Iterator

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Câmpus Rio Grande
Divisão de Computação

Trabalho

- Muito bem, sabemos que as interfaces de PancakeHouseMenu e DinerMenu são exatamente idênticas, mas ainda não definimos uma interface comum para elas.
- Agora faremos isso refinar um pouco mais o funcionamento da Garçonete.

Trabalho

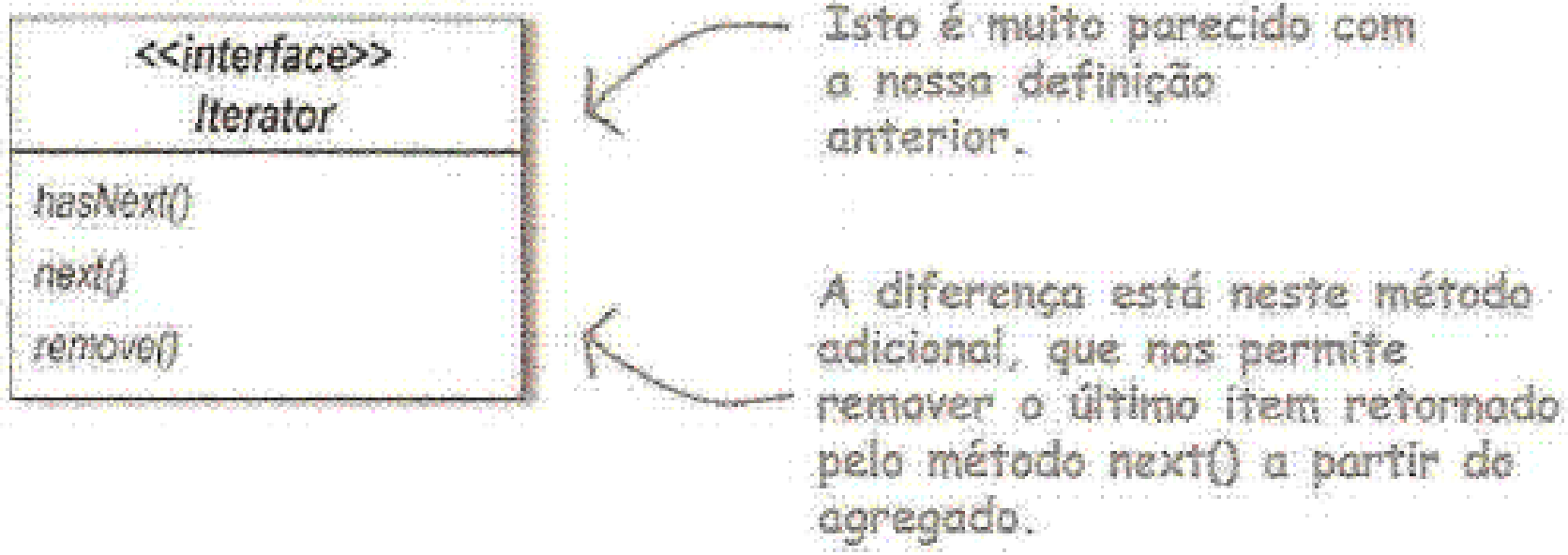
- Talvez você esteja se perguntando por que não estamos usando a interface Iterator do Java - bem, queríamos que você percebesse como é possível criar um iterador a partir de zero.
- Agora que já fizemos isso, vamos mudar para a interface Iterator do Java, porque ela é mais poderosa do que a nossa interface Iterator caseira.
- **Poderosa como? Logo você verá.**

Exercício 1

Adapte o exemplo para a interface java.util.Iterator

Trabalho

Em primeiro lugar, vamos conferir a interface java.util.Iterator:



