平成２４年度情報処理概論期末試験

問１． Java 言語とアルゴリズムに関する次の文章の空欄 (a) 〜 (x) を埋めよ。

- 64bit の整数型の配列変数 lgL を宣言し，その要素 6 個分の領域を確保し，その参照情報を lgL に代入するためには，(a) と記述する。このとき，配列 lgL のインデックスとして，(b) から (c) までを使うことが可能である。また，配列変数 lgL の宣言，領域確保と，要素の値を順番に-2，2，1，7，1，4 で初期化することを同時に行うためには，(d) と記述する。
- メソッドを区別する (e) は，メソッドの (f) と (g) の型の並びからなる。一つのクラスの中で，名前がが同じで，異なる (g) の型の並びをもつメソッドを定義することを，(h) すると言う。
- ある変数を利用できる範囲を，その変数の (i) (カタカナ 4 文字) と呼ぶ。識別子 var が，フィールドとローカル変数の両方で宣言されていて，両方の (i) に含まれる場合，ローカル変数の方を参照したい場合は (j) と記述し，フィールドの方を参照したい場合は (k) と記述する。
- コンストラクタが，自分のクラスの (g) の型の並びが異なるコンストラクタを呼び出すときには，キーワード (l) を用いる。
- すでに宣言されているクラス Student のオブジェクトを，ディフォルトのコンストラクタで作成して，そのクラス型変数 std に格納するときには，(m) と記述する。また，std のフィールド (クラス Student で定義されたフィールド)name を参照するときには，(n) と記述する。
- クラス Parts を元に，拡張・変更した新しいクラス Capacitor を宣言することを，Parts を (o) して，Capacitor を宣言すると言う。このとき，Parts を (p) クラスまたは (q) クラス，Capacitor を (r) クラスまたは (s) クラスと呼ぶ。(p) クラスのコンストラクタを呼び出したいときには，キーワード (t) を用いる。
- (p) クラスにあるメソッドと同じ (e) を持ったメソッドを，(r) クラスで定義することをメソッドの (u) と呼ぶ。
- データを昇順あるいは降順に並び替えるソートの方法には，(v)，バブルソート，クイックソート，マージソート，ヒープソートなどがあるが，データ数 N に対して，(v) の計算量のオーダーは N^2 であり，ヒープソートの計算量のオーダは (w) がある。
- (u) のプログラム例を示すと (x) となる。

問２． カメラ，レンズ，ディジタルカメラセットなどの仕様を表示する次のプログラムを作成した。このプログラムを表示されるものを記せ。

```
class Kimatsu2 {
  public static void main(String[] args) {
    Camera        petri = new Camera(300.0);
    FilmCamera    canon = new FilmCamera(620.0, "Lica");
    DigitalCamera nikon = new DigitalCamera(100.0, "35mm");
    DigitalCamera olympus = new DigitalCamera(430.0, "4/3");
    Lense         afs50 = new Lense(50, 1.4);
    DCameraSet    dcs   = new DCameraSet(nikon, afs50);
    dcs.dCam.weight = 520.0;
    afs50.printInf();   petri.printInf();
    canon.printInf();   nikon.printInf();
    olympus.printInf(); dcs.printInf();
  }
}
```

(次ページに続く)

```
class DCameraSet {
  DigitalCamera dCam;
  Lense         len;
  DCameraSet(DigitalCamera dCam, Lense len) {
    this.dCam = dCam;
    this.len  = len;
  }
  void printInf() {
    dCam.printInf();
    len.printInf();
  }
}
class Camera {
  double weight;
  Camera(double weight) { this.weight = weight; }
  void printInf() {
    System.out.println("カメラ：質量：" + weight + "g");
  }
}
class FilmCamera extends Camera {
  String filmSize; // "Lica", "half", "APS", "6x7"
  FilmCamera(double weight, String filmSize) {
    super(weight);
    this.filmSize = filmSize;
  }
  void printInf() {
    System.out.println("フィルムカメラ：フィルムサイズ = " + filmSize + ", 質量 = " + weight + "g");
  }
}
class DigitalCamera extends Camera {
  String sensorType; // "35mm", "half", "APS", "6x7"
  DigitalCamera(double weight, String sensorType) {
    super(weight);
    this.sensorType = sensorType;
  }
  void printInf() {
    System.out.println("ディジタルカメラ:センサータイプ = " + sensorType + ", 質量 = " + weight + "g");
  }
}
class Lense {
  double focalLength;
  double iris;
  Lense(double focalLength, double iris) {
    this.focalLength = focalLength;
    this.iris        = iris;
  }
  void printInf() {
    System.out.println("レンズ：焦点距離 = " + focalLength + ", 絞り = " + iris + "g");
  }
}
```

