

● ファミレス ECC、すべてはお客様のために！ (Before)

「すべてはお客様のために！」をモットーとするファミレス ECC では、メニュー表示はこれまでメニュー名と値段を表示するだけであったが、お客様の要望により「カロリー表示」と「メニュー名だけの横一列 (ライン) 表示」を追加することになった。業者に相談すると「メニューを構成するクラスに新しいメソッドを追加すればいいだっちゃ！」とのこと。今後どういう表示機能の追加があるかかもしれず、これではその都度メソッドの追加が必要になる。新しい表示が追加されても対応できるようにクラスを修正せよ。

実行時の画面イメージ

ファミレス ECC へようこそ！

どの形式で表示しますか？ (0 : 通常、1 : カロリー、2 : ライン、-1 : 終了) >0

★ファミレス ECC メニュー★

ハンバーグステーキ : 449 円

豚肉の生姜焼き : 599 円

スパゲッティ : 499 円

サラダメニュー

豆腐とオクラのサラダ : 349 円

シーザーサラダ : 399 円

シーフードサラダ : 399 円

どの形式で表示しますか？ (0 : 通常、1 : カロリー、2 : ライン、-1 : 終了) >1

★ファミレス ECC メニュー★

ハンバーグステーキ : 449 円 (662Kcal)

豚肉の生姜焼き : 599 円 (678Kcal)

スパゲッティ : 499 円 (625Kcal)

サラダメニュー

豆腐とオクラのサラダ : 349 円 (157Kcal)

シーザーサラダ : 399 円 (290Kcal)

シーフードサラダ : 399 円 (290Kcal)

どの形式で表示しますか？ (0 : 通常、1 : カロリー、2 : ライン、-1 : 終了) >2

★ファミレス ECC メニュー★

ハンバーグステーキ / 豚肉の生姜焼き / スパゲッティ /

サラダメニュー

豆腐とオクラのサラダ / シーザーサラダ / シーフードサラダ /

どの形式で表示しますか？ (0 : 通常、1 : カロリー、2 : ライン、-1 : 終了) >-1

これまでのメニュー表示
(print メソッド)

カロリー付きの表示
(princCal メソッド)

横一列表示
(printLine メソッド)

リスト1: 業者の提案 (ファイル「before.Menu.java」)

// メニュー表示インターフェイス

```
interface Component {
    void print();           // 通常表示
    void printCal();        // カロリー表示
    void printLine();       // 横一列表示
}
```

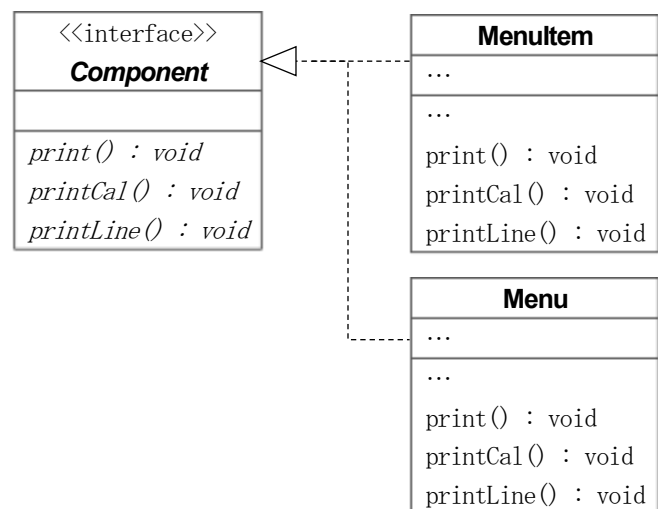
// メニュー (商品名)

```
class MenuItem implements Component {
    private String name;      // 名前
    private int price;        // 値段
    private int cal;          // カロリー
    :
    public void print() { System.out.println(name + ":" + price + "円"); }
    public void printCal() { System.out.println(name + ":" + price + "円 (" + cal + "Kcal) "); }
    public void printLine() { System.out.print(name + " / "); }
}
```

