

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

- 64bit の整数型の配列変数 lList を宣言し、その要素 4 個分の領域を確保し、その参照情報を lList に代入するためには、(a) と記述する。このとき、配列 lList 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 lList の宣言、領域確保と、要素の値を順番に 3, 2, 12345678900, 8 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 (h) の並びからなる。一つのクラスの中で、(f) が同じで、異なる (g) の (h) の並びをもつメソッドを定義することを、メソッドの (i) と言う。
- 変数の (j) とは、その変数を使うことができるプログラムにおける範囲である。識別子 a の変数が、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 (j) 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) と記述する。
- (l) は、オブジェクトを作成する特別なでありメソッドである。クラス Parts のクラス型変数 prt1 を宣言し、デフォルトの (l) でオブジェクトを生成し、その参照情報を prt1 に代入するためには、(m) と記述する。また、prt1 のフィールド price に 210 を代入するときには、(n) と記述する。自分のクラスの (g) の (h) の並びが異なる (l) を呼び出すときには、キーワード (o) を用いる。
- 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 (p) (漢字 2 文字) して、Resistor を宣言すると言う。このとき、クラス (q) をサブクラスまたは (r) クラス、クラス (s) をスーパークラスまたは (t) クラスと呼ぶ。サブクラスのコンストラクタからスーパークラスのコンストラクタを呼び出すときには、キーワード (u) を用いる ((q), (s) には Parts または Resistor が入る)。
- 親クラスにあるメソッドと同じ (e) を持つメソッドを、子クラスで定義することをメソッドの (v) と呼ぶ。
- メソッドの引数に関して、呼び出されるメソッドの定義で使われる引数を (w) 引数、呼び出す方で使う引数を (x) 引数と言う。
- 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、バカソート (俗称)、(y)、(z)、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、マージソートの計算量のオーダーは (A) であり、ヒープソートの計算量のオーダーは (B) である。
- 可視性を決めるキーワードである protected によって、宣言時に修飾されたフィールドやメソッドは、それを (C) だけからしか参照できない。
- クラス Parts のオブジェクトの参照情報 4 つを格納できる配列変数 prtL の宣言、領域確保を行うためには、(D) と記述する。この時点で、prtL のための Parts のオブジェクトは生成されて (E) (いる or いない)。

問 2. プリンタやインクの情報扱う次のプログラムを作成した。このプログラムを実行させたときに表示するものを示せ。

```
class Kimatsu2 {
    public static void main(String[] args) {
        Ink      ink1 = new Ink("LC-5B", 800), ink2;
        ColorInk ink3 = new ColorInk("LC-6M", "マゼンダ", 1000);
        ink2 = ink3;
        ink1.printInf(); ink2.printInf(); ink3.printInf();
        Printer  prn1 = new Printer("Ganon iP100", ink1);
        Printer  prn2 = new ColorPrinter("EBSON EP-100", ink1, ink3);
        ColorPrinter prn3 = new ColorPrinter("Sister DCP-100", ink2, ink3);
        prn1.printInf(); prn2.printInf(); prn3.printInf();
    }
}
```

(次ページに続く)

```

class Printer {
    String name;
    Ink    ink;
    Printer(String name, Ink ink) {
        this.name    = name;
        this.ink     = ink;
    }
    void printInf() {
        System.out.println("名前は" + name + "です。");
        ink.printInf();
    }
}

class ColorPrinter extends Printer {
    Ink colorInk;
    ColorPrinter(String name, Ink ink, ColorInk colorInk) {
        super(name, ink);
        this.colorInk = colorInk;
    }
    void printInf() {
        super.printInf();
        colorInk.printInf();
    }
}

class Ink {
    String type;
    int    price;
    Ink(String type, int price) {
        this.type    = type;
        this.price   = price;
    }
    void printInf() {
        System.out.println(type + "の黒インクで," + price + "円です。");
    }
}

class ColorInk extends Ink {
    String color;
    ColorInk(String typeName, String color, int price) {
        super(typeName, price);
        this.color = color;
    }
    void printInf() {
        System.out.println(type + "のインクで," + price + "円です。色は" + color + "です。");
    }
}

