

● J2Kad23D「Composite パターンへの道① (単品表示と一覧表示)」

メニュー単品を表す MenuItem クラスとモーニングメニュー一覧を表す MorningMenu クラス (MenuList クラスを継承) が準備されている。MenuItem と MorningMenu を表示する処理を作成せよ。

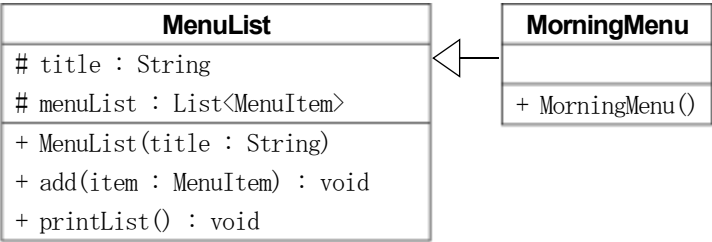
MenuItem (単品) での表示

・モーニングセット 400 円

MenuItem
- name : String
- price : int
+ MenuItem(name : String, price : int)
+ printItem() : void

モーニングメニュー (一覧での表示)

MorningMenu
・モーニングセット 400 円
・焼き魚定食 600 円



課題完成時の画面

ファミレス ECC へようこそ！
どうしますか？ (0：単品表示、1：一覧表示、-1：終了) >0
モーニングセット：400

どうしますか？ (0：単品表示、1：一覧表示、-1：終了) >1

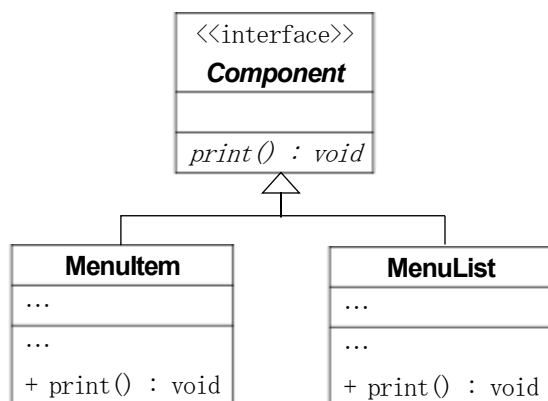
★ モーニングメニュー ★
モーニングセット：400
焼き魚定食：600

どうしますか？ (0：単品表示、1：一覧表示、-1：終了) >-1

● J2Kad23C 「Composite パターンへの道② (インターフェイスの共通化)」

Component インターフェイスを作成し、J2Kad23D の単品表示と一覧表示の表示処理を共通化せよ。

課題完成時のクラス構成



課題完成時の画面

(J2Kad23D と同じ)

● J2Kad23B 「Composite パターンへの道③ (参照の共通化)」

MenuList クラスの List コレクションを MenuItem と MenuList の共通部分を参照するように修正し、モーニング・ランチ・ディナーのそれぞれのメニューを表示する処理を作成せよ。なお、Component クラス、MenuItem クラス、MenuList クラスは J2Kad23C のものをコピーして作成すること (コピーせずに直接修正しても OK)。

ランチメニューとディナーメニュー

LunchMenu	
・日替わりランチ	680 円
・シェフのおすすめランチ	890 円
DrinkMenu	
・コーヒー	220 円
・紅茶	220 円

ランチにはサブメニューとしてドリンク、ディナーにはドリンクとデザートを追加する。

DinnerMenu	
・和風ハンバーグ	680 円
・サーロインステーキ	1180 円
DrinkMenu	
・コーヒー	220 円
・紅茶	220 円
DessertMenu	
・チーズケーキ	350 円
・アイスクリーム	150 円