// メニュー (一覧)

```
public class Menu implements Component {
    private String name;
    private ArrayList<Component> menuList = new ArrayList<>();
    :
    public void print() {
        System.out.println();
        System.out.println(name);
        for (var item : menuList) {
            item.print();           // 通常表示
        }
    }
    public void printCal() {
        System.out.println();
        System.out.println(name);
        for (var item : menuList) {
            item.printCal();        // カロリー表示
        }
    }
    public void printLine() {
        System.out.println();
        System.out.println(name);
        for (var item : menuList) {
            item.printLine();       // 横一列ライン表示
        }
    }
}
```

業者によると新たに「printCal メソッドと printLine メソッドを追加すればいいだっちゃ」とのこと (太字部分)。



● J2Kad27D「表示機能の切り離し (Visitor パターン)」

ハンバーグステーキの単品メニュー（名前と値段）を表示する処理が準備されている（リスト1）。MenuItem の print メソッドと同等の処理を行う Printer クラスを作成し、この Prniter クラスを使ってメニュー表示を行うように MenuItem クラスを修正せよ。

Printer クラスの仕様（新規ファイル「Printer.java」を作成する）

| メソッド | 説明 |
|----------------------------------|---|
| public void print(MenuItem item) | 受け取った item の名前と値段を表示する (MenuItem.print メソッドと同等の処理)。 |

MenuItem クラスに追加するメソッド

| 追加するメソッド | 説明 |
|------------------------------------|---------------------------|
| public void print(Printer printer) | 受け取った printer に表示処理を委譲する。 |

リスト1：ハンバーグステーキの単品表示（ファイル「J2Kad27D.java」）

```
public class J2Kad27D {
    public static void main(String[] args) {
        MenuItem item = new MenuItem("ハンバーグステーキ", 449, 662);
        ①Printer クラスのインスタンスを生成する
        item.print(); // ②引数として①で生成したインスタンスを渡す
    }
}
```

課題完成時の画面

| |
|-----------------|
| ハンバーグステーキ：449 円 |
|-----------------|

● J2Kad27C「カロリー表示の追加」

Printer クラスを継承して、カロリー表示を行う CalPrinter クラスを作成し、「0 : 通常」を選択したときは通常表示、「1 : カロリー」を選択したときはカロリー表示を行うように処理を修正せよ。

CalPrinter クラスの仕様 (Printer を継承、ファイル「Printer.java」に作成)

| メソッド | 説明 |
|----------------------------------|------------------------------|
| public void print(MenuItem item) | 受け取った item の名前と値段とカロリーを表示する。 |

リスト1 : カロリー表示の追加 (ファイル「J2Kad27C.java」)

```
public class J2Kad27C {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        MenuItem item = new MenuItem("ハンバーグステーキ", 449, 662);

        while(true) {
            System.out.print("どの形式で表示しますか？ (0 : 通常、1 : カロリー、-1 : 終了) >");
            int n = Integer.parseInt(in.next());
            if (n < 0) break;
            nが0のとき通常表示、1のときカロリー表示を行う (default のときは通常表示)
            System.out.println();
        }
    }
}
```

