

Professor: Renato Martins (renato.martins AT dcc.ufmg.br)
<https://www.dcc.ufmg.br/~renato.martins/courses/DCC004>
2º Semestre de 2018

Laboratório 3 – Polimorfismo e Identificação Dinâmica de Objetos (Run-time Type Identification)

1. Hoje iremos modelar um zoológico e realizar a identificação dinâmica dos animais que estão no zoológico. Para isso usaremos um operador de identificação dinâmica disponível em C++: `typeid`. Em geral não queremos saber qual o tipo real de um objeto porque isso pode danificar o polimorfismo e a capacidade de generalização do seu programa. Mas em algumas situações é obrigatório conhecer o tipo de objeto que foi instanciado: é o caso ao realizar uma serialização. A serialização consiste em salvar o estado de um objeto em um formato que possa ser armazenado na memória e posteriormente pode ser lido e usado pelo seu programa.
 - (a) Crie uma interface `Animal` contendo um único método `emitir_som`.
 - (b) Modele o tipo abstrato de dados (TAD) zoológico usando uma classe com um vetor de `Animal` alocado dinamicamente. O seu TAD deverá implementar os métodos `inserir_animal(Animal& animal)`: insere um animal no zoológico e `mostrar_animais()`: imprime o número de animais de cada classe no zoológico.
 - (c) Em seguida, crie as classes `Ave` (classe abstrata), `Panda`, `Arara` e `Tamanduá` que implementam `Animal`.
 - (d) Em um método `main`, instancie um zoológico contendo objetos dos tipos `Panda`, `Arara` e `Tamanduá`. Em seguida, através de um loop, percorra a sua lista de animais chamando o método `emitir_som`.
 - (e) Realize a identificação de cada um dos animais usando a biblioteca `<typeinfo>`:
`const type_info& tipo_animal = typeid(obj)`

```
1 #include <iostream>
2 #include <typeinfo>
3 #include "src/animal.h"
4 #include "src/zoológico.h"
5
6 int main()
7 {
8     Zoologico zoo(100);
9     Animal* panda_feliz = new Panda();
10    zoo.inserir_animal(panda_feliz);
11    zoo.mostrar_animais();
12    return 0;
13 }
```