アルゴリズムとプログラミング期末試験

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

- クラス testClass のディフォルトのコンストラクタを使って，オブジェクトを作成するときは，(a) と記述する。
- メソッド `int prob2(double arg1, double arg2) { return (int) arg1; }` において，そのクラスのメソッドを区別する (b) は，(c) と引数の (d) の並びからなる。上記のメソッドの場合は，それぞれ，(e) と (f) (並びの各要素をカンマで区切り，余分なものを書かないこと) である。
  このクラスに，`int prob2(int arg1, int arg2) { return arg1; }` のようなメソッドを定義することを，メソッドの (g) と呼ぶ。
- クラス ClassA を元に，拡張・変更した新しいクラス ClassB を宣言することを，ClassA を (h) して，ClassB を宣言すると言う。ClassB の宣言文の最初の行の { の前までの記述は，(i) となる。
- 親クラスと同じ (b) 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 (j) と呼ぶ。
- あるクラスのオブジェクトは，そのクラス以外の，その (k) クラスのオブジェクト参照型の変数からも参照することができる。このような仕組みを，オブジェクト指向における (l) と呼ばれる。
- 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，(m)，クイックソート，(n) などがある。選択ソートの場合，データ量が 10 倍になれば，計算量はおよそ (o) 倍になる。ソートにかかる時間は，一般的には，選択ソートを使う方がクイックソートを使うよりも (p) い。

問２． 経営不振に陥った東京興行大学興学部では，成績が最も不良な教員をリストラすることにした。次ページのプログラムは，教員の評価値を入力して成績の昇順に並び替えるプログラムである。

(1) クラス Teacher は，１人分の教員のデータを格納するためのクラスである。

- フィールド pt は，5 つの評価値を格納するための整数型配列で，フィールド numUnderThree には 3 点未満の評価の数，フィールド minPoint には 5 つの評価値の最小値，フィールド sumPoint には 5 つの評価値の和，フィールド name に教員名を格納する。
- Teacher(String inName, int p0, int p1, int p2, int p3, int p4) はコンストラクタで，名前 inName と同じ文字列を作成し name に代入し，整数型配列のための長さ 5 の領域を確保し pt に代入し，5 つの評価値 p0，p1，p2，p3，p4 を，pt の各要素に格納すると共に，それらの 3 点未満の数，最小値，和を計算し，それぞれ numUnderThree，minPoint，sumPoint に格納する。
- gt(Teacher t1, Teacher t2) は，t1 の成績が t2 の成績を上回る場合に true を，そうでない場合は false を返り値とする**クラス**メソッドである。教員の成績がより高いとは，3 点未満の評価値の数が少ない場合，それが同じ場合は合計点が高い場合，その両者とも同じ場合は最低点が高い場合である (その 3 つも同じ場合は同じ成績とする)。

(2) クラス Faculty は，学部の教員のデータを格納し，ソートするためのクラスである。

- フィールド teachers は，Teacher のオブジェクト参照型配列 (各要素が Teacher のオブジェクトを参照している配列) である。
- Faculty(int nTeachers) はコンストラクタである。クラス Teacher のオブジェクト参照型変数のための，長さ nTeachers の領域を確保し teachers に代入する。(nTeachers 人分の Teacher のオブジェクトを後で代入するものとする。)
- setData(int tn, String inName, int p0, int p1, int p2, int p3, int p4) は，クラス Teacher のオブジェクトを，名前 inName，評価値 p0，p1，p2，p3，p4 でコンストラクトし，teachers[tn] に代入する。
- sortSelect() は，teachers の内容を選択ソートによって，成績の昇順 (配列で成績が低いものが先にくる順番) に並び替えるメソッドである (成績が同じ場合の順番は問わない)。

