

# Corso di Linguaggi Dinamici

## A.A. 2010/2011

### Esercitazione 2 – 9 maggio 2011

**Esercizio 1.** Si consideri il file di esempio *log.txt*. Tale file contiene l'output di un possibile meccanismo di logging per l'applicazione *zoo.py*, in formato CSV (comma separated values). Il formato del file è descritto dalle seguenti regole:

- il carattere di hash mark # introduce una riga commento che può essere ignorata;
- una riga non di commento contiene sei campi separati da un numero arbitrario di spazi bianchi;
- il primo campo contiene l'istante di un evento, assunto univoco (float);
- il secondo campo contiene il nome di un animale (stringa);
- il terzo campo contiene il nome della classe associata all'animale (stringa);
- il quarto campo contiene il nome dell'azione effettuata dall'animale (stringa);
- il quinto campo contiene un argomento generico (float).

Si richiede l'implementazione di una classe *Parser.py* in grado di effettuare il parsing del formato di record appena descritto. Il costruttore della classe *Parser.py* prende in ingresso il nome di un file e costruisce un dizionario multilivello D del tipo:

```
D[1.0][nome] = 'cavallo1'
D[1.0][classe] = 'Cavallo'
D[1.0][azione] = 'mangia'
D[1.0][argomento] = 1.0
```

Si implementi l'algoritmo di costruzione del dizionario D in un metodo separato dal costruttore.

**Tempo a disposizione: 45 minuti.**

**Esercizio 2.** Si estenda la classe *Parser.py* in modo tale da sollevare le opportune eccezione ed uscire dal programma in caso di oggetti non congrui (ad esempio: istante di un evento non in formato float).

**Tempo a disposizione: 30 minuti.**

**Esercizio 3.** Si renda la classe *Parser.py* eseguibile tramite uno script.

**Tempo a disposizione: 15 minuti.**