課題完成時の MenuList クラス

MenuList
...
menuList : List<Component>
...
+ add(item : Component) : void

課題完成時の画面

何のメニューを表示しますか? (0 : モーニング、1 : ランチ、2 : ディナー、-1 : 終了) >1

★ ランチメニュー ★

日替わりランチ : 680

シェフのおすすめランチ : 890

* ドリンクメニュー *

コーヒー : 220

紅茶 : 220

何のメニューを表示しますか? (0 : モーニング、1 : ランチ、2 : ディナー、-1 : 終了) >2

★ ディナーメニュー ★

和風ハンバーグ : 680

サーロインステーキ : 1180

* ドリンクメニュー *

コーヒー : 220

紅茶 : 220

* デザートメニュー *

チーズケーキ : 350

アイスクリーム : 150

何のメニューを表示しますか? (0 : モーニング、1 : ランチ、2 : ディナー、-1 : 終了) >-1

● J2Kad23A 「ファミレス ECC (完成版)」

すべてのメニュー(モーニング・ランチ・ディナー)を表す AllMenu クラスを作成し、全メニューの表示を行え。AllMenu クラスの title には「★★★ すべてのメニュー ★★★」を設定すること。また以下のクラスは J2Kad23B のものをコピーして使用すること (コピーせずに直接修正しても OK、その方が Composite の練習になるかも)。

- ・ Component クラス
- ・ MenuItem クラス
- ・ MenuList クラス
- ・ LunchMenu クラス
- ・ DinnerMenu クラス

課題完成時の画面

ファミレス ECC へようこそ！

何のメニューを表示しますか？ (0 : モーニング、1 : ランチ、2 : ディナー、3 : すべてのメニュー、-1 : 終了) >3

★★★ すべてのメニュー ★★★

★ モーニングメニュー ★

モーニングセット : 400

焼き魚定食 : 600

★ ランチメニュー ★

日替わりランチ : 680

シェフのおすすめランチ : 890

* ドリンクメニュー *

コーヒー : 220

紅茶 : 220

★ ディナーメニュー ★

和風ハンバーグ : 680

サーロインステーキ : 1180

* ドリンクメニュー *

コーヒー : 220

紅茶 : 220

* デザートメニュー *

チーズケーキ : 350

アイスクリーム : 150

何のメニューを表示しますか？ (0 : モーニング、1 : ランチ、2 : ディナー、3 : すべてのメニュー、-1 : 終了) >-1

● J2Kad23S 「パッケージと Java ファイルの表示」

Composite パターンを使って J2Kad23 プロジェクトの src フォルダ以下のフォルダ (パッケージ) と Java ファイルを表示する処理を作成せよ。なお、パッケージへのファイルの追加、src フォルダへのパッケージの追加は main メソッドで行うこと (リスト1)。

リスト1 : main メソッド (ファイル「J2Kad23S.java」)

```
public class J2Kad23S {
    public static void main(String[] args) {
        Folder pac23a = new Folder("pac23a");
        pac23a.add(new File("J2Kad23A"));
        pac23a.add(new File("MenuList.java"));

        Folder pac23b = new Folder("pac23b");
        pac23b.add(new File("J2Kad23B"));
        pac23b.add(new File("MenuList.java"));

        Folder pac23c = new Folder("pac23c");
        pac23c.add(new File("J2Kad23C"));
        pac23c.add(new File("MenuList.java"));

        Folder pac23d = new Folder("pac23d");
        pac23d.add(new File("J2Kad23D"));
        pac23d.add(new File("MenuList.java"));

        Folder pac23s = new Folder("pac23s");
        pac23s.add(new File("Component.java"));
        pac23s.add(new File("J2Kad23S"));

        Folder pac23x = new Folder("pac23x");
        pac23x.add(new File("J2Kad23X.java"));

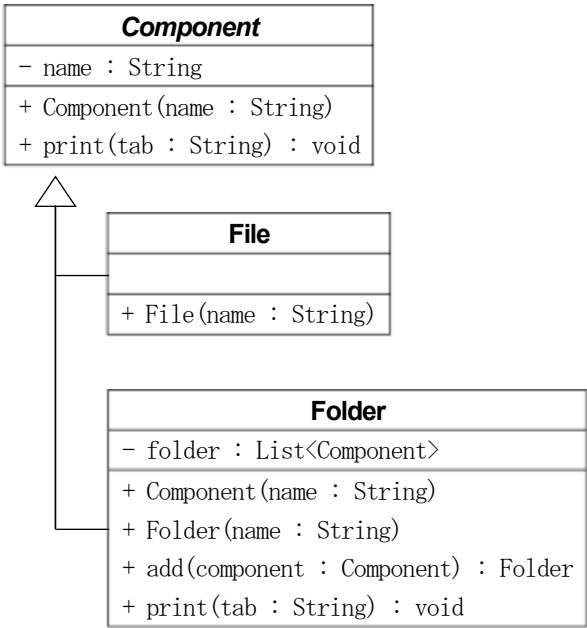
        Folder src = new Folder("src");
        src.add(pac23a);
        src.add(pac23b);
        src.add(pac23c);
        src.add(pac23d);
        src.add(pac23s);
        src.add(pac23x);
        src.print("");           // 表示
    }
}
```

