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) 〜 (t) を埋めよ。

- 32bit の整数型の配列変数 vals を宣言し，その要素 8 個分の領域を確保し，その参照情報を vals に代入するためには，(a) と記述する。このとき，vals の要素として，(b) から (c) までを使うことが可能である。また，配列変数 vals の宣言，領域確保と，要素の値を順番に 3, 4, 5, 7, 2, 3, 1, 3 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，メソッドの名前 (識別子) とその (f) の (g) 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 x 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。
- 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，(l) 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)a を参照するときには，(m) と記述する。
- クラス Student を元に，拡張・変更した新しいクラス UnivStudent を宣言することを，Student を (n) して，UnivStudent を宣言すると言う。このとき，親クラスは (o)，子クラスは (p) である。親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 (q) を用いる。
- 親クラスにあるメソッドと同じ (e) を持ったメソッドを，子クラスで定義することをメソッドの (r) 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，選択ソート，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，選択ソートの計算量のオーダーは (s) であり，ヒープソートの計算量のオーダは (t) がある。

問２． 電話機の仕様を表示する次のプログラムを作成した。このとき，次の問に答えよ。

(ア) このプログラムの出力を記せ。

(イ) このプログラムの終わりの方にある// pb.printOsType(); の//を削除すると，どのようなエラーが発生するか答えよ。

(ウ) (イ) のようになる理由を答えよ。

(エ) この文を修正して，pb が示すオブジェクトに対して，printOsType() を実行させるためにはどうすれば良いか答えよ。

```
class Phone {
  double width;
  double length;
  double height;
  Phone(double width, double length, double height) {
    this.width  = width;
    this.length = length;
    this.height = height;
  }
  void printInf() {
    System.out.println("width = " + width + " lenght = " + length + " height = " + height);
  }
}
```

(次ページに続く)

```
class CellPhone extends Phone {
  int osType; // 0 : jTron, 1: jOS, 2: Bndroid, 3: Vindows
  double weight;
  CellPhone(double width, double length, double height, int osType, double weight) {
    super(width, length, height);
    this.osType = osType; this.weight = weight;
  }
  void printInf() {
    System.out.println("width = " + width + " lenght = " + length
        + " height = " + height + " weight = " + weight);
    printOsType();
  }
  void printOsType() {
    switch (osType) {
    case 0: System.out.println("OS = jTron"); break;
    case 1: System.out.println("I do not like jOS"); break;
    case 2: System.out.println("クークル OS：Bndroid"); break;
    case 3: System.out.println("無茶だ。こんな OS(Vindows) でこれだけの機体を。。。"); break;
    }
  }
}

class Kimatsu2 {
  public static void main(String[] args) {
    Phone pa = new Phone(50.0, 200.0, 300.0);
    Phone pb = new Phone(100.0,  100.0, 50.0);
    Phone pc = new CellPhone(62.1, 115.5, 12.3, 1, 135.0);
    CellPhone pd = new CellPhone(80.0, 140.0, 11.0, 2, 140.0);
    CellPhone pe = new CellPhone(70.0, 120.0, 14.5, 3, 135.0);
    pb = pd;
    pa.printInf(); pb.printInf();
    pc.printInf(); pd.printInf();
    pe.printInf();
    pd.printOsType();
    // pb.printOsType();
  }
}
```

問３． 次のプログラムは，チームに対して試合の結果をもとに，勝点や勝数またはそれらを組み合わせたもので降順にソートして表示するものである。

仕様

(1) クラス Team 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，それぞれ，勝数，引き分け数，勝点を格納する int 型の変数 nWin，nDrawn，point である。

(2) コンストラクタ Team(String name) は，引数として与えるチーム名を対応するフィールドに格納する。

(3) メソッド static void game(int win，Team a，Team b) は，試合結果に応じて，勝数，引き分け数を変更するプログラムである。win が，1，-1 の場合は，それぞれ，a が b に，b が a に勝ったものとして，0 の場合は引き分けとして，フィールドを変更する。

(4) メソッド void calPoint() は，フィールドの勝数と引き分け数の情報から，勝点を計算して，フィールドの point に格納する。勝点の計算は，1 つの勝利に対して 3 点，1 つの引き分けに対して 1 点を加点する。

(5) メソッド static boolean comp(int type, Team a, Team b) は，2 つの Team のオブジェクト a と b を比べて，a が優位である時に true を返し，そうでないときに false を返す。優位性は type によって変わり，type が 0 の場合は勝点が多いときを，1 の場合は勝数が多いときを，2 の場合はまず勝点が多いときを，勝点が同じときは勝数が多いときを優位とする。

(6) メソッド void print() は，自身の名前，勝点，勝数を表示する。

(7) クラス Kimatsu3 は，クラス Team の配列とそれぞれの要素となるオブジェクトを作成し，勝数，引き分け数を試合結果に応じて設定し，その後で勝点を計算して設定し，勝点を優先して，勝点と勝数で降順にソートして表示するメインメソッドのためのクラスである。なお，10 試合の結果は以下の通りである。(○，×，△がついている場合は，横のチームが縦のチームに勝った，負けた，引き分けだったことを意味する。

	ビーナス	アース	マーズ	ジュピタ	サターン
ビーナス		○	×	×	×
アース	×		△	△	△
マーズ	○	△		○	○
ジュピタ	○	△	×		×
サターン	○	△	×	○	

