

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

- 32bit の浮動小数点型の配列変数 rL 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 rL に代入するためには、(a) と記述する。このとき、配列 rL 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 rL の宣言、領域確保と、要素の値を順番に 3.2, 2.4, 6.3, 1.7 で初期化することを同時にを行うためには、(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、(i) と言う。
- ある変数を利用できる範囲を、その変数の (j) (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 (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 のクラス型変数 pt において、そのフィールド price に 230 を代入するときには、(m) と記述する。コンストラクタは、オブジェクトを作成する特別な (n) であり、自分のクラスの (g) の (h) の並びが異なるコンストラクタ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 Reactor を宣言することを、Parts を (p) (漢字 2 文字) して、Reactor を宣言すると言う。このとき、クラス (q) を子クラスまたは (r) クラス、クラス (s) を親クラスまたは (t) クラスと呼ぶ。子クラスのコンストラクタから親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 Reactor 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。
- あるクラス型変数 obj を使ってメソッドを呼び出すとき、ポリモフィズムにより、obj の (w) のクラスではなく、それが (x) しているクラスで定義されるメソッドを呼び出す。
- 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)、選択ソート、バブルソート、(y)、(z)、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 (A) であり、ヒープソートの計算量のオーダーは (B) である。
- フィールドやメソッドの可視性を決めるキーワードの 1 つに、(C) がある。
- クラス Parts のオブジェクトの参照情報 3 つを格納できる配列変数 pL の宣言、領域確保を行い、Parts のデフォルトのコンストラクタでオブジェクトを 3 つ作成し、その配列にそれぞれの要素に格納するためには、(D) と記述する (複数行使ってよい)。

問 2. 飛行機やエンジンの仕様を表示する次のプログラムを作成した。このプログラムが表示するものを示せ。

```
class Kimatsu2 {
    public static void main(String[] args) {
        Engine eg1 = new Engine("AE 2100", 873.0), eg2;
        JetEngine jetEg = new JetEngine("CF6-80C2", 4472.0, 28802.0);
        eg1.printInf();      jetEg.printInf();
        eg2 = jetEg;         eg2.printInf();
        Plane pl1 = new Plane("C-130J", 4, eg1);
        Plane pl2 = new Plane("B767", 2, eg2);
        JetPlane jetPl = new JetPlane("B747", 4, jetEg);
        pl1.printInf();      pl2.printInf();      jetPl.printInf();
    }
}
```

(次ページに続く)

```

class Plane {
    String name;
    int    numberOfEngine;
    Engine engine;
    Plane(String name, int numberOfEngine, Engine engine) {
        this.name          = name;
        this.numberOfEngine = numberOfEngine;
        this.engine         = engine;
    }
    void printInf() {
        System.out.println("飛行機：" + name + " , 次のエンジンを" + numberOfEngine + "個搭載している。");
        engine.printInf();
    }
}

class JetPlane extends Plane {
    JetPlane(String name, int numberOfEngine, JetEngine engine) {
        super(name, numberOfEngine, engine);
    }
    void printInf() {
        System.out.println("ジェット機：" + name + " , 次のジェットエンジンを"
            + numberOfEngine + "個搭載している。");
        engine.printInf();
    }
}

class Engine {
    String name;
    double weight;
    Engine(String name, double weight) {
        this.name = name;
        this.weight = weight;
    }
    void printInf() {
        System.out.println("飛行機用エンジン。名前：" + name + " , 重量：" + weight + "kg");
        System.out.println();
    }
}

class JetEngine extends Engine {
    double maxTrust;
    JetEngine(String name, double weight, double maxTrust) {
        super(name, weight);
        this.maxTrust = maxTrust;
    }
    void printInf() {
        System.out.println("エンジンはジェットだ。名前は" + name + "で , 重量：" + weight + "だ。");
        System.out.println();
    }
}

```

問3. 次のプログラムは、学生の身長と体重から求めた BMI(肥満指数) の過去の平均値や最大値および直近の値を使って、降順にソートして表示するものである。その仕様を次に示す。

