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) 〜 (r) を埋めよ。

- 整数型の配列変数 arrayInt を宣言し，その要素 6 個分の領域を確保し，arrayInt[0] から arrayInt[5] を，それぞれ，3，2，4，1，3，5 で初期化する処理を 1 文で記述するときは，(a) と記述する。
- 一つのクラス内のメソッドを区別は (b) によって行われる。これは，メソッドの (c) とその引数の (d) の並びからなる。同じ (c) の，異なる引数の (d) の並びをもつメソッドを定義することを，(e) すると言う。
- フィールドとローカル変数に同じ識別子が使われている場合，その識別子だけで変数を表すと (f) ((f) の答えは，フィールドまたはローカル変数) と見なされる。そうでない方を示したい場合は，キーワード (g) を用いる。
- 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 の宣言の最初の'{' の前までを記すと，(i) となる。また，このときのスーパークラスはクラス (j) のことである。
- スーパークラスにあるメソッドと同じ (b) を持ったメソッドを，サブクラスで定義することをメソッドの (k) と呼ぶ。このときに，(l) (メソッドの定義の 1 行目で記述するもの) が異なると，コンパイルエラーが生じる。
- あるサブクラスのオブジェクトの参照情報は，そのスーパークラスのオブジェクト参照型の変数に代入することができる。このような仕組みを，オブジェクト指向における (m) と呼ばれる。サブクラスのオブジェクト参照型の変数に，スーパクラスのオブジェクト参照型の変数に格納されているサブクラスのオブジェクトの参照情報を代入するためには (n) を使う。
- 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，(o)，バブルソート，クイックソート，(p)，ヒープソートなどがある。データ数 N に対して，計算量が N^2 のオーダになるソートとして (q) が，計算量が $N \log_2 N$ のオーダになるソートとして (r) がある。

問２． 次のプログラムを実行した (java Kimatsu2) 時に，表示されるものを書け。また，その理由を記せ。

```
class Kimatsu2a {
  int fieldA;
  int fieldB;
  Kimatsu2a(int fA, int fB) {
    fieldA = fA; fieldB = fB;
  }
}
class Kimatsu2 {
  public static void main(String[] args) {
    int a   = 3;
    int b[] = {1, 2, 3, 4};
    Kimatsu2a ka = new Kimatsu2a(11, 22);
    setVal(a, b, ka);
    System.out.println("a = " + a
       + ", b[1] = " + b[1] + ", b[2] = " + b[2]
       + ", ka.fieldA = " + ka.fieldA + ", ka.fieldB = " + ka.fieldB);
  }
  static void setVal(int x, int[] y, Kimatsu2a z) {
    x = 21; y[2] = 27; z.fieldA = 28;
  }
}
```

(クラス Kimatsu2 のメソッド setVal() が，クラスメソッドになっている (static が付いている) が，特に気にする必要はない。)

問３． 次の立体の体積や表面積を計算し，その値を昇順にソートして表示するプログラムに書いた (仕様は 2 ページ後)。

```
class Solid {
  String name;
  private double volume, surface;
  Solid(String name) {
    ____________(a)____________name   = name;
  }
  void setVS(double v, double s) {
    volume = v;
    surface = s;
  }
  void printS() {
    System.out.println("表面積" + surface + "の" + name);
  }
  static boolean gt(Solid s1, Solid s2, int type) {
    if (type == 0) return (s1.volume > s2.volume);
    else ____________(b)____________;
  }
}

class Cylinder extends Solid { // 円柱
  double radius, height;
  Cylinder(double r, double h) {
    super("円柱");
    radius = r;
    height = h;
    double v = Math.PI * radius * radius * height;
    double s = 2.0 * Math.PI * (radius * radius + radius * height);
    setVS(v, s);
  }
}

class Cone extends Solid { // 円錐
  double radius, height;
  Cone(double r, double h) {
    super("円錐");
    radius = r;
    height = h;
    double edge = Math.sqrt(radius * radius + height * height);
    double v = Math.PI * radius * radius * height / 3.0;
    double s = ____________(c)____________;
    setVS(v, s);
  }
}
```

```
class Rectangular extends Solid { // 直方体
  double depth, width, height;
  Rectangular(double x, double y, double z) {
    super("直方体");
    depth = x; width = y; height = z;
    double v = x * y * z;
    double s = 2.0 * (x * y + y * z + z * x);
    setVS(v, s);
  }
  void printTotalLength() {
    System.out.println("Total length of edges = " + (4.0 * (depth + width + height)));
  }
}

class Kimatsu3 {
  public static void main(String[] args) {
    int nData = 6;
    Solid solid[] = new Solid[nData];
    solid[0] = new Cylinder(2.0, 1.0);
    solid[1] = new Cone(2.0, 2.0);
    solid[2] = new Rectangular(2.0, 2.0, 2.0);
    solid[3] = new Cylinder(2.0, 4.0);
    solid[4] = new Cone(2.0, 4.0);
    solid[5] = new Rectangular(2.0, 4.0, 2.0);

    //選択ソート
    for (int l = 0 ; l < nData - 1 ; ++l) {
      int minL = l;
      for (int j = ____(d)____ ; j < nData ; ++j) {
        if (____(e)____) minL = j;
      }
      Solid tmp   = ____(f)____;
      solid[minL] = ____(g)____;
      solid[l]    = ____(h)____;
    }
    // 表示
    for (int l = 0 ; l < nData ; ++l) {
      ____(i)____;
    }
    //solid[2].printTotalLength();
  }
}
```