課題完成時の画面

どの形式で表示しますか？ (0 : 通常、1 : カロリー、-1 : 終了) >0
ハンバーグステーキ : 449 円

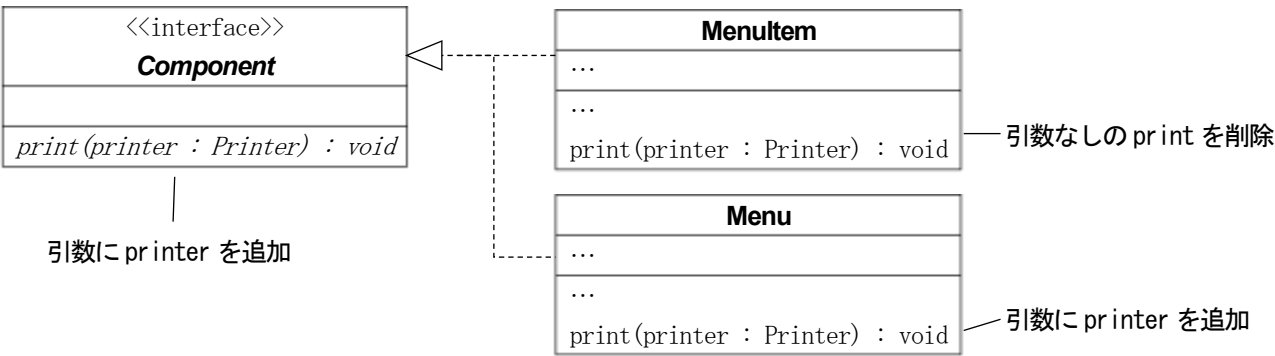
どの形式で表示しますか？ (0 : 通常、1 : カロリー、-1 : 終了) >1
ハンバーグステーキ : 449 円 (662Kcal)

どの形式で表示しますか？ (0 : 通常、1 : カロリー、-1 : 終了) >-1

● J2Kad27B「ライン表示の追加 (Composite 対応①)」

MenuItem に対しては Printer を指定することでメニュー表示を切り替えることが可能になっている。Menu クラスに対しても Printer を指定できるように修正し、ランチメニュー (lunch) を表示できるようにせよ。また「2:ライン」(横一列ライン表示)を追加せよ。

課題完成時のクラス図



LinePrinter クラスの仕様 (Printer を継承、ファイル「Printer.java」に作成)

| メソッド | 説明 |
|----------------------------------|------------------------------------|
| public void print(MenuItem item) | 受け取った item の名前に「/」を付けて表示する (改行なし)。 |

リスト1: ランチメニューの表示 (ファイル「J2Kad27B.java」)

```
public class J2Kad27B {
    public static void main(String[] args) {
        :
        while(true) {
            System.out.print("どの形式で表示しますか? (0:通常、1:カロリー、2:ライン、-1:終了) >");
            int n = Integer.parseInt(in.next());
            if (n < 0) break;
            nが0のとき通常表示、1のときカロリー表示、2のときライン表示を行う (defaultは通常表示)
            System.out.println();
        }
    }
}
```

課題完成時の画面

(最初のページの実行時の画面イメージと同じ)

