

● J2Kad03D1 「コンストラクタ」

モンスターを表す `Monster` クラスが準備されている。**リスト 1** を入力して実行したところ、最初の `setData` メソッドの呼び出しで、名前が「`null`」、HP が「0」と表示される。名前と HP にデフォルト値（「まだないよ～」と「1」）が設定されるように `Monster` クラスにコンストラクタを追加せよ。

追加するメソッド (Monster クラス)

書式	仕様
<code>public Monster()</code>	コンストラクタ。 <code>setData</code> メソッドを使って <code>name</code> に「まだないよ～」、 <code>hp</code> に 1 を設定する。

リスト 1 : J2Kad03D1 クラス

```
public class J2Kad03D1 {
    public static void main(String[] args) {
        Monster m1 = new Monster();
        m1.showData();
        m1.setData("ピカチュウ", 20);
        m1.showData();
    }
}
```

リスト 1 まで入力したときの画面

ぼくの名前は `null`、HP は 0 だよ！
ぼくの名前はピカチュウ、HP は 20 だよ！

課題完成時の画面

ぼくの名前はまだないよ～、HP は 1 だよ！
ぼくの名前はピカチュウ、HP は 20 だよ！

● J2Kad03D2 「引数付きコンストラクタ」

ヤドン (HP : 30) を生成してデータ表示する処理を作成せよ。なおヤドンのデータ (名前と HP) の設定は、引数付きコンストラクタで行うこと。

追加するメソッド (Monster クラス)

書式	仕様
<code>public Monster(String n, int h)</code>	引数付きコンストラクタ。 <code>setData</code> メソッドを使って <code>name</code> に <code>n</code> 、 <code>hp</code> に <code>h</code> を設定する。

課題完成時の画面

ぼくの名前はヤドン、HP は 30 だよ！

● J2Kad03C 「カプセル化」

リスト 1 を入力して実行すると、ピカチュウの名前が「ゲレゲレ」に、HP が「-1」になる。他のクラス（ここでは J2Kad03C クラス）から勝手にピカチュウの名前や HP を変更できないようにせよ。

追加するメソッド (Monster クラス)

書式	仕様
public void setName(String name)	name のセッター。引数 name が「ゲレゲレ」のときは「ゲレゲレなんていやだ～」と表示して、名前を設定しない。
public void setHp(int hp)	hp のセッター。引数 hp が 0 以下のときは「せめて HP、1 はちょうだい！」と表示して、1 を設定する。

リスト 1 : J2Kad03C クラス

```
public class J2Kad03C {
    public static void main(String[] args) {
        Monster m1 = new Monster("ピカチュウ", 20);
        m1.showData();
        m1.name = "ゲレゲレ";
        m1.hp = -1;
        m1.showData();
    }
}
```

リスト 1 まで入力したときの画面

ぼくの名前はピカチュウ、HP は 20 だよ！
ぼくの名前はゲレゲレ、HP は-1 だよ！

課題完成時の画面

ぼくの名前はピカチュウ、HP は 20 だよ！
ゲレゲレなんて名前はいやだ～
せめて HP、1 はちょうだい！
ぼくの名前はピカチュウ、HP は 1 だよ！

● J2Kad03B 「アクセサ」

Profile クラスを作成し、ドラえもんとドラミちゃんのプロフィール（右表）を表示する処理を作成せよ。

	身長	体重
ドラえもん	123.9	123.9
ドラミちゃん	100.0	91.0

- ① J2Kad03B クラスの main メソッドでドラえもんとドラミちゃんのインスタンスを生成し、それぞれのフィールドを直接参照して表示する処理を作成せよ。

Profile クラス（新規作成）の仕様①

書式	説明
public String name	名前
public double height	身長
public double weight	体重
public Profile(String name, double height, double weight)	コンストラクタ。引数を対応するフィールドに設定する。

①まで完成したときの画面

ドラえもんのプロフィールです！
身長：129.3
体重：129.3
ドラミちゃんのプロフィールです！
身長：100.0
体重：91.0

J2Kad03B クラスから直接、身長や体重を参照するので設定した値がそのまま表示される。

- ② Profile クラスのフィールドをすべて private にし、ゲッターを使って取得するように変更せよ。

Profile クラスの仕様②

書式	説明
public String getName()	name のゲッター。name を返す。
public double getHeight()	height のゲッター。height が 120 より小さいときは 170 を返す（身長をごまかす）。
public double getWeight()	weight のゲッター。weight が 100 より大きいときは 50 を返す（体重をごまかす）

②まで完成したときの画面

ドラえもんのプロフィールです！
身長：129.3
体重：50.0
ドラミちゃんのプロフィールです！
身長：170.0
体重：91.0

Profile クラスに教えてもらった値（セッターを使って取得した値）を表示する。ドラえもんは体重を、ドラミちゃんは身長をごまかしている。

● J2Kad03A 「そうだ！ECC 銀行へ行こう！！③」

J2Kad03A クラスと Account クラスに J2Kad02S の完成版相当のプログラムが準備されている。ただし Account クラスのフィールド（口座名義・口座番号・預金残高・暗証番号）は他のクラス（J2Kad03A クラス）から直接変更できる状態になっている（暗証番号もまる見えだ！）。Account クラスのフィールドを `private` に変更せよ。なお、この変更に伴う修正は各自で行うこと。

