

Animali.h

mercoledì 28 giugno 2023 21:30

```
#include <iostream>
#include <string>
using namespace std;
#ifndef ANIMALI_H
#define ANIMALI_H

class Animale{
protected:
    string nome;
    string razza;
    int eta;
    virtual void Stampa(ostream &os) const = 0;
public:
    Animale(string _nome, string _razza, int _eta) :
        nome(_nome), razza(_razza), eta(_eta){};
    ~Animale(){};
    string getNome() const{ return this->nome;}
    string getRazza() const{return this->razza;}
    int getEta()const {return this->eta; }
    virtual string verso() const = 0;
    virtual void stampa() const = 0;
    friend ostream& operator<<(ostream& out, const Animale& a);
};

ostream& operator<<(ostream& os, const Animale& a){
    a.Stampa(os);
    return os;
}

class Cane : public Animale{
private:
    void Stampa(ostream &os) const;
public:
    Cane(string _nome, string _razza, int _eta)
        : Animale(_nome,_razza, _eta){};
    ~Cane(){};
    string verso() const{ return "Bau";};
    void stampa() const;
};

void Cane::stampa() const{
    cout << "nome: " << getNome() << " , razza: "
    << getRazza() << " , eta': " << getEta()
    << " , verso: " << verso();
}

void Cane::Stampa(ostream &os) const{
    os << "nome: " << getNome() << " , razza: "
    << getRazza() << " , eta': " << getEta()
    << " , verso: " << verso();
}

class Gatto : public Animale{
private:
    void Stampa(ostream &os) const;
public:
```

```

        Gatto(string _nome, string _razza, int _eta)
            : Animale( _nome, _razza, _eta){};
        ~Gatto(){};
        string verso() const { return "Miao";}
        void stampa() const;
};

void Gatto::stampa() const{
    cout << "nome: " << getNome() << " , razza: "
    << getRazza() << " , eta': " << getEta()
    << " , verso: " << verso();
}

void Gatto::Stampa(ostream &os) const{
    os << "nome: " << getNome() << " , razza: "
    << getRazza() << " , eta': " << getEta()
    << " , verso: " << verso();
}

#endif

```

BST.h

mercoledì 28 giugno 2023 21:31

main.cpp

mercoledì 28 giugno 2023 21:31

```
#include <iostream>
#include <string>
#include "Animali.h"
#include "BST.h"
using namespace std;
int main(){
    Animale *a = new Cane("Adolfo", "Labrador", 4);
    Animale *a2 = new Cane("Bobby", "Pastore tedesco", 5);
    Animale *a3 = new Gatto("Ciondolino", "Siamese", 6);
    Animale *a4 = new Cane("Solo", "come un cane", 50);

    BST tree;
    tree.inserisci(a);
    tree.inserisci(a2);
    tree.inserisci(a3);
    cout << tree;

    string str;
    cout << "inserisci un nome da cercare: ";
    cin >> str;
    //ricerca del nome
    if(tree.ricerca(str) != nullptr){
        cout << "Trovato!" << endl;
        Nodo* temp = tree.ricerca(str);
        cout << "ecco tutte le informazioni di " << str << endl;
        cout << *(temp->dato) << endl;
    }
    else{
        cout << "questo nome non e' presente" << endl;
    }

    return 0;
}
```

Testo

mercoledì 28 giugno 2023 21:33

Programmazione 2

Si consideri una classe virtuale *Animale* che rappresenta un animale domestico. La classe ha i seguenti attributi privati:

- *nome, razza, età*

La classe ha i seguenti metodi pubblici:

- *costruttore, get degli attributi, stampa*
- *un metodo virtuale puro chiamato verso che restituisce il verso dell'animale*

Si considerino due classi derivate da *Animale*: *Cane* e *Gatto*. Ognuna di queste classi implementa il metodo *verso* in modo appropriato per il tipo di animale.

Si consideri una classe *AlberoBinario* che rappresenta una struttura dati ad albero binario di ricerca di animali domestici. La classe ha i seguenti attributi privati:

- *radice: un puntatore al nodo radice dell'albero*
- *dimensione: il numero di nodi nell'albero*

La classe ha i seguenti metodi pubblici:

- *costruttore, distruttore*
- *inserisci: inserisce un nuovo animale nell'albero in base al suo nome*
- *cerca: cerca un animale nell'albero dato il suo nome e restituisce un puntatore all'animale se trovato, altrimenti restituisce nullptr*
- *stampa: stampa tutti gli animali nell'albero in ordine alfabetico dei loro nomi*

Si scriva un programma che crea un albero binario di animali di dimensione 10 e inserisce nell'albero i seguenti animali:

- *nome Fido, razza Labrador, età 3, tipo Cane*
- *nome Luna, razza Siamese, età 2, tipo Gatto*
- *nome Rex, razza Pastore Tedesco, età 5, tipo Cane*
- *nome Leo, razza Persiano, età 4, tipo Gatto*
- *nome Milo, razza Beagle, età 1, tipo Cane*
- *nome Nala, razza Maine Coon, età 3, tipo Gatto*

Il programma poi chiede all'utente di inserire il nome di un animale da cercare nell'albero e stampa le informazioni sull'animale se trovato, altrimenti stampa un messaggio di errore. Il programma poi stampa il verso dell'animale trovato usando il metodo *verso* della classe corrispondente.

Output atteso inserendo il nome Luna:

Inserisci il nome di un animale da cercare: Luna

Luna è un gatto di razza Siamese di 2 anni.

Il verso di Luna è: Miao!

Output atteso inserendo il nome Bobo:

Inserisci il nome di un animale da cercare: Bobo

Nessun animale con questo nome trovato nell'albero.