- (1) クラス StudentBmi は、1 人の学生の身長と体重の履歴を記録し、BMI を計算し、そのオブジェクトが示す学生の BMI と他の学生の BMI を比較することができる。
- (2) クラス StudentBmi のフィールドは、名前を格納する String 型の変数 name と、記憶した身長と体重の数を格納する int 型の変数 nData、身長と体重をそれぞれ格納する double 型の配列変数 heightL と weightL、BMI の最大値と平均値および直近の値を格納する double 型の変数 maxBmi と aveBmi および lastBmi である。
- (3) StudentBmi のコンストラクタ StudentBmi(String name, int nData) は、学生の名前 name と記録する身長と体重の数 nData を引数として、それぞれを対応するフィールドに格納すると共に、nData 個分の double 型の領域を 2 セット確保し、配列変数 heightL と weightL にその参照情報を代入する。
- (4) メソッド void setHW(int index, double height, double weight) は、引数として与えられる身長 height と体重 weight を、heightL と weightL のインデックスが引数 index である要素に格納する。
- (5) メソッド void calBmi() は、フィールドに格納された身長の体重のデータから BMI を計算し、履歴での BMI の最大値と平均値および直近の値を、それぞれ、maxBmi と aveBmi および lastBmi に格納する。
- (6) メソッド boolean largerBmi(int type, StudentBmi rival) は、そのオブジェクトの学生が引数で指定される学生 rival より、type が 0 のときは、直近の BMI が大きい場合、type が 1 のときは、BMI の履歴の最大値が大きい場合、またはそれが同じ場合には BMI の履歴の平均値が大きい場合に true を返し、そうでないときに false を返す。
- (7) メソッド print() は、その学生の名前、直近の BMI、履歴における BMI の最大値と平均値を表示する。
- (8) クラス Kimatsu3 は、6 人分の StudentBmi の参照情報を格納する配列の宣言と領域確保を行い、6 人分のそのオブジェクトを作成し、配列の要素に代入し、身長体重の履歴をセットし、BMI を計算し、type = 1 の大小関係で降順にソートし、表示するプログラムである。6 人の名前と身長と体重の履歴を示す。ここで、N.A. はデータがないことを表す。

氏名	1 回目		2 回目		3 回目	
	身長	体重	身長	体重	身長	体重
宿海 仁太	1.24	30.0	1.65	55.0	N.A.	N.A.
本間芽衣子	1.15	22.0	1.27	0.1	1.47	36.0
安城 鳴子	1.26	31.0	1.64	57.0	N.A.	N.A.
松雪 集	1.32	35.0	1.81	68.0	N.A.	N.A.
鶴見知利子	1.27	32.0	1.70	58.0	N.A.	N.A.
久川 鉄道	1.20	26.0	1.85	100.0	N.A.	N.A.

このとき、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) ~ (l) に書き入れるべきものを答えよ。

```
class Kimatsu3 {
    public static void main(String[] args) {
        int N = 6;
        StudentBmi sbL[] = new StudentBmi[N];
        sbL[0] = new StudentBmi("宿海 仁太", 2); _____ (a) _____;
        sbL[2] = new StudentBmi("安城 鳴子", 2); sbL[3] = new StudentBmi("松雪 集", 2);
        sbL[4] = new StudentBmi("鶴見知利子", 2); sbL[5] = new StudentBmi("久川 鉄道", 2);
        sbL[0].setHW(0, 1.24, 30.0); sbL[0].setHW(1, 1.65, 55.0);
        sbL[1].setHW(0, 1.15, 22.0); sbL[1].setHW(1, 1.27, 0.1); _____ (b) _____;
        sbL[2].setHW(0, 1.26, 31.0); sbL[2].setHW(1, 1.64, 57.0);
        sbL[3].setHW(0, 1.32, 35.0); sbL[3].setHW(1, 1.81, 68.0);
        sbL[4].setHW(0, 1.27, 32.0); sbL[4].setHW(1, 1.70, 58.0);
        sbL[5].setHW(0, 1.20, 26.0); sbL[5].setHW(1, 1.85, 100.0);

        for (int i = 0 ; i < N ; ++i) sbL[i].calBmi();
    }
}
```

