02

Herói

Transcrição

[00:00] Vamos começar nosso trabalho pelo herói. Ele é quem está em alguma posição do nosso mapa, em uma linha e coluna. Quero criar uma definição, a estrutura, a abstração de um herói. Não é o herói em si. É o que ele tem, como se comporta.

[00:32] Vou criar um arquivo chamado herói.rb. Se estou definindo a estrutura, dizemos para o Ruby que estamos definindo a classe herói. A classe dos heróis. Agora, posso sair criando os heróis que eu quiser. Por exemplo, Guilherme é igual a herói.new. Cada uma dessas variáveis referencia um espaço na memória que é meu objeto, a instância de um herói. A abstração chamamos de classe. O objeto em si chamamos de objeto. E a variável referencia esses objetos. O Guilherme é um herói que está na memória. O Carlos é outro herói que está na memória.

[01:42] Mas um herói assim, sem nada, não tem muita graça. Um herói tem dois atributos, a linha e a coluna. Vou falar para o Ruby que quero ter dois atributos com a capacidade de ser lido e escrito. A linha pode ser alterada, a coluna pode ser alterada. E posso ler esses atributos de fora da minha classe. Quero ter acesso tanto à linha quanto à coluna. Eles serão atributos, cada herói terá uma linha, uma coluna. Mas o método, a função attr, quando chamo, passo para o Ruby dois símbolos. O símbolo linha e o símbolo coluna, como assessores do nosso herói. Todo herói vai ter uma linha e uma coluna.

[03:05] Se eu rodo o código, ele não faz nada, porque preciso imprimir algo. Por exemplo, o Guilherme está em Guilherme.linha. Rodo de novo, o Guilherme está em 15. Se eu colocar Guilherme.coluna, ele está na coluna 3. Guilherme é uma referência para o objeto herói, que tem tanto linha quanto coluna. Dois valores dentro do meu objeto. Dois atributos do meu objeto.

[03:42] Posso tentar imprimir o próprio Guilherme. Ele me dá que o Guilherme é um herói e um endereço na memória para identificar o objeto. Tanto que se eu tentar imprimir o Carlos, tenho outro endereço. São dois objetos distintos.

[04:33] Eu instancio objetos através do new e acesso os atributos através do operador de acesso, o ponto. Muito parecido com o que fazíamos com string, com array. Criei o meu objeto e acessei os atributos. Em outras linguagens, temos uma característica que se chama estrutura. A estrutura é só um valor que pode ser criado, que tem atributos lá dentro. Isso lembra bastante a estrutura, que tem valores que posso colocar e valores que posso ler. Mas em orientação de objetos, em Ruby, vamos ter muito mais que podemos colocar numa classe. Uma classe não é uma mera estrutura que acumula dados. Ela é muito mais do que isso.