# **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**

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

・モーニングセット 400円

**モーニングメニュー（一覧での表示）**

|  |
| --- |
| **MenuList** |
| # title : String  # menuList : List<MenuItem> |
| + MenuList(title : String)  + add(item : MenuItem) : void  + printList() : void |

|  |
| --- |
| **MorningMenu** |
|  |
| + MorningMenu() |

**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**

|  |
| --- |
| **MenuItem** |
| … |
| …  + print() : void |

|  |
| --- |
| **MenuList** |
| … |
| …  + print() : void |

|  |
| --- |
| <<interface>>  ***Component*** |
|  |
| *print() : void* |

**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 課題完成時のMenuListクラス**

|  |
| --- |
| **MenuList** |
| …  # menuList : List<Component> |
| …  + add(item : Component) : void |

**DinnerMenu**

・和風ハンバーグ 680円

・サーロインステーキ 1180円

**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

**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

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。

**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、-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、-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」） 課題完成時の画面（タブでインデントする）**

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

}

}

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」）**

|  |
| --- |
| ***Component*** |
| - name : String |
| + Component(name : String)  + print(tab : String) : void |

|  |
| --- |
| **File** |
|  |
| + File(name : String) |

|  |
| --- |
| **Folder** |
| - folder : List<Component> |
| + Component(name : String)  + Folder(name : String)  + add(component : Component) : Folder  + print(tab : String) : void |

# **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ー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。

**ガチャガチャマシーンの仕様**

|  |  |
| --- | --- |
| 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。