仕様

(1) クラス Solid のフィールドは，その名前を格納する String 型の変数 name と，体積と表面積を格納する double 型の変数 volume と surface である。コンストラクタ Solid(Stirng name) は，String 型の変数 name を引数として，フィールドの name に，引数の name に格納されているものを格納する。メソッド setVS(doulbe v，double s) は，引数 v と s をフィールド volume と surface に格納する。メソッド printS() は，自身の表面積を名前と共に表示する。クラスメソッド gt(Solid s1，Solid s2，int type) は，s1 の方が s2 よりも大きいときに true を返す。その大きさは，type が 0 の場合は体積によって，それ以外の場合は表面積によって評価される。

(2) クラス Cylinder は，クラス Solid を継承している。オリジナルのフィールドは，double 型の変数 radius，height である。コンストラクタ Cylinder(double r，double h) は，引数"円柱"でスーパークラスのコンストラクタを呼び出し，引数 r，h の値を，それぞれフィールド radius，height に格納する。そして，体積と表面積を計算して，メソッド setVS() によって，それぞれのフィールドに格納する。

(3) クラス Cone は，クラス Solid を継承している。オリジナルのフィールドは，double 型の変数 radius，height である。コンストラクタ Cone(double r，double h) は，引数"円錐"でスーパークラスのコンストラクタを呼び出し，引数 r，h の値を，それぞれフィールド radius，height に格納する。そして，体積と表面積を計算して，メソッド setVS() によって，それぞれのフィールドに格納する。

(3) クラス Rectangular は，クラス Solid を継承している。オリジナルのフィールドは，double 型の変数 depth，width，height である。コンストラクタ Rectangular(double x，double y，double z) は，引数"直方体"でスーパークラスのコンストラクタを呼び出し，引数 x，y，z の値を，それぞれフィールド depth，width，height に格納する。そして，体積と表面積を計算して，メソッド setVS() によって，それぞれのフィールドに格納する。

(4) Kimatsu3 は，メインメソッドのためのクラスである。Solid の参照情報を格納する配列 solid を宣言し，Cylinder，Cone，Rectangular のオブジェクトを作成し，solid[0]〜solid[5] に格納している。格納されたデータを**表面積で昇順に**ソートして表示する。

(5) Math.PI は double の精度での円周率を，Math.sqrt(x) は x の正の平方根を戻り値とするクラスメソッドである。

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) このプログラムの終わりの方にあるの//を削除するとどのようなことが生じるか答えよ。

(ウ) (イ) のようになる理由を答えよ。

(エ) この文を修正して，正しく動作するようにするためにはどうすれば良いか答えよ。

解答用紙

学科・類：　　学籍番号：　　名前：

問1.

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

(k)

(l)

(m)

(n)

(o), (p)

(q)

(r)

問2.

表示されるもの

理由

問3.

(ア)

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(イ)

(ウ)

(エ)

解答例

問 1 .

(a) `int arrayInt[] = {3, 2, 4, 1, 3, 5};`

(b) シグネチャー

(c) メソッドの識別子 (メソッドの名前)

(d) 型

(e) オーバーロード

(f) ローカル変数

(g) `this`

(h) 継承 (inheritance)

(i) `class ClassB extends ClassA`

(j) `ClassA`

(k) オーバーライド

(l) 戻り値の型

(m) ポリモーフィズム (多態性)

(n) キャスト

(o), (p) 選択ソート，マージソート，バカソートなどから 2 つ

(q) 選択ソート，バブルソートなどから 1 つ

(r) クイックソート，マージソート，ヒープソートなどから 1 つ

問 2 .

表示されるもの

```
a = 3, b[1] = 2, b[2] = 27, ka.fieldA = 28, ka.fieldB = 22
```

理由

メソッド `setVal()` を呼び出す前に，`a` には 3，`b[1]` には 2，`b[2]` には 3，`ka.fieldA` には 11，`ka.fieldA` には 22 という値が代入される。そして，メソッド `setVal()` に渡されるものは，`a` の値，配列 `b` の参照情報，オブジェクト `ka` の参照情報である。メソッド `setVal()` を呼び出すときに，`a`，`b`，`ka` とは異なる領域にある，`x`，`y`，`z` に，対応する引数の値がコピーされる。従って，メソッドから戻ってきたときは，`a` 値は変わらない。`b` と `y` は同じ参照情報が格納されているため，`b[2]` と `y[2]` は同じ領域を使う。従って，`y[2]` を 27 書き換えると `b[2]` の値も 27 に変わる。同様に，`ka` と `z` は同じ参照情報が格納されているため，`ka.fieldA` と `z.fieldA` は同じ領域を使う。従って，`ka.fieldA` を 28 に書き換えると `ka.fieldA` の値も 28 に変わる。それ以外の変数は変わらない。

問3.

(ア)

(a) `this.`

(b) `(s1.surface > s2.surface)`

(c) `Math.PI * radius * (edge + radius)`

(d) `1 + 1`

(e) `Solid.gt(solid[minL], solid[j], 1)`

(f) `solid[minL]`

(g) `solid[l]`

(h) `tmp`

(i) `solid[l].printS()`

(イ)

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

(ウ)

変数 `solid[2]` にはクラス `Solid` のオブジェクトの参照情報が格納されている。
そのためコンパイル時には，クラス `Solid` の中からだけからメソッドを探すので，
`Rectangular` のメソッドが見つからないためである。

(エ)

この文を，
`((Rectangular) solid[2]).printTotalLength();`
とする