

## 問題 1 (20 点)

プログラム A01User を実行したところ、以下のような出力があった。A01.java の空欄(1)から(4)を埋めよ。

```
a->name = First
a->x = 50
a->y = 50
a->z = 50
a->average = 50.0
b->name = Second
b->x = 100
b->y = 90
b->z = 80
b->average = 90.0
```

ただし、(1)には引数が 1 個のコンストラクタ、(2)には引数が 4 個のコンストラクタを記述せよ。(3)には getName, getX, getY, getZ メソッドを記述せよ。(4)には getAverage メソッドを記述せよ。すべて public 指定とする。

## 問題 2 (30 点)

A02User は A02 クラスのインスタンスを 2 個生成して何らかの処理を実行している。A02User.java を読んで、A02.java の空欄(1)~(3)を埋めよ。ただし、空欄(1)には 2 個の引数をとるコンストラクタ、(2)には setXs メソッド、(3)には getAverage メソッドをすべて public 指定で記述せよ。getAverage メソッドはインスタンス変数 updated の値が true のとき、インスタンス変数 xs が参照する int 型配列の平均値を計算してインスタンス変数 average に格納し、インスタンス変数 updated の値を false に変え、インスタンス変数 average の値を返す。updated の値が false のときは、単にインスタンス変数 average の値を返す。

また、A02User の出力を解答欄の(4)に記せ。以下の公式が必要かも知れない。

$$\sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$$

### 問題3 (20点)

A03クラスの親クラスはA02である。追加された `private` メソッド `getSumOfSquares` はインスタンス変数 `xs` が参照する `int` 型配列の各要素の平方の和を計算し、`int` 型で返す。`getSD` メソッドは `public` であり、インスタンス変数 `xs` が参照する `int` 型配列に格納されているデータの標準偏差を計算し、`double` 型で返す。

解答欄(1)に A03.java の完全なプログラムを記述せよ。

$a_0, a_1, \dots, a_{n-1}$  の標準偏差は  $\left( \sum_{k=0}^{n-1} a_k^2 \right) / n - \left( \sum_{k=0}^{n-1} a_k / n \right)^2$  の平方根で与えられる。

A03User の実行結果は以下の通り。

```
First: 28.86607004772212
Second: 26.852374196707448
```

### 問題4 (20点)

プログラム A04User を実行すると以下のような出力が得られる。A04.java の空欄(1)には `doPrint` メソッドを記述せよ。A041.java の(2)、A042.java の(3)を完成させよ。

ただし、A04 の `doPrint` メソッドは `public`、A041, A042 で実装されるメソッドは `protected` を指定すること。

```
Graduation Ceremony
> NO.1000
2009/3/24 at B.T.
-----
Graduation Ceremony
> NO.1001
2009/3/24 at B.T.
-----

省略

Graduation Ceremony
> NO.1009
```

```

2009/3/24 at B.T.
-----
New Year Party Ticket
> NO.100
Fee 1000 yen
-----
New Year Party Ticket
> NO.101
Fee 1000 yen
-----

省略

New Year Party Ticket
> NO.114
Fee 1000 yen
-----

```

### 問題5 (10点)

Course.java、Student.java、RegistrationDriver.javaをよく読んで、プログラムRegistrationDriverを実行したときのコンソール画面出力を書け。

ただし、LinkedList<Student>、Iterator<Student>は以下のようなメソッドをもつクラスと考えてよい。StudentをCourseに置き換えたものも同様である。

表 5.1 LinkedList<Student>のメソッドの一部

boolean	add(Student st)	指定した要素を追加する(optional)
void	clear()	すべての要素を削除する(optional)
Iterator<Student>	iterator()	この Collection 内の要素を走査するイテレータを返す

表 5.2 Iterator<Student>のメソッドの一部

boolean	hasNext()	繰り返し処理でさらに要素がある場合に true を返す
Student	next()	繰り返し処理で次の要素を返す