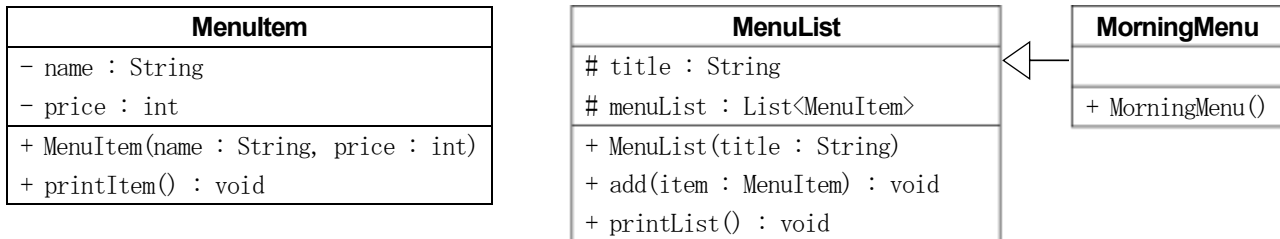


## ● J2Kad22D 「Composite パターンへの道① (単品表示と一覧表示)」

ファミレス ECC のメニュー表示を作成する。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) を使って、単品メニュー (モーニングセット、400 円) とモーニングメニュー一覧を表示する処理を作成せよ。



## 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
  
```

単品表示は「モーニングセット : 400」が表示されるようにする。

モーニングメニューは表示メソッドを呼び出せば、以下のメニューが表示される。

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

## ● J2Kad22C 「Composite パターンへの道② (インターフェイスの共通化)」

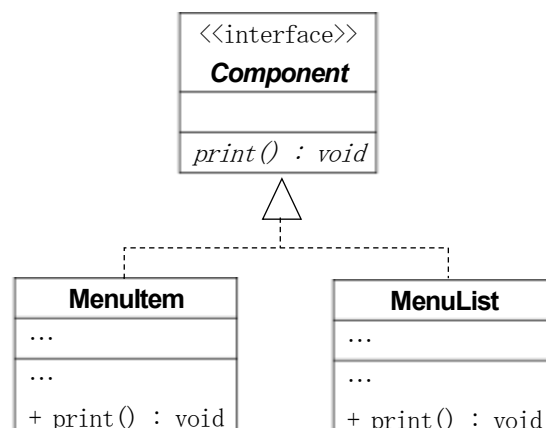
※ここからは src フォルダ直下のファイル「MenuList.java」を修正すること

Component インターフェイスを作成し、J2Kad22D の単品表示と一覧表示の表示処理を共通化せよ。

## 課題完成時の画面

(J2Kad22D と同じ)

## 課題完成時のクラス構成



## ● J2Kad22B 「Composite パターンへの道③ (容器と中身の同一視)」

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

## ● J2Kad22A 「ファミレス ECC (完成版)」 ※J2Kad22B をコピーして作成

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。

## 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

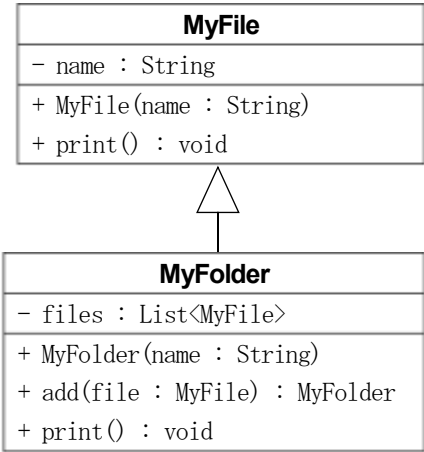
● J2Kad22S1「ディレクトリ表示」※J2Kad22S1 は新規作成

ファイル「MyFile.java」を新規作成し、ファイルを表す MyFile クラスとフォルダを表す MyFolder クラスを定義し、以下の構成のディレクトリ表示を行え。

ディレクトリ構成

ルート	サブフォルダ	ファイル
IdeaProjects	J2Kad01	J2Kad01A. java
		J2Kad01B. java
		J2Kad01C. java
	J2Kad02	J2Kad02A. java
		J2Kad02B. java
		J2Kad02C. java
	J2Kad03	J2Kad03A. java
		J2Kad03B. java
		J2Kad03C. java

クラス構成



リスト1：ディレクトリ表示（新規ファイル「J2Kad22S1.java」）

```
public class J2Kad22S {
    public static void main(String[] args) {
        MyFolder ip = new MyFolder("IdeaProjects");
        ファイルやサブフォルダを追加する
        ip.print(); // 表示
    }
}
```

課題完成時の画面

```
IdeaProjects
J2Kad01
J2Kad01A. java
J2Kad01B. java
J2Kad01C. java
J2Kad02
J2Kad02A. java
J2Kad02B. java
J2Kad02C. java
J2Kad03
J2Kad03A. java
J2Kad03B. java
J2Kad03C. java
```

ip.print で IdeaProjects とそのサブフォルダ、サブフォルダ内のファイルまですべて表示される。

### ● J2Kad22S2 「インデント表示」※J2Kad22S2 をコピーして作成

階層が深くなるごとにインデント（タブで字下げ）して表示するようにせよ。

ヒント：インデント表示用の print メソッドを追加する。

#### 課題完成時の画面（タブでインデントする）

```
IdeaProjects
  J2Kad01
    J2Kad01A. java
    J2Kad01B. java
    J2Kad01C. java
  J2Kad02
    J2Kad02A. java
    J2Kad02B. java
    J2Kad02C. java
  J2Kad03
    J2Kad03A. java
    J2Kad03B. java
    J2Kad03C. 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。

● J2Kad22X 「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。