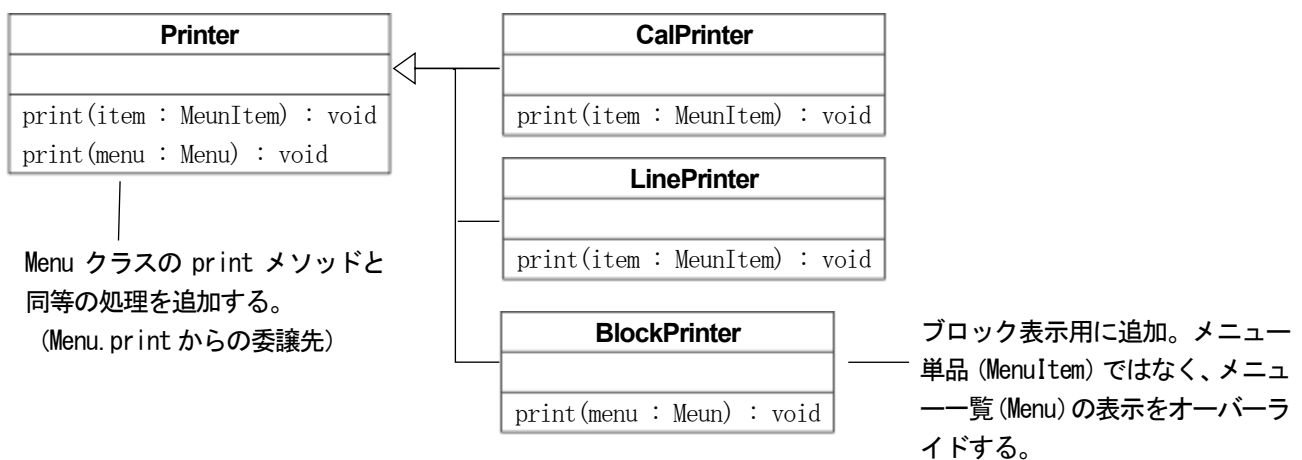
● J2Kad27A 「ブロック表示の追加 (Composite 対応②)」

あるお客様からメニュー一覧表示を中カッコで囲ってほしいという要望があった。どうやらどこかの会社のプログラマらしい (胸につけている社章に「ECC ソフト」と書いてあった)。「すべてはお客様のために！」をモットーとするファミレス ECC としてはどんな要望も無視するわけにはいかない。メニュー一覧表示を中カッコで囲うように処理を修正せよ。

ヒント：

中カッコを付けるためには Menu クラスの print メソッドを「通常表示」と「中カッコつき表示 (ブロック表示)」に対応させる必要がある。Menu.print メソッドの処理も printer に委譲すれば、Printer クラスでの対応が可能になる。

課題完成時のクラス図



課題完成時の画面

どの形式で表示しますか？ (0 : 通常、1 : カロリー、2 : ライン、3 : ブロック、-1 : 終了) >3

★ファミレス ECC メニュー★ {

ハンバーグステーキ : 449 円

豚肉の生姜焼き : 599 円

スパゲッティ : 499 円

サラダメニュー {

豆腐とオクラのサラダ : 349 円

シーザーサラダ : 399 円

シーフードサラダ : 399 円

}

デザートメニュー {

ソフトクリーム : 279 円

チーズケーキ : 299 円

チョコレートパフェ : 499 円

}

}

ランチメニューの
中カッコの対応

サラダメニューの
中カッコの対応

デザートメニューの
中カッコの対応

● J2Kad27S 「インデントの追加」

例のプログラマのお客様の要望で中カッコつき表示（ブロック表示）を追加したところ、「中カッコで挟まれているところは、右へ字下げするだっちゃ」という~~クレーン~~要望が入った。どこかで耳にした「だっちゃ」が気になるが、「すべてはお客様のために！」をモットーとするファミレス ECC としてはどんな言いがかり要望も無視するわけにはいかない。中カッコで囲まれた部分をタブ 1 分分右へ字下げするように修正せよ。

課題完成時の画面

どの形式で表示しますか？ (0 : 通常、1 : カロリー、2 : ライン、3 : ブロック、-1 : 終了) >3

```
★ファミレス ECC メニュー★ {
  ハンバーグステーキ : 449 円
  豚肉の生姜焼き : 599 円
  スパゲッティ : 499 円

  *サラダメニュー* {
    豆腐とオクラのサラダ : 349 円
    シーザーサラダ : 399 円
    シーフードサラダ : 399 円
  }

  *デザートメニュー* {
    ソフトクリーム : 279 円
    チーズケーキ : 299 円
    チョコレートパフェ : 499 円
  }
}
```

● J2Kad27X 「ECC ソフトの憂鬱」※次回、解答編の予定

「信頼と実績」の ECC ソフト株式会社では請け負った仕事はすべて下請け業者に丸投げしている。どうしても手が回らないときだけ協力会社を手伝ってもらっている。ただこのところどうもソフトウェアの品質が悪く、何度も「リファクタリング」されている。どうも仕事を発注するたびに同じ協力会社だと思っていたのが、入れ替わっているようだ！協力会社は ECC サービス、ECC パートナーズ、ECC ソリューションズの 3 社。同じ会社に発注していたつもりが、毎回、同じ名前の違う会社に発注しているらしい。このままでは「信頼と実績」にキズが付く。協力会社が入れ替わっていないかどうか調査し、名前が同じであれば毎回、同じ会社（インスタンス）に発注するように業務改善せよ。

