● J2Kad02D1「ピカチュウクラス！」（入門編 P.134「クラスの宣言」）

J2Kad02D1 に J2Kad01C 完成版相当のプログラムが準備されている。ピカチュウに関する処理を Pika クラス（新規ファイル「Pika.java」）として独立させよ。

リスト１:「ピカチュウ現る！」（ファイル「J2Kad02D1.java」）

```
public class J2Kad02D1 {
    public static String name = "ピカチュウ"; // 名前
    public static int hp = 20;                // 体力
    public static void showData() {…}
    public static void walk() {…}
    public static void sleep() {…}
    public static void main(String[] args) {
        showData();
        System.out.println(name + "を散歩させます！");
        walk();
        walk();
        walk();
        showData();
        System.out.println(name + "を眠らせます！");
        sleep();
        sleep();
        sleep();
        showData();
    }
}
```

Pika クラスへ持って行く

Pika クラスのメンバを
使うように変更する

課題完成時の画面

```
ぼくの名前はピカチュウ、HP は 20 だよ！
ピカチュウを散歩させます！
てくてく・・・
てくてく・・・
てくてく・・・
ぼくの名前はピカチュウ、HP は 17 だよ！
ピカチュウを眠らせます！
ぐうぐう・・・
ぐうぐう・・・
ぐうぐう・・・
ぼくの名前はピカチュウ、HP は 20 だよ！
```

● J2Kad02D2「ヤドンクラス！」

ヤドン（Yadon クラス）を新規作成し、散歩させて眠らせる処理を作成せよ。

Yadon クラスの仕様（新規ファイル「Yadon.java」）
フィールド、メソッドともに Pika クラスと同じ。ただし name の初期値は「ヤドン」、hp の初期値は 30 とする。

main メソッドの仕様（J2Kad02D2 クラスに作成）
J2Kad02D1 と同じ。ただし Pika クラスの代わりに Yadon クラスを使う。

課題完成時の画面

```
ぼくの名前はヤドン、HP は 30 だよ！
ヤドンを散歩させます！
てくてく・・・
てくてく・・・
てくてく・・・
ぼくの名前はヤドン、HP は 27 だよ！
ヤドンを眠らせます！
ぐうぐう・・・
ぐうぐう・・・
ぐうぐう・・・
ぼくの名前はヤドン、HP は 30 だよ！
```

● J2Kad02C「モンスタークラス！」

（P.138「インスタンスの生成」、P.139「インスタンス変数」、P.171「インスタンスメソッドとは」）

どのモンスターでも使えるMonsterクラスを作成し、ディアルガ（体力：1000）とコイキング（体力：1）を散歩させる処理を作成せよ。

Monsterクラスの仕様（新規ファイル「Monster.java」）

	書式（staticはつけないこと）	説明
フィールド	public String name	名前（初期値：設定しない）
	public int hp	体力（初期値：設定しない）
メソッド	public void setData(String n, int h)	名前にnを、体力にhを代入する。
	public void showData()	名前と体力を表示する。「ぼくの名前は～！HPはxxだよ！」
	public void walk()	体力が0以下のとき「つかれて歩けないよ～」と表示する。 そうでないとき「てくてく・・・」と表示し、体力を1減らす。
	public void sleep()	眠らせる。「ぐうぐう・・・」と表示したのち、体力を1増やす。

mainメソッドの仕様（J2Kad02Cクラスに作成）

（課題完成時の画面を参考に作成すること）

課題完成時の画面

```
ぼくの名前はディアルガ、HPは1000だよ！
ぼくの名前はコイキング、HPは1だよ！
ディアルガを散歩させます！
てくてく・・・
てくてく・・・
てくてく・・・
ぼくの名前はディアルガ、HPは997だよ！
コイキングを散歩させます！
てくてく・・・
つかれて歩けないよ～
つかれて歩けないよ～
ぼくの名前はコイキング、HPは0だよ！
```