```

問3 . 次のプログラムは、両親が家に残す兄弟を決めるために、兄弟のニート度を相対的に評価して、値の小さなものから表示するものである。その仕様を次に示す。

- (1) クラス Person は、個人の情報を記録する。クラス Person のフィールドは、名前を格納する String 型の変数 name と、能力の高さの値を格納する int 型の変数 aP, 意思の強さの値を格納する int 型の変数 wP, ずるの強さの値を格納する int 型の変数 zP, 計算したニート度を格納する int 型の変数 nP である。
- (2) Player のメソッドは次の2つである。Person(String name, int aP, int wP, int zP) がコンストラクタでそれぞれの引数をフィールドに格納し、nP を 0 に初期化する。print() は、その個人の情報を表示する。
- (3) クラス Brother は兄弟全員の情報を格納し、ニート度の計算とソートを行うことができる。クラス Brother のフィールドは、兄弟の人数を格納する nPersons, 兄弟の個人情報格納する Person の配列 pL である。
- (4) Brother のメソッドは次の4つである。Brother(int nPersons) はコンストラクタで、引数をフィールドに格納し、人数分の Person の領域を確保し、参照情報を pL に格納する。consPerson(int l, String name, int aP, int wP, int zP) は、引数の情報を使って Person をコンストラクトし、その参照情報を配列の l 番目の要素に代入する。calNPoint(int l) は、pL[l] とその他の兄弟と比較し、そのニート度を計算して、その値を返す。sort() は、配列 pL をそのニート度の昇順にソートする。print() は、兄弟全員の情報を配列の順番に表示する。
- (5) ニート度は、基本を 0 として、能力の高さの値と意思の強さの値をかけたものが、他人より低い場合は 1 人当たり+3 する。ずるの強さの値が他の人より高い場合は一人当たり-1 する。これを他の兄弟の人すべてに渡って得た値を合算したものである。
- (8) クラス Kimatsu3 は、メインメソッドのためのクラスである。メインメソッド main(String[] args) で、Brother を 6 人でコンストラクトして br に格納する。次に、各人の情報をセットした後、兄弟の各人のニート度を計算して、その人のフィールド nP の代入する。そして、br の要素のニート度で昇順にソートを行い、兄弟全員の情報をソート結果の順番に表示する。
- (9) 6 人の人の情報は以下のとおりである。

氏名	能力	意志力	ずるさ	氏名	能力	意志力	ずるさ
おそ松	2	2	2	一松	4	3	1
カラ松	2	3	2	十四松	1	5	1
チョロ松	4	2	1	トド松	5	4	5

このとき、無駄なくできるだけ一般性を持たせて、上記の仕様を満たすために、プログラム中の下線部 (a) ~ (l) に書き入れるべきものを答えよ。

```

class Person {
    String name;
    int aP, wP, zP, nP;
    Person(String name, int aP, int wP, int zP) {
        _____ (a) _____ .name = new String(name);
        _____ (a) _____ .aP = aP;
        _____ (a) _____ .wP = wP;
        _____ (a) _____ .zP = zP;
        nP = _____ (b) _____ ;
    }
    void print() {
        System.out.println(name + "のニートポイントは"+ nP + "です");
    }
}

