

## ● J2Kad22D 「ECC ソフト株式会社① (委譲)」

ECC がソフト開発会社を立ち上げた！その名も「信頼と実績の ECC ソフト株式会社」、どんな課題も真摯に解決してくれるらしい。ただその実態は請け負った仕事をそのまま下請けに丸投げするだけの会社らしい。ECC ソフト株式会社 (ECCSoft クラス) が下請け会社に丸投げする処理 (work メソッド) を作成せよ。なお、現状、下請け会社は下請け A (SubA クラス) と下請け B (SubB クラス) の 2 社のみ。

## ECCSoft クラスの仕様 (work メソッドを作成すること)

メンバ	説明
public int subNum	丸投げ先の下請け会社の番号 (0 : 下請け A、1 : 下請け B)
public void setSubNum(int subNum)	subNum のセッター。
public void work()	仕事をする。ただしその実態は下請けへの丸投げ。 subNum が 0 のとき下請け A (SubA クラス) に仕事をさせる (work メソッド)。 subNum が 1 のとき下請け B (SubB クラス) に仕事をさせる (work メソッド)。 それ以外の値のとき「あれ？下請けがないよ～」と表示する。

## 課題完成時の画面

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 : 下請け A を使う、1 : 下請け B を使う、-1 : もういい) >0  
 下請け A : 何とかがんばってみます・・・

どうしますか？ (0 : 下請け A を使う、1 : 下請け B を使う、-1 : もういい) >1  
 下請け B : 下請けはつらいなー！

どうしますか？ (0 : 下請け A を使う、1 : 下請け B を使う、-1 : もういい) >2  
 あれ？下請けがないよー

## ● J2Kad22C 「ECC ソフト株式会社② (下請けの追加)」

業績好調な ECC ソフトではさらに下請けを増やしたい。ただし現状、下請けを増やすごとに ECCSoft クラスの修正が必要になる。下請けが増えても追加修正しなくて済むように ECCSoft クラスを修正せよ。さらに下請け C (SubC クラス、作成済み) を追加せよ。

## 課題完成時の画面

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 : 下請け A を使う、1 : 下請け B を使う、2 : 下請け C を使う、-1 : もういい) >2  
 下請け C : よろこんでお引き受けいたします！

## ● J2Kad22B 「スーパーコンピュータ ECC1000 (Strategy パターン)」

ECC ホームエレクトロニクスがスーパーコンピュータ ECC1000 (サウザンド) を開発した！1 から 10 までの整数を足し合わせる高度な演算を、アルゴリズムを切り換えて処理できるというスグレモノだ！ECC1000 クラスを作成し、アルゴリズム (SumAlg) を切り換えて動作するかどうか確認せよ。

## SumAlg インターフェイス (作成済み)

メソッド	仕様
void sum(int n)	1 から n までの整数を足し合わせた結果を表示する。

## 演算アルゴリズム (SumAlg インターフェイスを実装、作成済み)

クラス	特徴
NobitaAlg	「のび太」をモデルにしたアルゴリズム。計算が苦手。
SuneoAlg	「スネ夫」をモデルにしたアルゴリズム。普通に計算する。
DekisugiAlg	「出木杉」をモデルにしたアルゴリズム。頭が良すぎるため、物事を複雑に考える。
SizukaAlg	「しずか」をモデルにしたアルゴリズム。物事を別の角度からシンプルに考える。

## ECC1000 クラスの仕様 (作成すること)

メンバ	説明
private SumAlg alg	使用するアルゴリズム
public void setAlg(SumAlg alg)	アルゴリズムのセッター (フィールド alg に引数 alg を設定する)
public void sum(int n)	設定されたアルゴリズムに 1~n までの合計を計算させる。 設定されていないときは「アルゴリズムがセットされていません」と表示する。

## main メソッドの処理

- ① 使用するアルゴリズムを入力 (マイナスの値で終了)。
- ② 指定されたアルゴリズムを ECC1000 に設定する (1~4 以外ときは null を設定)。
- ③ 1 から 10 までの加算結果を表示して、①へ戻る。