問３． 次のプログラムは，学生の成績係数と最高点または平均点で降順にソートして表示するものである。その仕様を次に示す。

(1) 科目の通常の成績は，0 から 100 点であり，その他に-1 とするとその科目の申告を取り消したことを意味する。

(2) 成績係数は，100 から 80 までを 3 点，79 から 70 までを 2 点，69 から 60 までを 1 点，60 から 0 までを 0 点とし，その合計を申告して申告を取り消していない科目数で割ったものである。

(3) 平均点は，60 点以上の科目の成績の合計を，60 点以上の科目数で割ったものである。

(4) クラス StdEvaP のフィールドは，名前を格納する String のクラス型の変数 name と，申告した科目数を格納する int 型の変数 nApply，各科目の成績を格納する int 型の配列変数 pointL，科目の成績を順番に格納するための int 型の変数 nSet，成績係数，平均点，最高点を格納する，double 型，double 型，int 型の変数 evaP，aveP，maxP である。name と pointL 以外は，0 または 0.0 に初期化する。

(5) コンストラクタ StdEvaP(String name, int nApply) は，引数の値を対応するフィールドに格納するとともに，int 型 nApply 個分の領域を確保し，その参照情報を pointL に格納する。

(6) メソッド void setP(int p) は，引数として与えられた成績 p を pointL[nSet] に格納し，nSet に 1 を加算する。こうすることによって，インデックスを指定することなく，順番に成績を格納することができる。

(7) メソッド calP() は，格納された成績から，(1), (2), (3) の説明に基づいて，成績係数，平均点，最高点を計算し，自分のフィールドに格納する。

(8) メソッド static boolean larger(int type, StdEvaP a, StdEvaP b) は，引数で指定される学生 a が b より，type が 0 のときは，成績係数が高いか成績係数が同じ場合は最高点が高いときに，type が 1 のときは，平均点が高いときに true を返し，そうでないときに false を返す。

(9) メソッド print() は，その学生の名前，成績係数，平均点，最高点を表示する。

(10) クラス Kimatsu3 は，5 人分のクラス StdEvaP の配列と，それぞれの要素となるオブジェクトを作成し，成績をセットし，type = 0 の大小関係で降順にソートし，表示するプログラムである。5 人の名前と成績と，計算した成績係数と平均値を以下の表に示す。ここで，N.A. は申告していないことを表す。

氏名	成績 1	成績 2	成績 3	成績 4	成績 5	計算した成績係数	計算した平均値
沖田十三	77	100	62	0	N.A.	1.5	79.66666666666667
古代　進	90	55	98	52	N.A.	1.5	94.0
島　大介	90	20	65	77	82	1.8	78.5
森　　雪	85	-1	82	83	N.A.	3.0	83.33333333333333
真田志郎	95	90	88	100	79	2.8	90.4

このとき，無駄なく上の仕様を満たすために，プログラム中の下線部 (a)〜(h) に書き入れるべきものを答えよ。また，出力として表示されるものを記せ。

```
class Kimatsu3 {
  public static void main(String[] args) {
    int N = 5;
    StdEvaP sL[] = new StdEvaP[N];
    sL[0] = new StdEvaP("沖田十三", 4); sL[1] = new StdEvaP("古代　進", 4);
    sL[2] = new StdEvaP("島　大介", 5); sL[3] = new StdEvaP("森　　雪", 4);
    sL[4] = new StdEvaP("真田志郎", 5);
    sL[0].setP(77); sL[0].setP(100); sL[0].setP(62); sL[0].setP(0);
    sL[1].setP(90); sL[1].setP(55); sL[1].setP(98); sL[1].setP(52);
    sL[2].setP(90); sL[2].setP(20); sL[2].setP(65); sL[2].setP(77); sL[2].setP(82);
    sL[3].setP(85); sL[3].setP(-1); sL[3].setP(82); sL[3].setP(83);
    sL[4].setP(95); sL[4].setP(90); sL[4].setP(88); sL[4].setP(100); sL[4].setP(79);

    for (int k = 0 ; k < N ; ++k) _____(a)_____
```

