● J2Kad24D「世界にはばたく ECC フーズ！（ループの復習)」

世界にはばたく ECC フーズは外食チェーン店を次々と買収している。現在傘下にあるのは、ECC コーヒーと ECC ドーナツ、さらに他のチェーン店の買収も計画している。そこで全チェーン店のメニューを表示する処理を作成したい。まずは for 文を使って ECC コーヒーと ECC ドーナツのすべてのメニューを表示する処理を作成せよ。ただし ECC コーヒー・ECC ドーナツともに元は別々の会社だったため、メニュー管理方式が異なるので注意すること。

リスト 1 : MenuItem クラス（ファイル「MenuItem.java」）

```
public class MenuItem {
    private String name;          // メニュー名
    private int price;            // 値段
    public MenuItem(String name, int price) {
        this.name = name;
        this.price = price;
    }
    public void print() { System.out.println(name + ":" + price + "円"); }
}
```

メニュー名と値段を格納するクラス。print メソッドで表示する。

リスト 2 : ECC コーヒーのメニュー（ファイル「CafeMenu.java」）

```
public class CafeMenu {
    private MenuItem[] menu = new MenuItem[100];
    public CafeMenu() {
        menu[0] = new MenuItem("ドリップコーヒー", 390);
        menu[1] = new MenuItem("アールグレイ", 430);
        menu[2] = new MenuItem("オレンジジュース", 220);
        menu[3] = null;          // 終了コード
    }
    public MenuItem[] getMenu() { return menu; }   // menu 配列のゲッター
}
```

MenuItem の配列で管理している。配列の値が null のとき、それ以上メニューはない。

リスト 3 : ECC ドーナツのメニュー（ファイル「DonutMenu.java」）

```
public class DonutMenu {
    private String[] names = new String[100];      // メニュー名
    private int[] prices = new int[100];           // 値段
    public DonutMenu() {
        names[0] = "ハニーディップ";   prices[0] = 120;
        names[1] = "ハニーチュロ";     prices[1] = 130;
        names[2] = "チョコリング";     prices[2] = 140;
        prices[3] = -1;       // 終了コード
    }
    public String[] getNames() { return names; }   // メニュー名のゲッター
    public int[] getPrices() { return prices; }    // 値段のゲッター
}
```

メニュー名と値段をそれぞれ別々の配列で管理している。値段の値が-1 のとき、それ以上メニューはない。

なぜ、こんな方法で管理しているのか知る由もない（買収したときすでにこうなっていた)。

リスト４：メニュー表示処理（ファイル「J2Kad24D.java」）

```
public class J2Kad24D {
    public static void main(String[] args) {
         ⋮
        while (true) {
            System.out.print("どのメニューを表示しますか？ (0：ECC コーヒー、1：ECC ドーナツ、-1：終了) ＞");
            int shop = Integer.parseInt(in.next());
            if (shop < 0) break;

            switch(shop) {
                default:
                case 0:     // ECC コーヒー
                    ECC コーヒーの全メニューを表示する
                    break;
                case 1:     // ECC ドーナツ
                    ECC ドーナツの全メニューを表示する
                    break;
            }
            System.out.println();
        }
    }
}
```

課題完成時の画面

```
世界にはばたく ECC フーズ！
ただいま M&A で拡大中！！
どのメニューを表示しますか？ (0：ECC コーヒー、1：ECC ドーナツ、-1：終了) ＞0
ドリップコーヒー：390 円
アールグレイ：430 円
オレンジジュース：220 円

どのメニューを表示しますか？ (0：ECC コーヒー、1：ECC ドーナツ、-1：終了) ＞1
ハニーディップ：120 円
ハニーチュロ：130 円
チョコリング：140 円

どのメニューを表示しますか？ (0：ECC コーヒー、1：ECC ドーナツ、-1：終了) ＞-1
```