このとき，次の問に答えよ。

(ア) 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)〜(j) に書き入れるべきものを答えよ。

(イ) このプログラムで表示されるものを答えよ。

```
class Team {
  String name;
  int nWin = 0, nDrawn = 0; // 勝数，引き分けの数
  int point = 0;            // 勝点
  Team (String name) {
    this.name = name;
  }

  static void game(int win, Team a, Team b) {
    switch (win) {
    case -1:b.nWin += 1; break;
    case  0:
      ____________(a)____________;
      ____________(b)____________;
      break;
    case  1: a.nWin += 1; break;
    }
  }

  void calPoint() {
    point = nWin * 3 + nDrawn;
  }
```

```
  static boolean comp(int type, Team a, Team b) {
    boolean result = false;
    switch (type) {
    case 0: result = (a.point > b.point); break;
    case 1: result = (a.nWin > b.nWin); break;
    case 2:
      if (a.point > b.point) result = true;
      else if (a.point == b.point) {
        ____________(c)____________;
      }
      break;
    }
    return result;
  }
  void print() {
    System.out.println(name + ": point = " + point + " nWin = " + nWin);
  }
}

class Kimatsu3 {
  public static void main(String[] args) {
    int N = 5;
    Team tm[] = new Team[N];
    tm[0] = new Team("ビーナス"); tm[1] = new Team("アース  ");
    tm[2] = new Team("マーズ  "); tm[3] = new Team("ジュピタ");
    tm[4] = new Team("サターン");
    Team.game( 1, tm[0], tm[1]); Team.game(-1, tm[0], tm[2]);
    Team.game(-1, tm[0], tm[3]); Team.game(-1, tm[0], tm[4]);
    Team.game( 0, tm[1], tm[2]); Team.game( 0, tm[1], tm[3]);
    Team.game( 0, tm[1], tm[4]); Team.game( 1, tm[2], tm[3]);
    ____________(d)____________; ____________(e)____________;

    ________________________(f)________________________;

    for (int k = 0 ; k < N - 1 ; ++k) {
      int maxInd = k;
      for (int l = k + 1 ; ____________(g)____________ ; ++l) {
        if (____________(h)____________) maxInd = l;
      }
      Team tmp   = tm[maxInd];
      ____________(i)____________;
      ____________(j)____________;
    }
    for (int i = 0 ; i < N ; ++i) tm[i].print();
  }
}
```

解答用紙 (配点は変更することがあります)

学科・類：　　学籍番号：　　名前：

問1．($2 \times 20 = 40$ 点)

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

(k)

(l)

(m)

(n)

(o)

(p)

(q)

(r)

(s)

(t)

問2．($10 + 5 + 5 + 5 = 25$)

(ア)

(イ)

(ウ)

(エ)

問3．$(3 \times 10 + 5 = 35)$

(ア)

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

(イ)

解答用紙

学科・類：　　学籍番号：　　名前：

問1.

(a) `int vals[] = new int[8];`

(b) `vals[0]`

(c) `vals[7]`

(d) `int vals[] = {3, 4, 5, 7, 2, 3, 1, 3};`

(e) シグネチャ

(f) 引数

(g) 型

(h) オーバーロード

(i) スコープ

(j) `x`

(k) `this.x`

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

(l) `std.a;`

(n) 継承 (inheritance)

(o) `Student`

(p) `UnivStudent`

(q) `super`

(r) オーバーライド

(s) N^2

(t) $N \log_2 N$

問2.

(ア)

```
width = 50.0 lenght = 200.0 height = 300.0
width = 80.0 lenght = 140.0 height = 11.0 weight = 140.0
クークル OS：Bndroid
width = 62.1 lenght = 115.5 height = 12.3 weight = 135.0
I do not like jOS
width = 80.0 lenght = 140.0 height = 11.0 weight = 140.0
クークル OS：Bndroid
width = 70.0 lenght = 120.0 height = 14.5 weight = 135.0
無茶だ。こんな OS(Vindows) でこれだけの機体を。。。
クークル OS：Bndroid
```

(イ)

コンパイル時に，メソッド printOsType() が見つからないというエラーが発生する。

(ウ)

この文の実行時には，変数 p2 にはクラス CellPhone のオブジェクトの参照情報が
格納されているが，コンパイル時には型の情報しか見ない。
pb の型が Phone であるため，コンパイル時はクラス Phone の中からだけからメソッドを
探すが，クラス Phone にはメソッド printOsType() が定義されていない，
従って，コンパイル時にメソッドがないとのエラーが生じる。

(エ)

この文を，
((CellPhone) pb).printOsType();
とする

問3.

(ア)

(a) a.nDrawn += 1

(b) b.nDrawn += 1

(c) result = (a.nWin > b.nWin)

(d) Team.game(1, tm[2], tm[4])

(e) Team.game(-1, tm[3], tm[4]);

(f) for (int k = 0 ; k < N ; ++k) tm[k].calPoint();

(g) l < N

(h) Team.comp(2, tm[l], tm[maxInd])

(i) tm[maxInd] = tm[k];

(j) tm[k] = tmp;

(イ)

```
マーズ  : point = 10 nWin = 3
サターン: point = 7 nWin = 2
ジュピタ: point = 4 nWin = 1
ビーナス: point = 3 nWin = 1
アース  : point = 3 nWin = 0
```