- ① 協力会社のインスタンスを生成するたびに異なる ID 番号（会社間でも異なるようにする）を割り当てる処理を追加し、協力会社が入れ替わっていることを確認せよ。
- ② 協力会社のインスタンスが複数生成されないように（1 つの会社のインスタンスは 1 つだけになるように）処理を修正せよ。

①まで作成時の画面（同じ会社でも ID 番号が異なる → インスタンスが異なる）

信頼と実績の ECC ソフト株式会社です！

どんな課題でも私たちが真摯に解決します！！

どうしますか？（0：仕事を依頼する、-1：もういいい） >0

ECC サービス株式会社（ID：0）に丸投げします！

ECC サービス株式会社「何とかがんばってみます！」

どうしますか？（0：仕事を依頼する、-1：もういいい） >0

ECC ソリューションズ株式会社（ID：1）に丸投げします！

ECC ソリューションズ株式会社「よろこんでお引き受けいたします！」

どうしますか？（0：仕事を依頼する、-1：もういいい） >0

ECC ソリューションズ株式会社（ID：2）に丸投げします！

ECC ソリューションズ株式会社「よろこんでお引き受けいたします！」

どうしますか？（0：仕事を依頼する、-1：もういいい） >0

ECC サービス株式会社（ID：3）に丸投げします！

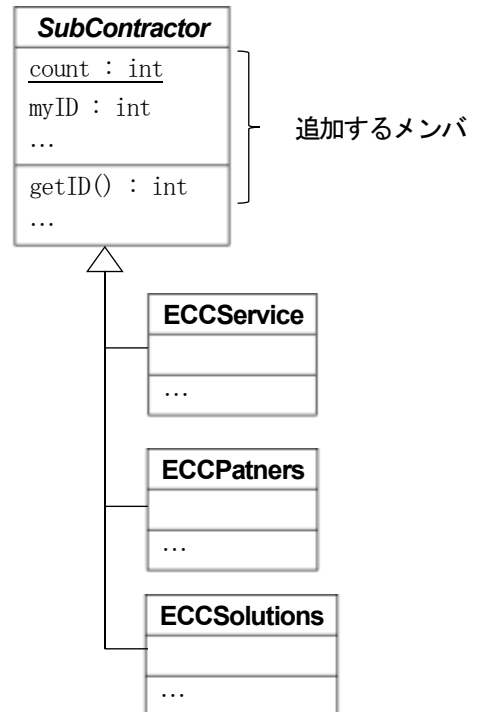
ECC サービス株式会社「何とかがんばってみます！」

どうしますか？（0：仕事を依頼する、-1：もういいい） >0

ECC パートナーズ株式会社（ID：4）に丸投げします！

ECC パートナーズ株式会社「下請けはつらいなー！」

仕様①は SubContractor に ID 番号を割り振る処理を追加すれば OK



課題完成時の画面（会社が同じときは ID も同じ → 同じインスタンス）

信頼と実績の ECC ソフト株式会社です！

どんな課題でも私たちが真摯に解決します！！

どうしますか？（0：仕事を依頼する、-1：もういいい） >0

ECC サービス株式会社（ID：0）に丸投げします！

ECC サービス株式会社「何とかがんばってみます！」

どうしますか？（0：仕事を依頼する、-1：もういいい） >0

ECC ソリューションズ株式会社（ID：1）に丸投げします！

ECC ソリューションズ株式会社「よろこんでお引き受けいたします！」

どうしますか？（0：仕事を依頼する、-1：もういいい） >0

ECC ソリューションズ株式会社（ID：1）に丸投げします！

ECC ソリューションズ株式会社「よろこんでお引き受けいたします！」

どうしますか？（0：仕事を依頼する、-1：もういいい） >0

ECC サービス株式会社（ID：0）に丸投げします！

ECC サービス株式会社「何とかがんばってみます！」

どうしますか？（0：仕事を依頼する、-1：もういいい） >0

ECC パートナーズ株式会社（ID：2）に丸投げします！

ECC パートナーズ株式会社「下請けはつらいなー！」