Trabalho

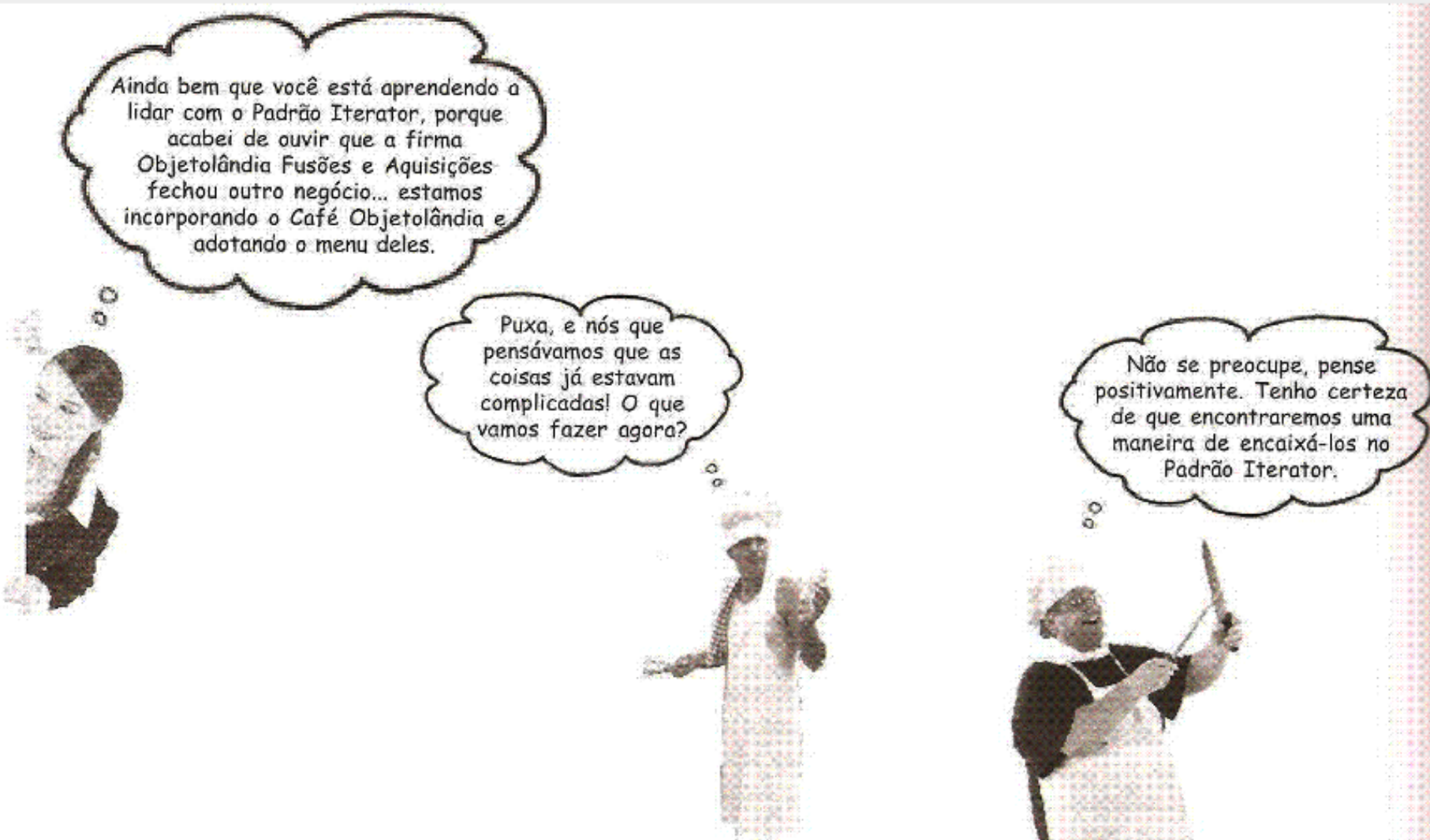
- Tudo que precisamos fazer é mudar a interface estendida por PancakeHouseMenuIterator e DinerMenuIterator, certo?
- Além de java.util possuir a sua própria interface Iterator, ArrayList também tem um método iterator() que retorna um iterador.
- Em outras palavras, na verdade nunca foi necessário que implementássemos o nosso próprio iterador para o ArrayList.