● J2Kad02B「基本型と参照型」（入門編 P.142「参照型」）

以下の処理を作成し、動作確認せよ。

作成するメソッド（J2Kad02B クラス）

書式	仕様
public static void addInt(int x)	引数 x に 5 を加算し、「x に 5 を加算しました！」と表示する。
public static void addArray(int[] b)	配列 b の全要素に 5 を加算し、「b[#]に 5 を加算しました！」と表示する。（#は要素番号）

main メソッドの仕様①

① int 型変数 x（初期値：10）を宣言する。
② x を引数にして addInt メソッドを呼び出す。
③ x の値を表示する。

main メソッドの仕様②

④ int 配列 a（要素数 3）を生成する。
⑤ a[0]に 0、a[1]に 1、a[2]に 2 を代入する。
⑥ 配列 a を引数にして addArray メソッドを呼び出す。
⑦ 配列 a の各要素の値を表示する。

課題完成時の画面

```
x：10
x に 5 を加算しました！
x：10

a：10 20 30
b[0]に 5 を加算しました！
b[1]に 5 を加算しました！
b[2]に 5 を加算しました！
a：15 25 35
```

仕様①

仕様②

int 型変数 x の値は addInt メソッドの前後で変わらないが、配列 a の値は addArray メソッドの前後で変化している。

● J2Kad02A「参照渡し」

Monster クラスを使ってピカチュウ（体力：10～19）とイワンコ（体力：10～19）を生成、useMonster メソッド（新規作成）を使って動かせ（**課題完成時の画面**を参照）。

作成するメソッド（J2Kad02A クラス）

書式	仕様
public static void useMonster(Monster m)	モンスターm に対して以下の処理を行う。 ① 3 回散歩させて、データを表示する。 ② 3 回眠らせて、データを表示する。

main メソッドの仕様

① ピカチュウとイワンコを生成し、データを表示する。
② ピカチュウを動かす（useMonster メソッドを呼び出す）。
③ イワンコを動かす（useMonster メソッドを呼び出す）。

課題完成時の画面

```
ぼくの名前はピカチュウ、HP は 18 だよ！
ぼくの名前はイワンコ、HP は 18 だよ！

ピカチュウを散歩させます！
てくてく・・・
てくてく・・・
てくてく・・・
ぼくの名前はピカチュウ、HP は 15 だよ！
ピカチュウを眠らせます！
ぐうぐう・・・
ぐうぐう・・・
ぐうぐう・・・
ぼくの名前はピカチュウ、HP は 18 だよ！
イワンコを散歩させます！
てくてく・・・
てくてく・・・
てくてく・・・
ぼくの名前はイワンコ、HP は 15 だよ！
イワンコを眠らせます！
ぐうぐう・・・
ぐうぐう・・・
ぐうぐう・・・
ぼくの名前はイワンコ、HP は 18 だよ！
```

● J2Kad02S「そうだ！ECC 銀行へ行こう！！②」

銀行口座を表す Account クラスを作成し、のび太とスネ夫が信頼と実績の ECC 銀行へ行く処理を作成せよ。必要なフィールドは各自で考えること。

Account クラスの仕様（新規作成）

	書式	説明
フィールド	public String name	口座名義
	public int accountNumber	口座番号（7 桁）
	public int money	預金残高
	public int secretNumber	暗証番号（4 桁）
メソッド	public void setData(String n, int a, int m, int s)	n：口座名義、a：口座番号、m：預金残高、s：暗証番号を対応するフィールドに設定する。
	public void showData()	口座情報（口座名義、口座番号、口座残高）を表示する。

