

● JKad20D 「for 文！」

リスト1はECC長屋の住人の名前を表示する処理である。while文をfor文に置き換えよ。

リスト1：ECC長屋の住人（ファイル「JKad20D.java」）

```
public class JKad20D {
    public static void main(String[] args) {
        String[] nagayaECC = {
            "ピタゴラス",
            "アルキメデス",
            "ユークリッド",
            "エラトステネス",
            "フィボナッチ",
        };
        int i = 0;           // 部屋番号
        while(i < nagayaECC.length) {
            System.out.println(i + "号室：" + nagayaECC[i]);
            i++;
        }
    }
}
```

課題完成時の画面

```
0号室：ピタゴラス
1号室：アルキメデス
2号室：ユークリッド
3号室：エラトステネス
4号室：フィボナッチ
```

● JKad20C1 「偶数・奇数判定！」

1から100までの数値に対して、偶数か奇数かを判定する処理をfor文を使って作成せよ。

課題完成時の画面

```
1：奇数です！
2：偶数です！
3：奇数です！
4：偶数です！
5：奇数です！
6：偶数です！
  ⋮
（中略）
  ⋮
95：奇数です！
96：偶数です！
97：奇数です！
98：偶数です！
99：奇数です！
100：偶数です！
```

● JKad20C2 「1 から n まで足してみよう ! ⑥ (for 文編)」

1 から入力された数値 n まで加算する処理を for 文を使って作成せよ。

課題完成時の画面

1 より大きい整数を入力してください>10
1 から 10 まで加算します !
合計は 55 です !!

● JKad20B 「カードを表示しよう ! ② (逆順表示)」

以下の処理を作成せよ。

- ① 要素数 9 の配列を宣言し、乱数を使って 0 から 99 までの値を設定する。
- ② 配列の先頭から最後まで値を表示する。
- ③ 配列の最後から先頭までの値を表示する (逆順)。

課題完成時の画面

カードを表示します !
cards[0] : 72
cards[1] : 48
cards[2] : 15
cards[3] : 20
cards[4] : 63
cards[5] : 97
cards[6] : 49
cards[7] : 54
cards[8] : 72
逆順で表示します !
cards[8] : 72
cards[7] : 54
cards[6] : 49
cards[5] : 97
cards[4] : 63
cards[3] : 20
cards[2] : 15
cards[1] : 48
cards[0] : 72

● JKad20A 「カードを逆順に並べ替えよう！」

入力された枚数のカード（配列、値は乱数で 0～99）を生成し、逆順に並び替える処理を作成せよ。なお、以下のメソッドを作成し、データを入れ替えるたびにカードの並びを表示すること。

書式	処理
<code>public static void showCards(int[] cards)</code>	配列 cards の先頭から最後まで要素を表示する。

課題完成時の画面（10 を入力したとき）

カードの枚数を入力してください>10
最初の並びを表示します！
60 45 30 36 50 84 13 28 68 88
逆順にします！
88 45 30 36 50 84 13 28 68 60
88 68 30 36 50 84 13 28 45 60
88 68 28 36 50 84 13 30 45 60
88 68 28 13 50 84 36 30 45 60
88 68 28 13 84 50 36 30 45 60

逆順に表示するのではなく、実際にデータを逆順に並べ替えること。

	0	1	2	3	4	5	6	7	8	9
cards	60	45	30	36	50	84	13	28	68	88
				↓						
cards	88	68	28	13	84	50	36	30	45	60

データを入れ替えるたびに showCards メソッドでカードの並びを表示する。

● JKad20S 「出席率を計算しよう！」

出席率を計算する処理を作成せよ。

- ① 第1週から第15週までの出欠状況（0：出席、1：欠席、2：遅刻）を入力する。
- ② 出席・欠席・遅刻それぞれの回数を表示する。
- ③ 出席率を計算し小数第2位で四捨五入して表示する。出席率を計算する際には遅刻3回で欠席1回とみなすこと。

課題完成時の画面（出席率が75%以上のとき）

出席状況を入力してください。
第1週目：（0：出席、1：欠席、2：遅刻）>0
第2週目：（0：出席、1：欠席、2：遅刻）>1
⋮
（中略）
⋮
第14週目：（0：出席、1：欠席、2：遅刻）>0
第15週目：（0：出席、1：欠席、2：遅刻）>2
出席：9回
欠席：2回
遅刻：4回
出席率は80.0%です！出席率OKです！

課題完成時の画面（出席率が75%未満のとき）

出席状況を入力してください。
第1週目：（0：出席、1：欠席、2：遅刻）>0
第2週目：（0：出席、1：欠席、2：遅刻）>1
⋮
（中略）
⋮
第14週目：（0：出席、1：欠席、2：遅刻）>1
第15週目：（0：出席、1：欠席、2：遅刻）>0
出席：11回
欠席：4回
遅刻：0回
出席率は73.3%です！出席率NGです！

● JKad20X 「バブルソート！」

入力された枚数のカード（配列、値は乱数で 0～99）を生成し、バブルソート（もしまだ習っていなかったら検索すること）を使って昇順に並べる処理を作成せよ。なお、以下のメソッドを作成し、データの比較（データの入れ替えでないので注意）を行うたびに比較回数とカードの並びを表示すること。

書式	処理
public static void showCards(int count, int[] cards)	比較回数（count）と配列 cards の先頭から最後までを表示する。

課題完成時の画面

バブルソートを使ってカードを昇順に並べます！
カードの枚数を入力してください>10

0回目

7 3 85 13 40 34 26 14 16 79

1回目

3 7 85 13 40 34 26 14 16 79

2回目

3 7 85 13 40 34 26 14 16 79

3回目

3 7 13 85 40 34 26 14 16 79

4回目

3 7 13 40 85 34 26 14 16 79

5回目

3 7 13 40 34 85 26 14 16 79

6回目

3 7 13 40 34 26 85 14 16 79

7回目

3 7 13 40 34 26 14 85 16 79

8回目

3 7 13 40 34 26 14 16 85 79

9回目

3 7 13 40 34 26 14 16 79 85

⋮

(中略)

⋮

36回目

3 7 13 14 16 26 34 40 79 85

37回目

3 7 13 14 16 26 34 40 79 85

38回目

3 7 13 14 16 26 34 40 79 85

39回目

3 7 13 14 16 26 34 40 79 85

40回目

3 7 13 14 16 26 34 40 79 85

41回目

3 7 13 14 16 26 34 40 79 85

42回目

3 7 13 14 16 26 34 40 79 85

43回目

3 7 13 14 16 26 34 40 79 85

44回目

3 7 13 14 16 26 34 40 79 85

45回目

3 7 13 14 16 26 34 40 79 85

昇順（値が小さい順）に並べ替える。

	0	1	2	3	4	5	6	7	8	9
cards	7	3	85	13	40	34	26	14	16	79

↓

	0	1	2	3	4	5	6	7	8	9
cards	3	7	13	14	16	26	34	40	79	85

0 回目はバブルソートを行う前の表示（カードに値を設定した段階で表示する）。

1 回目～45 回目はデータの比較を行ったあとに表示する（入れ替えを行っても行わなかったも表示する）。

カード枚数による比較回数は以下の通り。

- ・ 5 枚 …10 回
- ・ 10 枚 …45 回
- ・ 20 枚 …190 回
- ・ 50 枚 …1225 回
- ・ 100 枚 …4950 回

※比較回数は枚数を n とすると以下ようになる

回数 = $n \times (n - 1) \div 2$