(次ページに続く)

```

for (int k = 0 ; k < ____ (c) ____ ; ++k) {
    int topInd = k;
    for (int l = ____ (d) ____ ; l < N ; ++l) {
        if ( ____ (e) ____ ) topInd = l;
    }
    StudentBmi tmp = ____ (f) ____;
    sbL[topInd] = ____ (g) ____;
    sbL[k] = ____ (h) ____;
}
for (int i = 0 ; i < N ; ++i) sbL[i].print();
}
}

class StudentBmi {
    String name;
    int nData;
    double heightL[], weightL[];
    double maxBmi, aveBmi, lastBmi;
    StudentBmi(String name, int nData) {
        this.name = new String(name);          this.nData = nData;
        heightL = new double[nData];          weightL = new double[nData];
    }
    void setHW(int index, double height, double weight) {
        heightL[index] = height; ____ (i) ____;
    }
    void calBmi() {
        double tmpBmi = 0.0;
        maxBmi = aveBmi = 0.0;
        for (int i = 0 ; i < nData ; ++i) {
            aveBmi += tmpBmi = weightL[i] / (heightL[i] * heightL[i]);
            if ( ____ (j) ____ ) maxBmi = tmpBmi;
        }
        lastBmi = tmpBmi; ____ (k) ____; // aveBmi の計算
    }
    boolean largerBmi(int type, StudentBmi rival) {
        boolean result = false;
        switch (type) {
            case 0:
                result = lastBmi > rival.lastBmi;
                break;
            case 1:
                result = ____ (l) ____;
                break;
        }
        return result;
    }
    void print() {
        System.out.printf(''%s 最終 BMI: %5.2f 最大 BMI: %5.2f 平均 BMI: %5.2f \n'',
            name, lastBmi, maxBmi, aveBmi);
    }
}

```

学科・類：

学籍番号：

名前：

問 1 . $2 \times 30 = 60$ 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)

問 2 . (16 点)

問 3 . $2 \times 12 = 24$ 点

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

学科・類：

学籍番号：

名前：

問1 . $2 \times 30 = 60$ 点

- (a) `float rL[] = new float[4];`
- (b) `0 (rL[0])`
- (c) `3 (rL[3])`
- (d) `float rL[] = {3.2, 2.4, 6.3, 1.7};`
- (e) シグネチャ
- (f) 名前
- (g) 引数
- (h) 型
- (i) オーバーロード
- (j) スコープ
- (k) `x`
- (l) `this.x`
- (m) `pt.price = 230;`
- (n) メソッド
- (o) `this`
- (p) 継承 (inheritance)
- (q) `Reactor`
- (r) サブ
- (s) `Parts`
- (t) スーパー
- (u) `super`
- (v) オーバーライド
- (w) 型
- (x) 参照
- (y) クイックソート ((y) と (z) は順不同)
- (z) マージソート
- (A) N^2
- (B) $N \log_2 N$
- (C) `public, protected, private` の中の1つ

(D)

```
Parts pL[] = new Parts[3];  
pL[0]      = new Parts();  
pL[1]      = new Parts();  
pL[2]      = new Parts();
```

問 2 . (16 点)

飛行機用エンジン。名前：AE 2100，重量：873.0kg

エンジンはジェットだ。名前は CF6-80C2 で，重量：4472.0 だ。

エンジンはジェットだ。名前は CF6-80C2 で，重量：4472.0 だ。

飛行機：C-130J，次のエンジンを 4 個搭載している。

飛行機用エンジン。名前：AE 2100，重量：873.0kg

飛行機：B767，次のエンジンを 2 個搭載している。

エンジンはジェットだ。名前は CF6-80C2 で，重量：4472.0 だ。

ジェット機：B747，次のジェットエンジンを 4 個搭載している。

エンジンはジェットだ。名前は CF6-80C2 で，重量：4472.0 だ。

問 3 . $2 \times 12 = 24$ 点

(a) sbL[1] = new StudentBmi("本間芽衣子", 3)

(b) sbL[1].setHW(2, 1.47, 36.0)

(c) N - 1

(d) k + 1

(e) ! sbL[topInd].largerBmi(1, sbL[1])

(f) sbL[topInd]

(g) sbL[k]

(h) tmp

(i) weightL[index] = weight

(j) tmpBmi > maxBmi

(k) aveBmi /= nData

(l) maxBmi > rival.maxBmi || maxBmi == rival.maxBmi && aveBmi > rival.aveBmi