## 課題完成時の画面

スーパーコンピュータ ECC1000 を開発した！  
アルゴリズムをセットすればどんな高度な計算でもできます！！  
計算アルゴリズムをセットしてください (0 : Nobita, 1 : Suneo, 2 : Dekisugi, 3 : Sizuka, -1 : 終了) >0  
のび太：え～っ、わかんないよ～

計算アルゴリズムをセットしてください (0 : Nobita, 1 : Suneo, 2 : Dekisugi, 3 : Sizuka, -1 : 終了) >1  
スネ夫：こんなの簡単さ。順番に足していけばいいんだ。  
スネ夫：1 を足して  
スネ夫：2 を足して  
      ⋮  
スネ夫：10 を足して  
スネ夫：答えは 55 だよ！

計算アルゴリズムをセットしてください (0 : Nobita, 1 : Suneo, 2 : Dekisugi, 3 : Sizuka, -1 : 終了) >4  
アルゴリズムがセットされていません！

## ● J2Kad22A 「ジョブチェンジ！」

ECC ソフトに依頼して RPG のジョブチェンジの処理を作成した。しかしジョブは選択できるがキャラクターの名前がどんどん変わっていく！まるでキャラクター（インスタンス）が入れ替わっているようだ！！キャラクターの名前が変わらないように（インスタンスが入れ替わらないように）処理を修正せよ。

## 課題作成前の画面

あなたは RPG のキャラクターです！  
今から世界を救う冒険に出かけます！  
でもその前に職業を選んでください！！

フックさん、こんにちは！  
ジョブチェンジしますか？（0：戦士、1：魔法使い、2：モンク、-1：これでいい）>0  
武器で攻撃します！  
盾で防御します！

ファラデーさん、こんにちは！  
ジョブチェンジしますか？（0：戦士、1：魔法使い、2：モンク、-1：これでいい）>1  
攻撃の魔法を唱えます！  
防御の魔法を唱えます！

オイラーさん、こんにちは！  
ジョブチェンジしますか？（0：戦士、1：魔法使い、2：モンク、-1：これでいい）>3  
攻撃方法を知らない！  
防御方法を知らない！

ジョブチェンジと同時に  
キャラクターの名前も  
変わる。

## 課題完成時の画面

あなたは RPG のキャラクターです！  
今から世界を救う冒険に出かけます！  
でもその前に職業を選んでください！！

フリーエさん、こんにちは！  
ジョブチェンジしますか？（0：戦士、1：魔法使い、2：モンク、-1：これでいい）>0  
武器で攻撃します！  
盾で防御します！

フリーエさん、こんにちは！  
ジョブチェンジしますか？（0：戦士、1：魔法使い、2：モンク、-1：これでいい）>1  
攻撃の魔法を唱えます！  
防御の魔法を唱えます！

フリーエさん、こんにちは！  
ジョブチェンジしますか？（0：戦士、1：魔法使い、2：モンク、-1：これでいい）>3  
攻撃方法を知らない！  
防御方法を知らない！