```

(次ページに続く)

```

class Brother {
    int    nPersons;
    Person pL[];
    Brother(int nPersons) {
        _____ (a) _____ .nPersons = nPersons;
        pL = _____ (c) _____ Person[nPersons];
    }
    void constPerson(int l, String name, int aP, int wP, int zP) {
        pL[l] = new _____ (d) _____ (name, aP, wP, zP);
    }
    int calNPoint(int l) {
        int aw = 0, z = 0;
        for (int k = 0 ; k < nPersons ; ++k) {
            if (k == 1) _____ (e) _____ ;
            if (pL[l].aP * pL[l].wP < pL[k].aP * pL[k].wP) ++aw;
            if (pL[l].zP > pL[k].zP) ++z;
        }
        _____ (f) _____ 3 * aw - z;
    }
    void sort() {
        for (int l = 0 ; l < nPersons - 1 ; ++l) {
            int m = l;
            for (int k = _____ (g) _____ ; k < nPersons ; ++k) {
                if ( _____ (h) _____ ) m = k;
            }
            _____ (i) _____ tmp = pL[m];
            pL[m] = pL[l];
            pL[l] = tmp;
        }
    }
    void print() {
        for (int i = 0 ; i < nPersons ; ++i) _____ (j) _____ ;
    }
}

class Kimatsu3 {
    public static void main(String[] args) {
        int nPersons = 6;
        Brother br = new Brother(nPersons);
        br.constPerson(0, " おそ松", 2, 2, 2);
        br.constPerson(1, " カラ松", 2, 3, 2);
        br.constPerson(2, " チヨロ松", 4, 2, 1);
        br.constPerson(3, " 一松", 4, 3, 1);
        br.constPerson(4, " 十四松", 1, 5, 1);
        br.constPerson(5, " トド松", 5, 4, 5);
        for (int l = 0 ; l < nPersons ; ++l) br.pL[l].nP = _____ (k) _____ ;
        _____ (l) _____ ; br.print();
    }
}

```

学科・類： _____ 学籍番号： _____ 名前： _____
問 1 . $2 \times 31 = 62$ 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)
- (y)
- (z)
- (A)
- (B)
- (C)
- (D)
- (E)

問2 . (14 点)



問3 . $2 \times 12 = 24$ 点

- (a)
- (b)
- (c)
- (d)
- (e)
- (f)
- (g)
- (h)
- (i)
- (j)
- (k)
- (l)

解答用紙

学科・類： _____ 学籍番号： _____ 名前： _____

問1 . 2 × 31 = 62 点

- (a) `long lList[] = new long[4];`
- (b) `0 (lList[0])`
- (c) `3 (lList[4])`
- (d) `long lList[] = {3L, 2L, 12345678900L, 8L};` (int で収まる場合は, L は付加しても付加しなくても良い。)
- (e) シグネチャ
- (f) 名前 (識別子)
- (g) 引数
- (h) 型
- (i) オーバーロード
- (j) スコープ
- (k) a
- (l) コンストラクタ
- (m) `Parts prt1 = new Parts();`
- (n) `prt1.price = 210;`
- (o) `this`
- (p) 継承 (inheritance)
- (q) `Resistor`
- (r) 子
- (s) `Parts`
- (t) 親
- (u) `super`
- (v) オーバーライド
- (w) 仮
- (x) 実
- (y) 選択ソート ((y) と (z) は順不同)
- (z) バブルソート
- (A) $N \log_2 N$
- (B) $N \log_2 N$
- (C) 継承したクラスまたはまたは同じパッケージのクラス
- (D) `Parts prtL[] = new Parts[4];`
- (E) いない。

問2 . (14 点)

LC-5B の黒インクで , 800 円です。
LC-6M のインクで , 1000 円です。色はマゼンダです。
LC-6M のインクで , 1000 円です。色はマゼンダです。
名前は Ganon iP100 です。
LC-5B の黒インクで , 800 円です。
名前は EBSON EP-100 です。
LC-5B の黒インクで , 800 円です。
LC-6M のインクで , 1000 円です。色はマゼンダです。
名前は Sister DCP-100 です。
LC-6M のインクで , 1000 円です。色はマゼンダです。
LC-6M のインクで , 1000 円です。色はマゼンダです。

問3 . $2 \times 12 = 24$ 点

- (a) this
- (b) 0.0
- (c) new
- (d) Person
- (e) continue
- (f) return
- (g) 1 + 1
- (h) `pL[k].nP < pL[m].nP`
- (i) Person
- (j) `pL[i].print()`
- (k) `br.calNPoint(1)`
- (l) `br.sort()`