迭代器模式

模式定义

迭代器模式提供一种方法顺序访问一个聚合对象中的各个元素, 而又不暴露其内部的表示结构

问题描述

模拟一个菜单, 并遍历输出

Code

Food.java

```
package Iterator;

public class Food {
    int index;
    static int NUM = 0;

    public Food() {
        index = NUM++;
    }

    @Override
    public String toString() {
        return "food index = " + index;
    }
}
```

Iterable.java

```
package Iterator;

public abstract class Iterable {
    public abstract Iterator getIterator();
}
```

Iterator.java

```
package Iterator;

public abstract class Iterator<T> {
    public abstract boolean hasNext();

    public abstract T next();

    public abstract void remove();
}
```

Menu.java

```
package Iterator;

class MenuIterator extends Iterator<Food> {
    int index, size;
    Food[] foods;
    Menu menu;

    public MenuIterator(Menu menu) {
        this.menu = menu;
        foods = menu.foods;
        size = menu.size;
        index = 0;
    }

    @Override
    public boolean hasNext() {
        return index != size;
    }

    @Override
    public Food next() {
        return foods[index++];
    }

    @Override
    public void remove() {
        for (int i = index; i < size; ++i)
            foods[i] = foods[i + 1];
        size = (size - 1) % Menu.maxn;
        menu.size = size;
    }
}

public class Menu extends Iterable {
    Food[] foods;
    int size;
    final static int maxn = 1000;

    Menu() {
        foods = new Food[maxn];
        size = 0;
    }

    void addFood(Food food) {
        foods[(size++) % maxn] = food;
        size %= maxn;
    }

    @Override
    public Iterator getIterator() {
        return new MenuIterator(this);
    }
}
```

运行结果

Main.java

```
package Iterator;

public class Main {
    public static void main(String[] args) {
//        System.out.println(new Food());
        Menu menu = new Menu();
        menu.addFood(new Food());
        menu.addFood(new Food());
        menu.addFood(new Food());
        Iterator iterator = menu.getIterator();
        while (iterator.hasNext())
            System.out.println(iterator.next());
        iterator = menu.getIterator();
        iterator.next();
        iterator.next();
        iterator.remove();
        iterator = menu.getIterator();
        while (iterator.hasNext())
            System.out.println(iterator.next());
    }
}
```

out

```
food index = 0
food index = 1
food index = 2
food index = 0
food index = 1
```