● J2Kad24C「世界にはばたく ECC フーズ！（ループの本質)」※J2Kad24D をコピーして作成

ECC コーヒー・ECC ドーナツともにメニュー一覧のデータ構造が異なるため、それぞれにメニュー表示処理を作る必要がある。「う〜ん、これは面倒だ」と思っていたところ、「私が ECC コーヒーのメニューを教えて差し上げましょう！」という人物が現れた。その名もカフェイテレータ（CafeIterator)。このカフェイテレータに尋ねれば ECC コーヒーのメニューをひとつずつ教えてくれるそうだ！CafeIterator を使って ECC コーヒーのメニュー表示処理を修正せよ。同じく ECC ドーナツに関しても DonutIterator を導入してメニュー表示処理を修正せよ。

CafeIterator クラス（ファイル「CafeMenu.java」に作成）

メンバ	説明
private MenuItem[] menu;	Café メニュー配列への参照。コンストラクタで設定。
private int i = 0;	配列のインデックス。初期値は 0。
public CafeIterator(MenuItem[] menu)	コンストラクタ。menu への参照を設定する。
public boolean hasNext()	i 番目のメニューがあれば true、なければ false を返す。
public MenuItem next()	i 番目のメニューを返し、i の値を 1 増やす。

DonutIterator クラス（ファイル「DonutMenu.java」に作成）

メンバ	説明
private String[] names;	Donut のメニュー名配列への参照。コンストラクタで設定。
private int[] prices;	Donut の値段配列への参照。コンストラクタで設定。
private int i = 0;	配列のインデックス。初期値は 0。
public DonutIterator(String[] names, int[] prices)	コンストラクタ。names と prices への参照を設定する。
public boolean hasNext()	i 番目のメニューがあれば true、なければ false を返す。
public MenuItem next()	i 番目のメニューを MenuItem として返し、i の値を 1 増やす。

ヒント：
ループ構造は「データがあるかどうか（hasNext)」と「データを取得する（next)」で構成することができる。

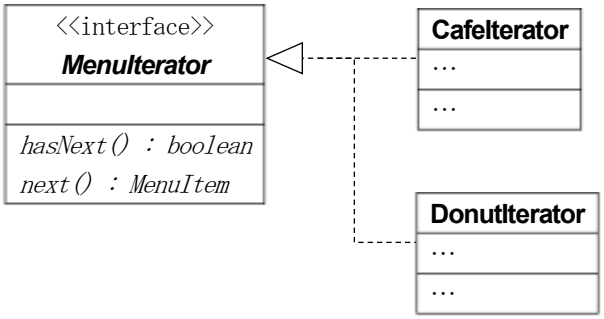
```
while( データがある？ ) {
    データを取得して処理をする
}
```

課題完成時の画面
（J2Kad24D と同じ）

● J2Kad24B1「世界にはばたく ECC フーズ！（ループの共通化)」※J2Kad24C をコピーして作成

CafeIterator と DonutIterator を使うとメニュー表示のループがほぼ同じ形になった。MenuIterator インターフェイスを導入し（ファイル「MenuItem.java」に作成)、ループ処理を共通化せよ。

課題完成時のクラス図



課題完成時の画面

（J2Kad24D と同じ）

● J2Kad24B2「世界にはばたく ECC フーズ！（Iterator パターン)」※J2Kad24B1 をコピー

CafeMenu と DonutMenu にそれぞれのイテレータを生成する処理（iterator メソッド）を作成し、Menu インターフェイス（ファイル「MenuItem.java」に新規作成）を導入してイテレータ取得処理を共通化せよ。