ジョブチェンジしても  
キャラクターの名前は  
変わらない。  
（変わるのはジョブの  
み）

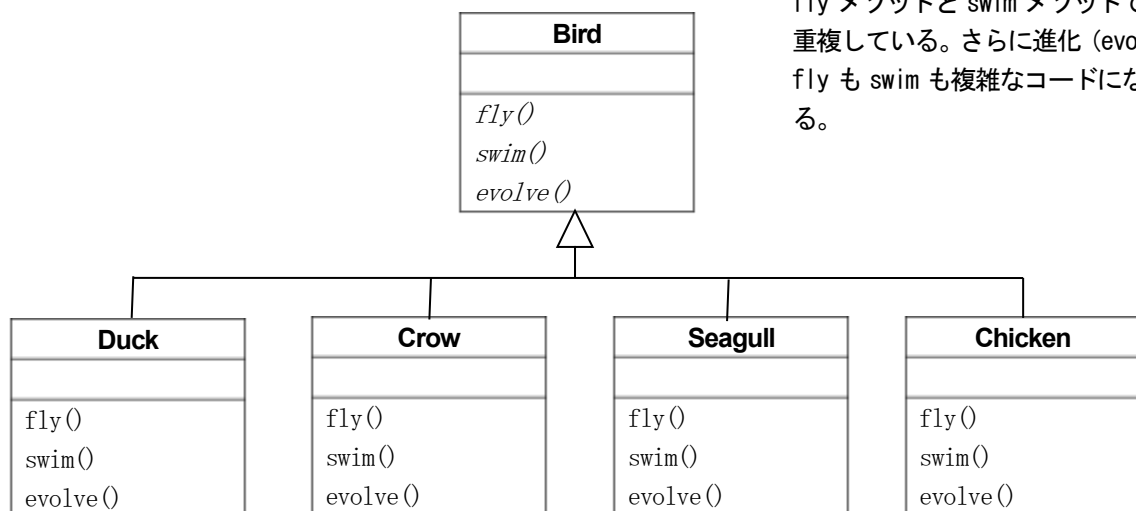
## ● J2Kad22S 「ポケット Duck !」

世界にはばたく ECC ゲームスでは野鳥をゲットして収集するオリジナルゲーム「ポケット Duck !」を制作している。登場する野鳥は4種類 (アヒル・カラス・カモメ・ニワトリ)、それぞれ飛ぶ (fly)・泳ぐ (swim)・進化する (evolve) のコマンドに対応している。プログラム開発を ECC ソフトへ発注したところ、「飛ぶ」と「泳ぐ」まで制作できたが、「進化する」を追加するという段階で難航しているという報告が来た。今後、さらにさまざまな仕様 (例えば「ワープする」とか「深海まで潜る」など) を追加していくことを考えると、このままの設計では「のっぴきならない」 (←意味がわからない人は検索) ことになる! 今後の開発を順調に進めるためにも、簡潔かつ修正のしやすい設計になるようにクラス設計を見直し、進化 (evolve) を追加せよ。

## 野鳥の仕様 (Bird クラスを継承)

クラス	飛ぶ (fly)	泳ぐ (swim)	進化する (evolve)
Duck (アヒル)	飛べない	水面を泳ぐ	空を飛べるようになる
Crow (カラス)	空を飛ぶ	泳げない	ジェット噴射で飛べるようになる
Seagull (カモメ)	空を飛ぶ	水面を泳ぐ	潜れるようになる
Chicken (ニワトリ)	飛べない	泳げない	飛べるようになる&泳げるようになる

## 課題作成前のクラス構成



fly メソッドと swim メソッドで同じコードが重複している。さらに進化 (evolve) させると fly も swim も複雑なコードになる可能性がある。

## 課題完成時の画面

どの野鳥をゲットしますか? (0: アヒル、1: カラス、2: カモメ、3: ニワトリ、-1: 終了) >0  
 ぼくアヒル!  
 飛べないけど泳げるよ!

何をさせますか? (0: 飛ぶ、1: 泳ぐ、2: 進化、-1: 終了) >0  
 う〜ん、飛べない!!

何をさせますか? (0: 飛ぶ、1: 泳ぐ、2: 進化、-1: 終了) >1  
 スイスイ! 水面を泳いでいます!!

何をさせますか? (0: 飛ぶ、1: 泳ぐ、2: 進化、-1: 終了) >2  
 空を飛べるようになった!!

何をさせますか? (0: 飛ぶ、1: 泳ぐ、2: 進化、-1: 終了) >0  
 バタバタ! がんばって飛んでいます!!

