**UENF**

**João Víttor Vieira Pinto**

**Problema inicial para se resolver:**

João Vítor Fernandes Dias **+ João Víttor Vieira Pinto - Ruby**:

Dada uma classe denominada **Controle**, definida em um sistema de controle de transporte de pluviômetros, que possui dois métodos estáticos: **leString()** que solicita ao usuário uma string pelo teclado e a retorna através do método e **leInteiro()** que solicita ao usuário um valor inteiro pelo teclado e o retorna através do método.

Escreva uma classe herdeira de **Controle** que acrescente um método estático responsável pela seleção do caminhão mais apto a distribuição de pluviômetros. O caminhão será aquele capaz de conduzir pluviômetros cuja soma de capacidade seja a maior, independente do tipo de cada um deles.

O programa irá solicitar uma lista de caminhões. Para cada caminhão ele pergunta:

* Tipo do caminhão (‘Alfa’ ou ‘Beta’);
* Número de pluviômetros a ser transportados;
* Lista dos pluviômetros a ser transportados no caminhão (para cada pluviômetro é digitado apenas seu tipo).

A lista encerra quando é digitado ‘Fim’ para o Tipo do caminhão.  
Depois de digitada toda a lista o programa imprime na tela os seguintes dados do caminhão mais apto:

* Tipo do caminhão (Alfa ou Beta);
* Lista dos pluviômetros transportados no caminhão.

Em ambas as questões, devem ser criados os atributos que se fizerem necessários.

**Relatório Programação Orientada a Objeto**

* Foi nos dado o projeto acima para se implementar com a linguagem Ruby.
* No nosso código, foi necessário implementar outro projeto para fazer “certo” sentido, pois faltava algumas informações e só com ele conseguimos ter um norte (o nosso projeto é uma continuação).
* Tomamos a liberdade de fazer algumas mudanças na proposta.
* Irei explicar a nossa parte, mesmo tendo feito as duas (tem bastantes comentários no código em ambas partes).

Inicialmente, fizemos uma classe chamada “Controle” (como foi pedido) com dois métodos estáticos para exercer essa função. A classe “Truck”, que vem logo em seguida, se refere ao caminhão (mais especificamente o seu tipo, a quantidade de pluviômetros que ele possui e os seus tipos (com uma lista). Logo após, criamos uma classe herdeira de Controle (classe mãe), que é a ObterCaminhao (classe filha), usada para criar um caminhão com todas as informações antes pedidas, e assim, usando o conceito de polimorfismo. O principal conceito do polimorfismo no nosso código foi o fato da classe herdeira(filha) poder responder à mensagem da superclasse, de forma diferente (se necessário).

Por final, foi pedido para comparar os caminhões e descobrir o mais apto, mas não entendemos muito bem o critério, então colocamos para printar o que consegue carregar mais pluviômetro.

Ao decorrer do projeto utilizamos de objetos diversas vezes e, separadamente, de lógicas simples. Nós passamos um bom tempo fazendo um código para deixar da forma mais compreensível possível e tentando fazer funcionar, mas foi bem difícil. Mesmo estando grande, comentamos a atividade praticamente toda e espero que esteja entendível!