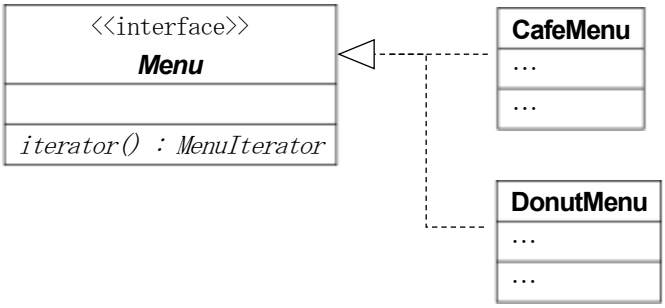
CafeMenu に追加するメソッド（Menu インターフェイスを実装）

メソッド	説明
public MenuIterator iterator()	CafeIterator を生成して返す。

DonutMenu に追加するメソッド（Menu インターフェイスを実装）

メソッド	説明
public MenuIterator iterator()	DonutIterator を生成して返す。

課題完成時のクラス図



課題完成時の画面

（J2Kad24D と同じ）

● J2Kad24A「世界にはばたく ECC フーズ！（M&A）」※J2Kad24B2 をコピーして作成

世界にはばたく ECC フーズが今度はあのハンバーガーチェーンを買収した！ハンバーガーチェーンのメニュー表示を追加せよ。なお、バーガーチェーン（BurgerMenu）にもイテレータ（BurgerIterator）を導入すること。

リスト１：BurgerMenu クラス（ファイル「BurgerMenu.java」）

```
public class BurgerMenu {
    private ArrayList<MenuItem> menu = new ArrayList<>();
    public BurgerMenu() {
        menu.add(new MenuItem("ハンバーガー", 150));
        menu.add(new MenuItem("チーズバーガー", 180));
        menu.add(new MenuItem("ビッグマック", 410));
    }
    public ArrayList<MenuItem> getMenu() { return menu; }
}
```

ArrayList で管理している。

課題完成時の画面

```
世界にはばたく ECC フーズ！
ただいま M&A で拡大中！！
どのメニューを表示しますか？（0：ECC コーヒー、1：ECC ドーナツ、2：ECC バーガー、-1：終了）＞2
ハンバーガー：150 円
チーズバーガー：180 円
ビッグマック：410 円

どのメニューを表示しますか？（0：ECC コーヒー、1：ECC ドーナツ、2：ECC バーガー、-1：終了）＞
```

● J2Kad24S「世界にはばたく ECC フーズ！（匿名クラス)」※J2Kad24A をコピーして作成

CafeIterator・DonutIterator・BurgerIterator はそれぞれのショップの店員（内部の人間）だった！どうりで各ショップの内部事情に通じていたわけだ。

① 各イテレータを対応するメニューの内部クラスにせよ（iterator メソッド内で定義する)。なお、内部クラスにすれば private なメニューデータを直接参照できるため、簡略化することができる。
② さらに①で作成した内部クラスを匿名クラスにせよ。

※ J2Kad24A をコピーして作成するが、J2Kad24S は特に修正する箇所はない（とりあえず A とは別の課題ということで)。
※ 各 Iterator クラスは内部クラス・匿名クラスにすると不要になるが、削除せずに残しておくこと（他の課題でエラーが出るので)。

ヒント：内部クラス・匿名クラスについては以下を参照、もしくは検索すること
・（もし教科書を持ってきていれば）実践編 P. 125～P. 130
・J2Kad19D「内部クラス」、J2Kad19C「匿名クラス（無名クラス)」

課題完成時の画面
（J2Kad24A と同じ）

● J2Kad24X「世界にはばたく ECC フーズ！（拡張 for 文対応)」

※パッケージ pac24x のファイルを修正

パッケージ pac24x に業者が作った恐ろしいメニュー表示処理が準備されている。これまでの知識を動員してリファクタリングせよ。良い子は恐ろしいコードを見てはいけないが、課題を作るために今回は許可する。なお、メニュー表示処理は拡張 for 文を使って記述すること。

ヒント：
Iterable インターフェイス（←検索）と Iterator インターフェイス（←検索）を実装すれば、拡張 for 文を使って記述することができる。

課題完成時の画面
（J2Kad24A と同じ）