(次ページに続く)

```
    for (int k = 0 ; k < N - 1 ; ++k) {
      int maxInd = k;
      for (int l = k + 1 ; l < N ; ++l) {
        if (_____(b)_____) maxInd = l;
      }
      StdEvaP tmp   = _____(c)_____;
      sL[k]         = _____(d)_____;
      sL[maxInd]    = _____(e)_____;
    }
    for (int i = 0 ; i < N ; ++i) sL[i].print();
  }
}
class StdEvaP {
  String name;
  int nApply = 0, pointL[], nSet = 0, maxP = 0;;
  double evaP = 0.0, aveP = 0.0;
  StdEvaP(String name, int nApply) {
    this.name   = new String(name);
    this.nApply = nApply;
    pointL      = new int[nApply];
  }
  void setP(int p) { pointL[nSet++] = p; }
  void calP() {
    int sumOver60 = 0, nOver60 = 0;   // 60 点以上の科目の合計点と科目数
    int sumEva    = 0, nEva    = 0;   // 成績係数の計算に必要な成績の合計と科目数
    for(int l = 0 ; l < nApply ; ++l) {
      if (pointL[l] >= 80) sumEva += 3;
      else if (pointL[l] >= 70) sumEva += 2;
      else if (pointL[l] >= 60) sumEva += 1;
      if (pointL[l] >= 0) ++nEva;             // 申告してそれを取り消しない科目数の計算
      if (pointL[l] >= 60) {
        sumOver60 += pointL[l]; ++nOver60;    // 60 点以上の科目の合計点の計算
      }
      _____(f)_____; // 最高点の計算
    }
    evaP = _____(g)_____ sumEva / nEva;
    aveP = _____(g)_____ sumOver60 / nOver60;
  }
  static boolean larger(int type, StdEvaP a, StdEvaP b) {
    boolean result = false;
    switch (type) {
    case 0: result = a.evaP > b.evaP || _____(h)_____;  break;
    case 1: result = a.aveP > b.aveP; break;
    }
    return result;
  }
  void print() {
    System.out.println(name + ": evaP = " + evaP + " aveP = " + aveP + " maxP = " + maxP);
  }
}
```

解答用紙

学科・類：　　学籍番号：　　名前：

問1．$2 \times 23 + 4 = 50$ 点

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

(k)

(l)

(m)

(n)

(o)

(p)

(q)

(r)

(s)

(t)

(u)

(v)

(w)

(x)

問２．(16 点)

問３．$3 \times 8 + 10 = 34$ 点

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

出力

解答例

学科・類：　　学籍番号：　　名前：

問1．$2 \times 23 + 4 = 50$ 点

(a) `long lgL[] = new long[6];`

(b) `lgL[0]`

(c) `lgL[5]`

(d) `int vals[] = {-2, 2, 1, 7, 1, 4};`

(e) シグネチャ

(f) 名前

(g) 引数

(h) オーバーロード

(i) スコープ

(j) `var`

(k) `this.var`

(l) `this`

(m) `std = new Student();`

(n) `std.name`

(o) 継承 (inheritance)

(p) 親

(q) スーパー

(r) 子

(s) サブ

(t) `super`

(u) オーバーライド

(v) 選択ソート

(w) $N \log_2 N$

(x)

```
class Oya {
  void func() {
    System.out.println("親");
  }
}
class Ko {
  void func() {
    System.out.println("子");
  }
}
```

問２．(16 点)

```
レンズ：焦点距離 = 50.0, 絞り = 1.4g
カメラ：質量：300.0g
フィルムカメラ：フィルムサイズ = Lica, 質量 = 620.0g
ディジタルカメラ：センサータイプ = 35mm, 質量 = 520.0g
ディジタルカメラ：センサータイプ = 4/3, 質量 = 430.0g
ディジタルカメラ：センサータイプ = 35mm, 質量 = 520.0g
レンズ：焦点距離 = 50.0, 絞り = 1.4g
```

問３．$3 \times 8 + 10 = 34$ 点

(a) `sL[k].calP()`

(b) `StdEvaP.larger(0, sL[l], sL[maxInd])`

(c) `sL[k]`

(d) `sL[maxInd]`

(e) `tmp`

(f) `if (maxP < pointL[l]) maxP = pointL[l]`

(g) `(double)`

(h) `a.evaP == b.evaP && a.maxP > b.maxP`

出力

```
森　　雪: evaP = 3.0 aveP = 83.33333333333333 maxP = 85
真田志郎: evaP = 2.8 aveP = 90.4 maxP = 100
島　大介: evaP = 1.8 aveP = 78.5 maxP = 90
沖田十三: evaP = 1.5 aveP = 79.66666666666667 maxP = 100
古代　進: evaP = 1.5 aveP = 94.0 maxP = 98
```