

クラス型インスタンスの配列

次に、Day型インスタンスの配列を作ることになります。キーボードから読み込んだ値を要素数とする配列を生成して、全要素の日付を2017年10月15日に設定・表示するプログラムをList 9-7に示します。

List 9-7

day1/DayArrayError.java

```
// 日付クラスDay【第1版】の配列（その1：実行時エラー）
import java.util.Scanner;

class DayArrayError {

    public static void main(String[] args) {
        Scanner stdIn = new Scanner(System.in);
        String[] wd = {"日", "月", "火", "水", "木", "金", "土"};

        System.out.print("日付は何個：");
        int n = stdIn.nextInt();

        1 → Day[] a = new Day[n]; // 要素数n個のDay型配列
        2 → for (int i = 0; i < a.length; i++)
            a[i].set(2017, 10, 15); // 全要素を2017年10月15日に設定

        for (int i = 0; i < a.length; i++)
            System.out.println("a[" + i + "] = " + a[i].getYear() + "年"
                + a[i].getMonth() + "月" + a[i].getDate() + "日("
                + wd[a[i].dayOfWeek() + "]");
    }
}
```

誤

実行例

```
日付は何個：3
Exception in thread "main" java.lang.NullPointerException
at Day1.DayArrayError.main(DayArrayError.java:17)
```

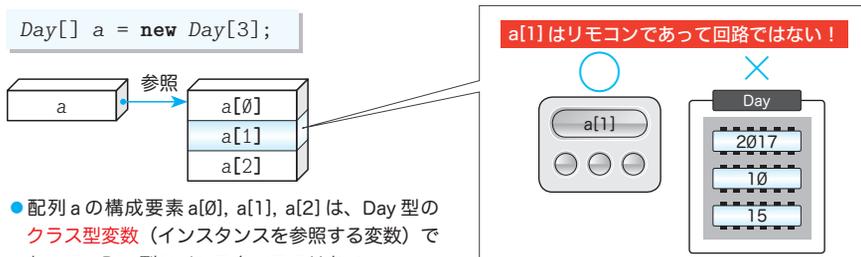
実行すると、日付の個数nを読み込んだ直後に

実行時エラーが発生します(第16章)。1でa用の配列本体を生成しているのに、なぜエラーが発生するのでしょうか。

Fig.9-7に示すように、要素a[1]は、Dayを参照するクラス型変数(リモコン)であって、日付のインスタンス(本体)ではありません。もちろん、a[0]とa[2]も同様です。

▶ ここに示す図は、要素数が3である例です。配列aは、日付本体の配列ではなく、3個のリモコンが集まった配列です。配列の生成時に、各要素はnullで初期化されます(p.183)。

2の実行時にエラーが発生するのは、何も参照していない空参照であるa[i]に対して、メソッドsetを呼び出そうとするからです。



- 配列aの構成要素a[0], a[1], a[2]は、Day型のクラス型変数(インスタンスを参照する変数)であって、Day型のインスタンスではない。

Fig.9-7 Day型配列の構成要素

個々の日付インスタンスは、クラス型変数とは別に、**new** 演算子を用いて生成する必要があります。したがって、正しいプログラムは **List 9-8** となります。

▶ **List 9-7** の **2** を **List 9-8** の網かけ部に置きかえます（それ以外は同じですから省略します）。

List 9-8

day1/DayArray.java

```
// 日付クラスDay【第1版】の配列（その2）
```

```
for (int i = 0; i < a.length; i++)
    a[i] = new Day(2017, 10, 15);
```

実行例

```
日付は何個：3
a[0] = 2017年10月15日(日)
a[1] = 2017年10月15日(日)
a[2] = 2017年10月15日(日)
```

正

この **for** 文で行うことを分解して示したのが **Fig.9-8** です。繰返しのたびに **Day** 型インスタンスを生成して2017年10月15日で初期化します。そして、生成したインスタンスへの参照が、代入演算子によって **a[i]** に代入されます。

重要 クラス型のインスタンスの配列を利用するためには、クラス型変数の配列に加えて、個々の要素のインスタンスも生成しなければならない。

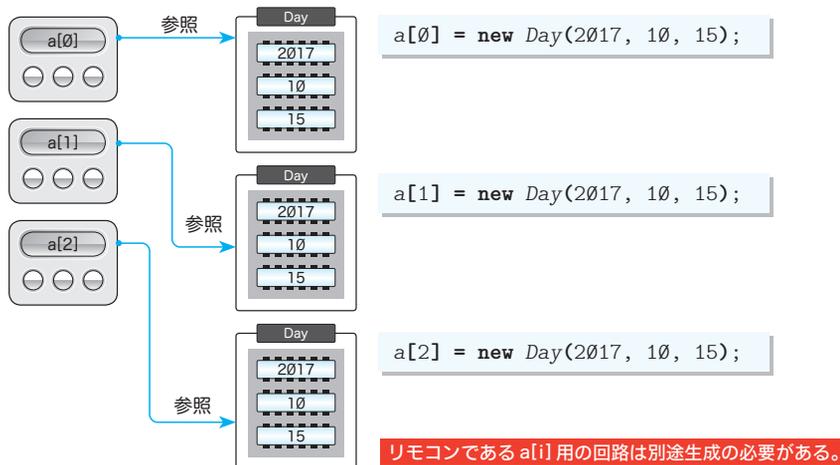


Fig.9-8 Day 型配列の構成要素の個々のインスタンスの生成

なお、プログラムを以下のようにすることはできません。コンストラクタには三つの **int** 型引数を渡さなければならないからです。

```
for (int i = 0; i < a.length; i++) // とりあえず生成
    a[i] = new Day(); // エラー
for (int i = 0; i < a.length; i++) // その後で
    a[i].set(2017, 10, 15); // 値を設定
```

▶ 日付クラスの第2版では、このコードが動作するようになります（p.317）。