(3) Kimatsu2 は，メインメソッドのためのクラスである。

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)〜(i) に書き入れるべきものを答えよ。

(イ) このプログラムを実行したとき (コマンド java Kimatsu2 によって動かしたとき)，出力されるものを答えよ。
(確認：System.out.println() を呼んでいる箇所はプログラム中に 2 カ所だけである。)

```
class Teacher {
  int pt[], numUnderThree = 0, sumPoint, minPoint;
  String name;
  Teacher(String inName, int p0, int p1, int p2, int p3, int p4){
    name = new String(inName);
    pt   = new int[5];
    pt[0] = p0; pt[1] = p1; pt[2] = p2; pt[3] = p3; pt[4] = p4;
    minPoint = 10;
    for (int i = 0 ; i < 5 ; ++i) {
      sumPoint += pt[i];
      if (minPoint > pt[i]) minPoint = pt[i];
      if (pt[i] < 3) ++numUnderThree;
    }
  }
  static boolean gt(Teacher t1, Teacher t2) {
    if       (____________(a)____________) return true;
    else if  (____________(b)____________) return false;
    else if  (____________(c)____________) return true;
    else if  (____________(d)____________) return false;
    else if  (____________(e)____________) return true;
    return false;
  }
}

class Faculty {
  Teacher[] teachers;
  Faculty(int nTeachers) {
    teachers = new Teacher[nTeachers];
  }
  void setData(int tn, String inName, int p0,int p1, int p2, int p3, int p4){
    teachers[tn] = ________________(f)________________;
  }
  void sortSelect() {
    for (int i = 0 ; i < teachers.length - 1 ; ++i) {
      int minj = i;
      for (int j = i + 1 ; j < teachers.length ; ++j) {
        if (____________(g)____________) minj = j;
      }
      ________________(h)________________;
      ________________(i)________________;
      teachers[minj] = tmp;
      System.out.println(teachers[i].name + "のデータと" + teachers[minj].name + "のデータを交換");
    }
  }
}

class Kimatsu2 {
  public static void main(String[] args) {
    Faculty eng = new Faculty(5);
    eng.setData(0, "綿鍋", 5, 5, 5, 5, 5);   eng.setData(1, "平木", 2, 2, 3, 4, 4);
    eng.setData(2, "山師多", 1, 2, 2, 3, 4); eng.setData(3, "宝",   2, 2, 4, 4, 4);
    eng.setData(4, "和泉", 2, 2, 2, 3, 3);
    eng.sortSelect();
    System.out.println("リストラ対象者は「" + eng.teachers[0].name +"」に決定！");
  }
}
```

問 3. 右下に示すようなクラス図を，下記の Java プログラム Kimatsu03.java に実装することを考える．下記の設問に答えよ．

（1） プログラム中のコメント文を参照して，(ア) ～ (ウ) の下線部に入る適当な語句を記しなさい．
（2） クラスＺを（エ）に実装しなさい．ただし，フィールド変数 z(int 型)は，z = x + y で初期化され，アクセス修飾子は無指定とする．また，持っている全てのフィールドを表示するための，返り値を持たないメソッド printFieldValue()を実装せよ．
（3） コメント文を参照して適当な文を（オ）～（ク）に記せ．
（4） （ケ）はコンパイルエラーとなる．理由を述べ，正しく動作するように下線部を書き換えよ．
（5） プログラム(Kimatsu03.java)の実行結果を記せ．ただし（ケ）は正しく書き換えられているものとする．

Kimatsu03.java

```
class X {

  static int x = 0;

  X(){
    x++;
  }

  // printFieldValue()は、所属するクラスの全てのフィールド変数
  //の内容を表示するメソッド（表示フォーマットは自由）
  void printFieldValue(){
    System.out.println( ＿＿＿＿（ア）＿＿＿＿ );
  }

}
// クラス図を参照してクラス Y を実装せよ
class Y ＿＿＿＿（イ）＿＿＿＿ {

  int y = x * x - 1;

  // printFieldValue()は、所属するクラスの全てのフィールド変数
  //の内容を表示するメソッド（表示フォーマットは自由）
  void printFieldValue(){
    System.out.println( ＿＿＿＿（ウ）＿＿＿＿ );
  }

}
// クラス図を参照してクラス Z を実装せよ。ただし、
// クラス Z のフィールド変数 z(int 型)は、z = x + y で初期化され、
// アクセス修飾子は無指定。また、所属するクラスの全てのフィールド変数
// の内容を表示するメソッド printFieldValue()を実装せよ。

 （エ）


class Kimatsu03 {
 public static void main(String[] args){

// クラス Z のインスタンスを生成し、
// クラス X のオブジェクト参照変数 o1 に代入せよ
＿＿＿＿（オ）＿＿＿＿;

// クラス Y のインスタンスを生成し、
// クラス X のオブジェクト参照変数 o2 に代入せよ
＿＿＿＿（カ）＿＿＿＿;

// クラス X のインスタンスを生成し、
// クラス X のオブジェクト参照変数 o3 に代入せよ
＿＿＿＿（キ）＿＿＿＿;

// o2 に対して printFieldValue()メソッドを呼べ
＿＿＿＿（ク）＿＿＿＿;

// o1 のフィールドを出力しようとして下記の文を書いたがコンパイルエラー
//となった。その理由を述べ、正しく動作するように下線部を書き換えよ。
(ケ)System.out.println( "o1.x=" + o1.x + " o1.y=" + o1.y );

 }
}
```

クラス図