追加するメソッド（Account クラス）

書式	説明
<code>public Account(String name, int accountNumber, int money, int secretNumber)</code>	コンストラクタ。引数を対応するフィールドに設定する。
<code>public String getName()</code>	口座名義のゲッター。
<code>public int getAccountNumber()</code>	口座番号のゲッター。
<code>public int money()</code>	預金残高のゲッター。
<code>public addMoney(int money)</code>	預金残高に引数 <code>money</code> を加算する。
<code>public boolean subMoney(int money)</code>	預金残高より引数 <code>money</code> が大きいとき <code>false</code> を返す（残高不足）。そうでないとき預金残高から引数 <code>money</code> を減らし、 <code>true</code> を返す。
<code>public boolean checkSecretNumber(int secretNumber)</code>	暗証番号と引数 <code>secretNumber</code> が一致していたら <code>true</code> を返す。一致していなかったら <code>false</code> を返す。 ※暗証番号についてはゲッターを作らない。

課題完成時の画面

（J2Kad02S と同じ）

● J2Kad03S 「3 回目のおつかい①」

Stack クラス (J2Kad02X 完成版相当) が準備されている。int 型ではなく String 型データを格納するように修正し、のび太がおつかいへ行って、逆順に戻ってくる処理を作成せよ。なお、のび太の記憶力 (スタックサイズ) は 5 とする。

ヒント: 次の場所へ移動するとき、戻ってくる場所 (今いる場所) をスタックに積んでいく (メソッド呼び出しと同じしくみ)。

課題完成時の画面

のび太がおつかいに行きます！
のび太の家です。どこへ行きますか？ (0 : ECC コーヒー、1 : ECC バーガー、2 : ECC ドーナツ、-1 : 戻る) >0
ECC コーヒーです。どこへ行きますか？ (0 : ECC コーヒー、1 : ECC バーガー、2 : ECC ドーナツ、-1 : 戻る) >2
ECC ドーナツです。どこへ行きますか？ (0 : ECC コーヒー、1 : ECC バーガー、2 : ECC ドーナツ、-1 : 戻る) >1
ECC バーガーです。どこへ行きますか？ (0 : ECC コーヒー、1 : ECC バーガー、2 : ECC ドーナツ、-1 : 戻る) >0
ECC コーヒーです。どこへ行きますか？ (0 : ECC コーヒー、1 : ECC バーガー、2 : ECC ドーナツ、-1 : 戻る) >1
ECC バーガーです。どこへ行きますか？ (0 : ECC コーヒー、1 : ECC バーガー、2 : ECC ドーナツ、-1 : 戻る) >2
これ以上進むと帰れないよ～
ECC バーガーです。どこへ行きますか？ (0 : ECC コーヒー、1 : ECC バーガー、2 : ECC ドーナツ、-1 : 戻る) >-1
もと来た道に戻ります！
ECC コーヒーに戻ってきた！
ECC バーガーに戻ってきた！
ECC ドーナツに戻ってきた！
ECC コーヒーに戻ってきた！
のび太の家に戻ってきた！

● J2Kad03X 「3 回目のおつかい②」 ※J2Kad03S のコードをコピーして作成

J2Kad03S で場所移動をしたときインデント（タブ 1 つ分）を行うように処理を追加せよ。

課題完成時の画面（文字が小さいがインデント以外は J2Kad03S と同じ）

```
のび太がおつかいに行きます！
のび太の家です。どこへ行きますか？（0：ECC コーヒー、1：ECC バーガー、2：ECC ドーナツ、-1：戻る）>0
  ECC コーヒーです。どこへ行きますか？（0：ECC コーヒー、1：ECC バーガー、2：ECC ドーナツ、-1：戻る）>2
    ECC ドーナツです。どこへ行きますか？（0：ECC コーヒー、1：ECC バーガー、2：ECC ドーナツ、-1：戻る）>1
      ECC バーガーです。どこへ行きますか？（0：ECC コーヒー、1：ECC バーガー、2：ECC ドーナツ、-1：戻る）>0
        ECC コーヒーです。どこへ行きますか？（0：ECC コーヒー、1：ECC バーガー、2：ECC ドーナツ、-1：戻る）>1
          ECC バーガーです。どこへ行きますか？（0：ECC コーヒー、1：ECC バーガー、2：ECC ドーナツ、-1：戻る）>2
            これ以上進むと帰れないよ～
            ECC バーガーです。どこへ行きますか？（0：ECC コーヒー、1：ECC バーガー、2：ECC ドーナツ、-1：戻る）>-1
              もと来た道に戻ります！
              ECC コーヒーに戻ってきた！
              ECC バーガーに戻ってきた！
              ECC ドーナツに戻ってきた！
              ECC コーヒーに戻ってきた！
            のび太の家に戻ってきた！
```