Trabalho

Observação

Por outro lado, nossa implementação continua a ser necessária em DinerMenu, porque ele é baseado numa matriz (Array), que não implementa o método iterator() (nem qualquer outra maneira de criar um iterador de matrizes)

Examinando o Menu do Café



Examinando o Menu do Café

Aqui está o menu do Café. Parece que não vai muito difícil integrá-lo à nossa estrutura. Vamos dar uma olhada:

```
public class CafeMenu {
  Hashtable menuItems = new Hashtable();

public CafeMenu() {
  addItem("Veggie Burger and Air Fries",
        "Veggie burger on a whole wheat bun, lettuce, tomato, and fries",
        true, 3.99);
  addItem("Soup of the day",
        "A cup of soup of the day, with a side salad",
        false, 3.69);
  addItem("Burrito",
          "A large burrito, with whole pinto beans, salsa, guacamole",
          true, 4.29);
  }
```

O menu do Café não implementa a nossa nova interface Menu, mas isso é fácil de corrigir.

O Café armazena seus itens de menu numa hashtable. Esse formato aceita o Iterator? É o que veremos a seguir...

Como nos outros menus, os itens de menu são inicializados no construtor.

Examinando o Menu do Café


```
public void addItem(String name, String description,
                    boolean Vegetarian, double price)
{
      MenuItem menuItem = new MenuItem(name, description, vegetarian, price);
      menuItems.put(menuItem.getName(), menuItem);
}
public Hashtable getItems() {
      return menuItems;
}
```

É aqui que criamos um novo item de menu e o acrescentamos à hashtable MenuItem.

o valor do objeto menuItem.

a chave é o nome do item.

Não precisaremos mais disto.

Adaptando o código do menu do Café

- A integração do menu do Café à nossa estrutura é fácil. Porque Hashtable é uma das coleções em Java que comportam o Iterador.
- Mas não exatamente como ArrayList...

Hashtable é uma coleção obsoleta. Caso queira, mude a coleção.

Adaptando o código do menu do Café

```
public class CafeMenu implements Menu {
  Hashtable menuItems = new Hashtable();

  public CafeMenu() {
        //código do construtor aqui
  }
  public void addItem(String name, String description,
                      boolean Vegetarian, double price)
  {
        MenuItem menuItem = new MenuItem(name, description, vegetarian, price);
        menuItems.put(menuItem.getName(), menuItem);
  }

   public Hashtable getItems() {
         return menuItems;
   }

   public Iterator createIterator() {
         return menuItems.values().iterator();
   }
}
```

O menu do Café agora implementa a interface Menu, que a Garçonete pode usar exatamente como nos dois outros menus.

Estamos usando Hashtable porque é uma estrutura comum de dados para armazenar valores; você também poderia usar uma estrutura mais recente, HashMap.

Como antes, podemos nos livrar de getItems() para não expor a implementação de menuItems à Garçonete.

E aqui é onde implementamos o método createIterator(). Observe que não estamos obtendo um Iterador para toda a Hashtable, mas apenas para os valores.

Adaptando o código do menu do Café

Close do código

A estrutura Hashtable é um pouco mais complexa que ArrayList, porque aceita simultaneamente chaves e valores, mas nada impede que obtenhamos um Iterador para os valores (que são os itens de menu).

```
public Iterator createIterator() {
    return menuItems.values().iterator();
}
```

Primeiro obtemos os valores da Hashtable, que formam uma coleção de todos os objetos que ela contém.

Felizmente, essa coleção aceita o método iterator(), que retorna um objeto do tipo java.util.Iterator.

Adaptando o código do menu do Café

Exercício 2

Adapte o exemplo para o cardápio de cafés.

Hashtable é uma coleção obsoleta. Caso queira, mude a coleção.

Iterator

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Câmpus Rio Grande
Divisão de Computação