作成するメソッド（J2Kad02S クラス）

書式	仕様
public static void gotoECCBank(Account account)	処理の対象となる銀行口座を引数 account として受け取る。（処理内容は J2Kad01S と同じ）
public static void deposit(Account account)	
public static void withdraw(Account account)	

main メソッドの仕様

① のび太とスネ夫の銀行口座を作成する（値は各自で設定する）。
② のび太またはスネ夫を選択し、gotoECCBank メソッドを呼び出す。

課題完成時の画面

```
そうだ！銀行へ行こう！！
誰が行きますか？（1：のび太、2：スネ夫、-1：誰もいかない）＞2
信頼と実績の ECC 銀行へようこそ！
口座名義：スネ夫
口座番号：8901234
預金残高：10000000 円
どうしますか？（1：預ける、2：引き出す、-1：帰る）＞2
暗証番号を入力してください＞5678
いくら引き出しますか？＞5000000

口座名義：スネ夫
口座番号：8901234
預金残高：5000000 円
どうしますか？（1：預ける、2：引き出す、-1：帰る）＞-1
ありがとうございました！

誰が行きますか？（1：のび太、2：スネ夫、-1：誰もいかない）＞-1
```

基本的に J2Kad01S と同じ処理。銀行へ行く人（口座）を選択する処理が追加されている。

なお、左ではスネ夫のデータは以下の通り。
- 口座名義：スネ夫
- 口座番号 8901234
- 預金残高：10000000
- 暗証番号：5678

● J2Kad02X「スタック！②」

J2Kad01X のスタック操作に関する処理を Stack クラスとして取り出せ。格納できるデータ数は 10 個とする。なお、スタックオーバーフローなどの不具合が発生しないように、スタックのデータ数をチェックする処理も追加すること。

Stack クラスのメソッド（フィールドは各自で考えること）

書式	説明
public void push(int data)	スタックにデータ（data）を格納する。
public int pop()	スタックからデータを取り出し値を返す。

main メソッドの仕様（課題完成時の画面を参考）

① スタックを生成し、「-1：終了」が選択されるまで以下の処理を行う。
② 「1：push」を選択するとスタックに 0～99 までの値（乱数で決定）を格納する。ただしこれ以上データを格納できないときは「スタックがいっぱいです！」と表示する。
③ 「2：pop」を選択するとスタックからデータを取り出す。ただしデータがないときは「データがありません！」と表示する。

課題完成時の画面

```
スタック操作をします！
どうしますか？（1：push、2：pop、-1：終了）＞1
stack：4

どうしますか？（1：push、2：pop、-1：終了）＞1
stack：4 51

どうしますか？（1：push、2：pop、-1：終了）＞1
stack：4 51 26

どうしますか？（1：push、2：pop、-1：終了）＞1
stack：4 51 26 34

どうしますか？（1：push、2：pop、-1：終了）＞1
stack：4 51 26 34 71

どうしますか？（1：push、2：pop、-1：終了）＞1
stack：4 51 26 34 71 87
      ⋮
   （中略）
      ⋮
どうしますか？（1：push、2：pop、-1：終了）＞1
stack：4 51 26 34 71 87 18 5 17 94

どうしますか？（1：push、2：pop、-1：終了）＞1
スタックがいっぱいです！
stack：4 51 26 34 71 87 18 5 17 94
```

（続き）

```
どうしますか？（1：push、2：pop、-1：終了）＞2
94 を取り出しました！
stack：4 51 26 34 71 87 18 5 17

どうしますか？（1：push、2：pop、-1：終了）＞2
17 を取り出しました！
stack：4 51 26 34 71 87 18 5

どうしますか？（1：push、2：pop、-1：終了）＞2
5 を取り出しました！
stack：4 51 26 34 71 87 18
      ⋮
   （中略）
      ⋮
どうしますか？（1：push、2：pop、-1：終了）＞2
51 を取り出しました！
stack：4

どうしますか？（1：push、2：pop、-1：終了）＞2
4 を取り出しました！
stack：

どうしますか？（1：push、2：pop、-1：終了）＞2
データがありません！
stack：

どうしますか？（1：push、2：pop、-1：終了）＞-1
```