l'eccezione "ChiavePresente", altrimenti inserisce.

Per verificare la correttezza del codice ho implementato all'interno del file dizionario.ml una serie di test per le varie operazioni.