平成２３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。

- 倍精度浮動小数点型の配列変数 array を宣言し，その要素 10 個分の領域を確保し，その参照情報を array に代入するためには，(a) と記述する。このとき，array の要素として，(b) から (c) までを使うことが可能である。
- メソッドを区別するシグネチャは，メソッドの (d) とその (e) の (f) の並びからなる。一つのクラスの中で，(d) が同じで，異なる (f) の並びをもつメソッドを定義することを，(g) すると言う。
- ある変数を利用できる範囲を，その変数の (h) (カタカナ 4 文字) と呼ぶ。ローカル変数の (h) は，その変数を (i) してから，その (j) の終了までである。識別子 a 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 (h) に含まれる場合，ローカル変数の方を参照したい場合は (k) と記述し，フィールドの方を参照したい場合は (l) と記述する。
- クラス Parts を元に，拡張・変更した新しいクラス Resistor を宣言することを，Parts を (m) して，Resistor を宣言すると言う。このとき，サブクラスは (n) であり，スーパークラスは (o) である。
- サブクラス (スーパークラスの誤り) にあるメソッドと同じシグネチャを持ったメソッドを，スーパークラス (サブクラスの誤り) で定義することをメソッドの (p) と呼ぶ。
- オブジェクト指向におけるポリモーフィズムとは，上の例で示せば，クラス (q) の参照情報を，クラス (r) の型の変数に代入することができることである。
- 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，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，計算量が N^2 のオーダになるソートとして (s) が，計算量が $N \log_2 N$ のオーダになるソートとして (t) がある。

問２． パソコン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// p2.printWeight(); 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，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。

```
class Pasocon {
  int cpuType;
  int memorySize;
  int hddSize;
  Pasocon(int cpuType, int memorySize, int hddSize) {
    this.cpuType = cpuType; this.memorySize = memorySize;
    this.hddSize = hddSize;
  }
  void printInf() {
    System.out.println("cpuType = " + cpuType + " memorySize = " + memorySize
        + " httSize = " + hddSize);
  }
}
```

(次ページに続く)

```
class NotePasocon extends Pasocon {
  double weight;
  double lcdSize;
  NotePasocon(int cpuType, int memorySize, int hddSize, double weight, double lcdSize) {
    super(cpuType, memorySize, hddSize);
    this.weight = weight; this.lcdSize = lcdSize;
  }
  void printInf() {
    System.out.println("cpuType = " + cpuType + " memorySize = " + memorySize
        + " httSize = " + hddSize + " lcdSize = " + lcdSize);
  }
  void printWeight() {
    System.out.println("weight = " + weight);
  }
}

class Kimatsu2 {
  public static void main(String[] args) {
    Pasocon p1 = new Pasocon(1, 16, 2000);
    Pasocon p2 = new Pasocon(2,  8, 1000);
    Pasocon p3 = new NotePasocon(4, 4, 300, 1.5, 13.3);
    NotePasocon p4 = new NotePasocon(6, 4, 500, 2.3, 15.4);
    NotePasocon p5;
    p2 = p3; p5 = p4;

    p1.printInf();
    p2.printInf();
    p4.printWeight();
    p5.printInf();
    // p2.printWeight();
  }
}
```

問３． 分数の四則演算を計算し，分数の値を降順にソートして表示する次のプログラムを記述した。

仕様

(1) クラス FNumber のフィールドは，それぞれ，分子と分母を格納する int 型の変数 numa と deno である。

(1) FNumber の deno の値は正として利用するものとする。

(1) コンストラクタ FNumber(int nume, int deno) は，引数の値を対応するフィールドに格納する。

(2) メソッド reduct() は，自分のオブジェクトの約分を行う。

(3) メソッド GCD(int a, int b) は，a と b (両方とも正の整数) の最大公約数を計算し，戻り値として返す。

(3) メソッド add(FNumber x, FNumber y), sub(FNumber x, FNumber y), mul(FNumber x, FNumber y), div(FNumber x, FNumber y) は，それぞれ，x と y を分数として加算，減算，乗算，除算を計算し，自身のオブジェクトに格納する。

(4) gt(FNumber x) は，自身が表す分数としての値が x が表す値よりも大きければ true を返し，そうでなければ false を返すメソッドである。

