

問題用紙

2025	科目名	知能情報： プログラミング系科目群	1 / 3	通し番号	
------	-----	----------------------	-------	------	--

先生は A 君と B 君にそれぞれにソートアルゴリズムを実装するよう課題を与えた。先生はスーパークラスとしてクラス `Sorter` を A 君と B 君に渡した。A 君はクラス `ASorter` を、B 君はクラス `BSorter` をクラス `Sorter` のサブクラスとして作成した。先生はそれらを統合して、クラス `SAC` を実行した。

なお、`BSorter` 内で利用しているクラス `Stack` は `int` 型変数に対してデータ構造のスタック機能を提供しており、メソッド `isEmpty` はスタックが空の場合 `true` を返す(クラス `StackExample` の実行結果を参照)。

各クラスのソースコードの一部および実行結果を次ページ以降に示す。なお、黒塗りの箇所には、ソースコードあるいは標準出力の一部が入る。それらを参照して、以下の問題に解答せよ。

問題 1 空欄(1)には、自分のクラス内からのみアクセス可能を示すアクセス修飾子が入る。空欄(1)を記述しなさい。

問題 2 空欄(2)には、サブクラスにてメソッドのオーバーロードを要求する修飾子が入る。空欄(2)を記述しなさい。

問題 3 空欄(3)には、クラス `ASorter` のコンストラクタが入る。空欄(3)を記述しなさい。

問題 4 空欄(4)には、メソッド `getPivot` において、`x1,x2,x3` の中央値を `pv` に代入する処理が入る。空欄(4)を記述しなさい。

問題 5 空欄(5)にはクラス `ASort` のメソッド `performSort` 内から呼び出されるメソッド `printData` の出力が入る。空欄(5)を記述しなさい。

問題 6 空欄(6)にはクラス `BSort` のメソッド `performSort` 内から呼び出されるメソッド `printData` の出力が入る。空欄(6)を記述しなさい。

問題用紙

2025	科目名	知能情報： プログラミング系科目群	2 / 3	通し番号
------	-----	----------------------	-------	------

```

public class Sorter {
    空欄(1) int data[];
    空欄(1) String sortName;

    public Sorter(int data[], String sortName) {
        this.data = new int[data.length];
        for (int i=0; i < data.length; i++) {
            this.data[i]=data[i];
        }
        this.sortName=sortName;
    }

    void printSortName() {
        System.out.println(sortName);
    }

    int getDataLen() {
        return data.length;
    }

    int getData(int i) {
        return data[i];
    }

    void swapData(int l, int r) {
        int temp=data[l];
        data[l]=data[r];
        data[r]=temp;
    }

    void printData() {
        for (int i=0; i < data.length-1; i++) {
            System.out.print(data[i]);
            System.out.print(",");
        }
        System.out.println(data[data.length-1]);
    }

    空欄(2) void performSort();
}

```

```

public class ASorter extends Sorter {

    空欄(3)

    void performSort(){
        System.out.println("ここでソートします");

        int loop=1;
        for (int i=getDataLen()-1; i > 0; i--) {
            for (int j=i-1; j >= 0; j--) {
                if (getData(i) > getData(j)) {
                    swapData(i, j);
                }
            }

            System.out.println("loop = "+(loop++));
            printData();
        }
    }
}

```

```

public class BSorter extends Sorter {

    //ここに適切にコンストラクタが定義されている

    int getPivot(int pl, int pr) {
        int x1=getData((pl+pr)/2);
        int x2=getData(pl);
        int x3=getData(pr);
        int pv=x1;

        空欄(4)

        return pv;
    }

    void performSort(){
        System.out.println("ここでソートします");
        int left, right;

        Stack stack = new Stack();
        int leftMin=0;
        int rightMax=getDataLen()-1;
        stack.push(leftMin);
        stack.push(rightMax);

        int loop=1;
        while (!stack.isEmpty()) {
            int pr=right=stack.pop();
            int pl=left=stack.pop();
            int pv=getPivot(pl, pr);

            do{
                while(getData(pl) < pv) {
                    pl++;
                }
                while(getData(pr) > pv) {
                    pr--;
                }
                if (pl <= pr) {
                    swapData(pl, pr);
                    pl++;
                    pr--;
                }
            } while (pl <= pr);

            if (left < pr) {
                stack.push(left);
                stack.push(pr);
            }
            if (pl < right) {
                stack.push(pl);
                stack.push(right);
            }

            System.out.println("loop = "+(loop++));
            printData();
        }
    }
}

```

問題用紙

2025	科目名	知能情報： プログラミング系科目群	3 / 3	通し番号
------	-----	----------------------	-------	------

```
public class StackExample {

    public static void main(String[] args) {
        Stack stack = new Stack();
        stack.push(10);
        stack.push(9);
        System.out.println(stack.isEmpty());
        System.out.println(stack.pop());
        System.out.println(stack.isEmpty());
        System.out.println(stack.pop());
        System.out.println(stack.isEmpty());
    }
}
```

StackExample の実行結果

```
false
9
false
10
true
```

```
public class SAC {
    final static int N=9;

    public static void main(String[] args) {
        int x[] = new int[N];
        //x[]に 0～50 の整数乱数が入る

        ASorter aSorter = new ASorter(x);
        BSorter bSorter = new BSorter(x);

        Sorter[] sorter = {aSorter,bSorter};
        for(int i=0; i < sorter.length; i++) {
            System.out.println("-----");
            sorter[i].printSortName();
            sorter[i].printData();
            sorter[i].performSort();
        }
    }
}
```

SAC の実行結果

```
-----
A_SelectionSort
3,24,6,9,49,37,25,46,12
ここでソートします
loop = 1
[redacted]
loop = 2
[redacted]
loop = 3
[redacted]
loop = 4
空欄 (5)
loop = 5
[redacted]
loop = 6
[redacted]
loop = 7
[redacted]
loop = 8
[redacted]
-----
B_QuickSort
3,24,6,9,49,37,25,46,12
ここでソートします
loop = 1
3,12,6,9,49,37,25,46,24
loop = 2
[redacted]
loop = 3
[redacted]
loop = 4
空欄 (6)
loop = 5
[redacted]
loop = 6
[redacted]
loop = 7
[redacted]
```