Simulazione Esame Programmazione 2 25/06/2024 – Tutorato Qualificato

Si consideri la classe virtuale `Animale` con i seguenti attributi privati:

- `nome` (stringa)
- `peso` (float)

La classe `Animale` deve avere i seguenti metodi pubblici:

- Costruttore che inizializza `nome` e `peso` come attributi privati.
- Metodi get.
- Metodo virtuale puro `verso()`: restituisce una stringa rappresentante il verso dell'animale.
- Overload dell'operatore << per stampare le informazioni dell'animale.

Creare due classi derivate da `Animale`:

- **`Gatto`**: Implementa il metodo `verso()` per restituire "Miao!".
- **`Cane`**: Implementa il metodo `verso()` per restituire "Bau!".

Si consideri una classe template `OrderedLinkedList` che rappresenta una lista concatenata ordinata di elementi di tipo T.

La classe ha solo la testa come attributo privato e i seguenti metodi pubblici:

- costruttore, distruttore
- inserisci: inserisce un elemento nella lista
- adotta: prende in input un puntatore a nodo e lo rimuove
- stampa: stampa tutti gli elementi tramite operatore <<

Si scriva un programma che prenda almeno 10 stringhe da console.

Per ogni stringa, crei dei cani o dei gatti ciascuno con una probabilità del 50% e assegni la stringa al nome. Inoltre, il peso viene generato casualmente tenendo conto del fatto che I gatti possono avere un peso casuale tra i 3 e i 5 kg, mentre i cani tra i 2 e i 20kg.

Inserire gli animali nella lista. Per ordinare gli animali sarà necessario implementare l'overload dell'operatore < che ordini per peso, dando precedenza ai gatti in caso di parità.

Si creino inoltre i seguenti metodi che prendono come input un puntatore a `OrderedLinkedList` di tipo puntatore ad `Animale`:

- gatti: stampa tutti i nomi dei gatti
- cani: stampa tutti i nomi dei cani

Fornire degli esempi per tutte le funzionalità implementate.