● J2Kad22X「ファミレス ECC」※冬休みの宿題、年明けに解答編の予定

世界にはばたく ECC フーズは「ファミレス ECC」を出店することになった。メニューは朝食・昼食・夕食で異なり、それぞれにドリンクやデザートなどのサブメニューもある。以下の仕様でメニュー表示する処理を作成せよ。

※ 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 (Strategy) とは関係ないので注意すること (ヒント参照)。

メニューの仕様

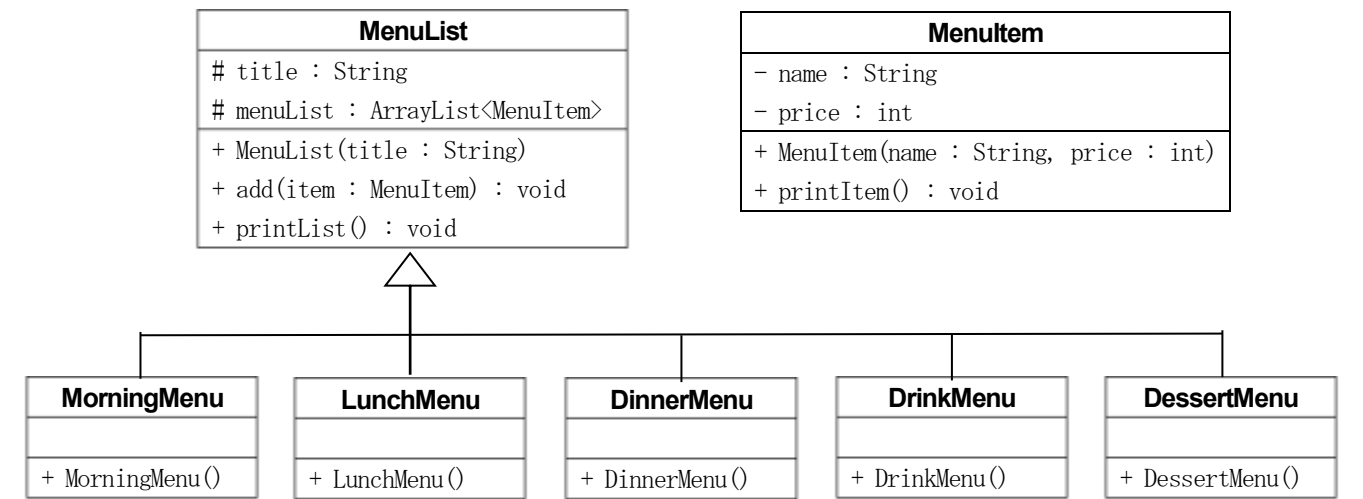
ランチにはサブメニューとしてドリンク、ディナーにはドリンクとデザートがある。

<div><div>MorningMenu</div><div><div>・モーニングセット400 円</div><div>・焼き魚定食600 円</div></div></div>	<div><div>LunchMenu</div><div><div>・日替わりランチ680 円</div><div>・シェフのおすすめランチ890 円</div></div><div><div>DrinkMenu</div><div><div>・コーヒー220 円</div><div>・紅茶220 円</div></div></div></div>	<div><div>DinnerMenu</div><div><div>・和風ハンバーグ680 円</div><div>・サーロインステーキ1180 円</div></div><div><div>DrinkMenu</div><div><div>・コーヒー220 円</div><div>・紅茶220 円</div></div></div><div><div>DessertMenu</div><div><div>・チーズケーキ350 円</div><div>・アイスクリーム150 円</div></div></div></div>
---	--	---

メニュー表示の仕様

入力	表示するメニュー
0	朝食メニュー (MorningMenu) を表示する。
1	昼食メニュー (LunchMenu) + ドリンクメニュー (DrinkMenu) を表示する。
2	夕食メニュー (DinnerMenu) + ドリンクメニュー (DrinkMenu) + デザートメニュー (DessertMenu) を表示する。
3	上記0、1、2 のすべてを表示する。

課題作成前のクラス構成



## 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) >0

★ モーニングメニュー ★

モーニングセット : 400

焼き魚定食 : 600

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

★ 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、3 : すべて、-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、3 : すべて、-1 : 終了) >3

★★★★ すべてのメニュー ★★★★★

⋮  
⋮  
⋮

(0 : モーニング、1 : ランチ、2 : ディナーのすべてのメニューを表示する)

ヒント :

「Composite 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メニュー表示ができていれば本課題は OK とする。