課題完成時の画面 (タブでインデントする)

```
src
  pac23a
    J2Kad23A
    MenuList.java
  pac23b
    J2Kad23B
    MenuList.java
  pac23c
    J2Kad23C
    MenuList.java
  pac23d
    J2Kad23D
    MenuList.java
  pac23s
    Component.java
    J2Kad23S
  pac23x
    J2Kad23X.java
```

タブでインデントしている。ということは親のタブの情報を渡すしくみが必要。

クラス構成 (新規ファイル「Component.java」)



● J2Kad23X 「ガチャガチャマシーン！」※次回、解答編の予定

あの世界的大ヒット作「ポケット Duck!」を制作した ECC ゲームスが今度はガチャガチャマシーンに進出することになった！コインを入れても返却ボタンを押せば戻ってくるという画期的な仕様だ！！ただしその前にガチャガチャマシンの仕様のシミュレーションをしたい。シミュレーションプログラムを作成せよ。

ガチャガチャマシンの仕様

操作	動作
0：コインを入れる	コインが入っていないとき：「コインを入れました！」と表示する。 コインが入っているとき：「何も起こりません・・・」と表示する。
1：ハンドルを回す	コインが入っていないとき：「何も起こりません・・・」と表示する。 コインが入っているとき：「カプセルが出ました！」と表示する。
2：返却ボタンを押す	コインが入っていないとき：「何も起こりません・・・」と表示する。 コインが入っているとき：「コインが返却されました！」と表示する。

※ 可能な限りエレガントなコードを記述すること。なお、本課題は今回のテーマ (Composite) とは関係ないので注意すること (ヒント参照)。

課題完成時の画面 (仕様のすべての動作に問題がないか確認すること)

```

ガチャガチャをします！
コインが入っていません。
どうしますか？ (0：コインを入れる、1：ハンドルを回す、2：返却ボタンを押す、-1：終わる) >1
何も起こりません・・・
コインが入っていません。
どうしますか？ (0：コインを入れる、1：ハンドルを回す、2：返却ボタンを押す、-1：終わる) >2
何も起こりません・・・
コインが入っていません。
どうしますか？ (0：コインを入れる、1：ハンドルを回す、2：返却ボタンを押す、-1：終わる) >0
コインを入れました！
コインが入っています。
どうしますか？ (0：コインを入れる、1：ハンドルを回す、2：返却ボタンを押す、-1：終わる) >0
何も起こりません・・・
コインが入っています。
どうしますか？ (0：コインを入れる、1：ハンドルを回す、2：返却ボタンを押す、-1：終わる) >1
カプセルが出ました！
コインが入っていません。
どうしますか？ (0：コインを入れる、1：ハンドルを回す、2：返却ボタンを押す、-1：終わる) >0
コインを入れました！
コインが入っています。
どうしますか？ (0：コインを入れる、1：ハンドルを回す、2：返却ボタンを押す、-1：終わる) >2
コインが返却されました！
コインが入っていません。
どうしますか？ (0：コインを入れる、1：ハンドルを回す、2：返却ボタンを押す、-1：終わる) >-1
  
```

ヒント：

「State パターン」(←検索、次回予定)を適用するとエレガントになる。ただし、わからないときはベタベタのコードでも動作していれば本課題は OK とする。