(4) クラス Kimatsu3 は，クラス FNumber の配列とそれぞれの要素を作成し，値をコンストラクターや計算によって与え，配列を分数の値として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。

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) このプログラムで表示されるものを答えよ。

```
class FNumber {
  int nume, deno; //分子, 分母
  FNumber(int nume, int deno) {
    this.nume = nume; this.deno = deno;
  }
  void reduct() {
    if (nume == 0) {
      deno = 1;
    } else if (nume < 0) {
      int gcd = GCD(-nume, deno);
      nume /= gcd; deno /= gcd;
    } else {
      int gcd = GCD(nume, deno);
      nume /= gcd; deno /= gcd;
    }
  }
  int GCD(int a, int b) {
    ____(a)____;
      ____(b)____;
      ____(c)____;
      ____(d)____;
    ____(e)____;
    return a;
  }
  void add(FNumber x, FNumber y) {
    nume = x.nume * y.deno + x.deno * y.nume;
    deno = x.deno * y.deno;
    reduct();
  }
  void sub(FNumber x, FNumber y) {
    nume = x.nume * y.deno - x.deno * y.nume;
    deno = x.deno * y.deno;
    reduct();
  }
  void mul(FNumber x, FNumber y) {
    nume = x.nume * y.nume;
    deno = x.deno * y.deno;
    reduct();
  }
  void div(FNumber x, FNumber y) {
    nume = x.nume * y.deno;
    deno = x.deno * y.nume;
    reduct();
  }
  boolean gt(FNumber x) {
    return (nume * x.deno - deno * x.nume > 0);
  }
```

```
  void print() {
    System.out.println(nume + "/" + deno);
  }
}
class Kimatsu3 {
  public static void main(String[] args) {
    int N = 6;
    FNumber fn[] = new FNumber[N];
    fn[0] = new FNumber(2, 3);
    fn[1] = new FNumber(1, 2);
    for(int i = 2 ; i < 6 ; ++i) fn[i] = new FNumber(0, 1);
    fn[2].add(fn[0], fn[1]);
    fn[3].sub(fn[0], fn[1]);
    fn[4].mul(fn[0], fn[1]);
    fn[5].div(fn[0], fn[1]);

    for (int k = 0 ; k < N - 1 ; ++k) {
      ____________(f)____________
      for (int l = _____(g)_____ ; l < N ; ++l) {
        if (____________(h)____________)) maxInd = l;
      }
      ____________(i)____________;
      fn[maxInd] = fn[k];
      fn[k] = t;
    }
    for (int i = 0 ; i < N ; ++i) fn[i].print();
  }
}
```

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)

(s)

(t)

問2.

(ア)

(イ)

(ウ)

(エ)

問3.

(ア)

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(イ)

解答用紙

学科・類：　　学籍番号：　　名前：

問1.

(a) `double array[] = new double[10];`

(b) `aray[0]`

(c) `aray[9]`

(d) 名前 (識別子)

(e) 引数

(f) 型

(g) オーバーロード

(h) スコープ

(i) 宣言

(j) ブロック

(k) `a`

(l) `this.a`

(m) 継承 (inheritance)

(n) `Resistor`

(o) `Parts`

(p) オーバーライド問題に誤りがあったため単なる「宣言」or「定義」でも可

(q) `Resistor`

(r) `Parts`

(s) 選択ソート，バブルソートなどから 1 つ

(t) 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.

(ア)

```
cpuType = 1 memorySize = 16 httSize = 2000
cpuType = 4 memorySize = 4 httSize = 300 lcdSize = 13.3
weight = 2.3
cpuType = 6 memorySize = 4 httSize = 500 lcdSize = 15.4
```

(イ)

コンパイル時に，メソッド `printWeight()` 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 NotePasocon 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。
p2 の型が Pasocon であるため，コンパイル時はクラス Pasocon の中からだけからメソッドを
探すが，クラス Pasocon にはメソッド printWeight() 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，
((NotePasocon) p2).printWeight();
とする

問3.

(ア)

(a) `while(b > 0) {`

(b) `int res = a % b;`

(c) `a = b;`

(d) `b = res;`

(e) `}`

(f) `int maxInd = k;`

(g) `k + 1`

(h) `fn[l].gt(fn[maxInd])`

(i) `FNumber t = fn[maxInd];`

(イ)

```
4/3
7/6
2/3
1/2
1/3
1/6
```