Capítulo 10

Armazenamento Associativo

"A cooperação é a convicção plena de que ninguém pode chegar à meta se não chegarem todos"
— Virginia Burden

10.1 - Motivação

Hoje em dia, devido a correria do dia a dia, estamos acostumados com os restaurantes "fast food". Como o próprio nome diz o principal atrativo destes restaurantes é ou deveria ser a rapidez no atendimento. Normalmente, para tornar mais ágil o atendimento, os pedidos são feitos por número. O cliente escolhe o que deseja comer e informa o número correspondente ao seu pedido ao atendente. Cada número está associado a um prato ou a um pedido.

Para controlar a circulação de carros e motos nas ruas das cidades, algum órgão governamental associa uma placa a cada veículo. Para emitir uma multa ou denunciar um roubo à polícia, informamos a placa que identifica o veículo. A placa é formada de letras e números e funciona como uma chave para encontrar o veículo desejado.

A televisão está presente na vida de todas as pessoas. Ela é o principal meio de comunicação nos dias atuais e é dividida em canais. Cada canal escolhe a sua própria programação de acordo com o público alvo. Para escolher qual canal assistir, uma pessoa digita no controle ou no próprio aparelho de TV o número que está associado ao canal desejado.

Além destes, há muitos outros exemplos em que um elemento está associado a outro. Por isso, vamos estudar alguma estrutura de dados que permita associações deste tipo.

10.2 – Mapa

A primeira funcionalidade que precisamos na nossa estrutura e uma que faça as associações entre os elementos. Esta operação deve receber uma chave e um valor para criar a associação entre eles.

É necessário recuperar o valor associado a uma chave rapidamente. Uma operação que faça isso deve ser implementada. Ela receberá uma chave e devolverá o valor associado se ele existir.

Provavelmente, será necessário remover uma associação da nossa estrutura em alguns casos. Deverá existir uma funcionalidade para fazer isso. Dado uma chave remove a associação que a chave participa.

Outra operação interessante seria poder verificar se uma determinada chave está ou não associada a algum valor.

Além disso, precisamos saber quantas associações existem na estrutura.

Um requisito fundamental para o Mapa é que as chaves sejam únicas. Não pode existir dois carros com a mesma placa, dois canais de televisão no mesmo número ou dois pedidos de números iguais.

Os Mapas são as estruturas de dados que implementam o tipo de situação que estamos discutindo aqui.

1. Adicionar uma associação.
2. Pegar um valor dado uma chave.
3. Remover uma associação dado uma chave.
4. Verificar se existe uma associação para uma determinada chave.
5. Informar a quantidade de associações.

Queremos modelar uma estrutura de dados para ser utilizada em um sistema de armazenamento de registros de veículos. Este sistema poderia ser útil para o Detran.

As informações sobre um determinado carro são obtidas a partir da placa do mesmo. Então, o mapa que queremos fazer aqui vai associar as placas aos carros. As placas podem ser `String` e para os carros podemos implementar uma classe

Carro.

```
class Carro {

  private String nome;
  private String marca;
  private String cor;
  private int ano;

  // getters e setters
}
```

O ponto fundamental do Mapa é a associação das chaves com os valores. Para modelar uma associação, vamos definir uma classe.

```
class Associacao {
  private String placa;
  private Carro carro;

  public Associacao(String placa, Carro carro){
    this.placa = placa;
    this.carro = carro;
  }
  // getters
}
```


10.3 – Exercícios: Armazenamento Associativo

1. Implemente a classe `Carro` no pacote **br.com.caelum.ed**.

```
public class Carro {
  private String nome;

  public Carro(String nome) {
    this.nome = nome;
  }

  public String getNome() {
```

```
    return nome;
  }

  @Override
  public String toString() {
    return "Carro: " + this.nome;
  }
}
```

2. Implemente a classe `Associacao` que modela o relacionamento entre as chaves e os valores do Mapa no pacote **br.com.caelum.ed.mapas**.

```
public class Associacao {
  private String placa;
  private Carro carro;

  public Associacao(String placa, Carro carro) {
    this.placa = placa;
    this.carro = carro;
  }

  public String getPlaca() {
    return placa;
  }

  public Carro getCarro() {
    return carro;
  }
}
```

CAPÍTULO ANTERIOR:

Tabelas de Espalhamento

PRÓXIMO CAPÍTULO:

Mapas com Lista

Cursos Online

Facebook

Newsletter

Casa do Código

Twitter