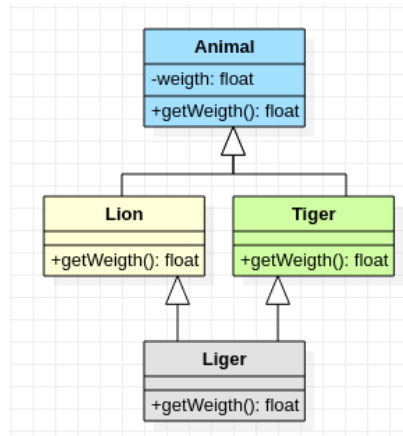


Programação Orientada a Objetos

Prof. Delano M. Beder

Roteiro 13b – Herança Múltipla (*Diamond problem*)

Diamond problem: <https://www.programmersought.com/article/81474725343/>



1. Crie um projeto (C++) denominado Liger
2. Implementação da classe C++ denominada Animal

Dois arquivos: Animal.h (Cabeçalho) e Animal.cpp (Código-fonte)

2.1 Arquivo Animal.h

```
#ifndef ANIMAL_H
#define ANIMAL_H

class Animal {
public:
    Animal(float weight);
    virtual ~Animal();
    virtual float getWeight() const;
private:
    float weight;
};

#endif /* ANIMAL_H */
```

2.2 Arquivo Animal.cpp

```
#include "Animal.h"

Animal::Animal(float weight) :
weight(weight) {
}

Animal::~~Animal() {
}

float Animal::getWeight() const {
    return weight;
}
```

3. Implementação da classe C++ denominada Lion

Dois arquivos: Lion.h (Cabeçalho) e Lion.cpp (Código-fonte)

3.1 Arquivo Lion.h

```
#ifndef LION_H
#define LION_H

#include "Animal.h"

class Lion : public Animal {
public:
    Lion(float weight);
    virtual ~Lion();
    float getWeight() const;
};

#endif /* LION_H */
```

3.2 Arquivo Lion.cpp

```
#include "Lion.h"

Lion::Lion(float weight) :
Animal(weight) {
}

Lion::~Lion() {
}

float Lion::getWeight() const {
    return Animal::getWeight();
}
```

4. Implementação da classe C++ denominada Tiger

Dois arquivos: Tiger.h (Cabeçalho) e Tiger.cpp (Código-fonte)

4.1 Arquivo Tiger.h

```
#ifndef TIGER_H
#define TIGER_H

#include "Animal.h"

class Tiger : public Animal {
public:
    Tiger(float weight);
    virtual ~Tiger();
    float getWeight() const;
};

#endif /* TIGER_H */
```

4.2 Arquivo Tiger.cpp

```
#include "Tiger.h"

Tiger::Tiger(float weight) :
Animal(weight) {
}

Tiger::~Tiger() {
}

float Tiger::getWeight() const {
    return Animal::getWeight() * 3;
}
```

5. Implementação da classe C++ denominada Liger

Dois arquivos: Liger.h (Cabeçalho) e Liger.cpp (Código-fonte)

5.1 Arquivo Liger.h

```
#ifndef LIGER_H
#define LIGER_H

#include "Lion.h"
#include "Tiger.h"

class Liger : public Lion, public Tiger {
public:
    Liger(float weight);
    virtual ~Liger();
    float getWeight() const;
};

#endif /* LIGER_H */
```

5.2 Arquivo Liger.cpp

```
#include "Liger.h"

Liger::Liger(float weight) :
Lion(weight), Tiger(weight) {
}

Liger::~Liger() {
}

float Liger::getWeight() const {
    return Lion::getWeight();
}
```

6. Arquivo main.cpp

```
#include <iostream>
#include "Liger.h"

using namespace std;

int main() {

    Liger lg(100.5);

    float weight = lg.getWeight();

    cout << weight << endl;

    return 0;
}
```

7. Compile e execute (verifique a saída impressa)

8. Fim