

応用プログラミング A 第 7 回演習問題 配列

下記の問題を解くプログラムを作成せよ。必須問題は授業時間中に必ず解答すること。

## 問題 1 (必須) フレンド関数

色付きの円を表す **Circle** クラスと、色名と色値を表す **CName** クラスがある。円の色が **CName** オブジェクトの色と一致しているか判定する **bool** 型関数 **isSame(CName)** をフレンド関数で実装する。動作の確認のため、**Circle** オブジェクトの色が **CName** オブジェクトの色と一致しているか判定して右図のように結果を出力するメイン関数を作成せよ。ヒントとしてプログラムリストの一部を以下に示す。あるクラスのメンバ関数を他のクラスのフレンド関数として使う方法は例 3.4 の 2 を参考にすると良い。

```
円の情報を表示
座標：(2,3)
半径：5
RGB：(255,0,0)
色1の情報を表示
Red：(255,0,0)
色2の情報を表示
Green：(0,255,0)
色3の情報を表示
Blue：(0,0,255)
この円の色はRedです
この円の色はGreenではありません
この円の色はBlueではありません
続行するには何かキーを押してください
```

```

class Circle {
    double x, y;        // 座標
    double radius;      // 半径
    int r, g, b;        // RGB値
public:
    void set_xy(double s, double t, double r) {
        x = s; y = t; radius = r;
    }
    void set_color(int u, int v, int w) {
        r = u; g = v; b = w;
    }
    void show();
    bool isSame(CName col); // CNameのメンバで
                             // はないので注意
};

class CName {
    char name[50];      // 色名
    int r, g, b;        // RGB値
public:
    CName(char* str, int u, int v, int w);
    char* get_name() { return name; }
    void show();
    ?????????????????????????????????????????? //
CircleクラスのisSame()をフレンド関数とする
};

void Circle::show() {
    cout << "座標 : (" << x << ", " << y << ")¥n";
    cout << "半径 : " << radius << "¥n";
    cout << "RGB : (" << r << ", " << g << ", " <<
b << ")¥n";
}

bool Circle::isSame(CName col) {
    if (?????????????????) return true; // 条件は
各自で考えること
    else return false;
}

```

```

CName::CName(char* str, int u, int v, int w) {
    strcpy(name, str);
    r = u; g = v; b = w;
}

void CName::show() {
    cout << name << " : (" << r << ", " << g << ", " << b << ")¥n";
}

int main() { // 以下はmain()関数のヒント
    Circle ob;
    CName co1("Red", 255, 0, 0);
    CName co2("Green", 0, 255, 0);
    CName co3("Blue", 0, 0, 255);

    ob.set_xyr(????????????);
    ob.set_color(????????????);

    cout << "円の情報を表示¥n";
    ob.show();
    cout << "色1の情報を表示¥n";
    co1.show();
    // 色2以降も同様に出力

    cout << "この円の色は" << co1.get_name();
    if (????????????????) // isSame()関数で円と
色1の色値が一致しているか判定
        cout << "です¥n";
    else
        cout << "ではありません¥n";
    // 色2、色3とも同様に比較

```

## 問題2（必須） オブジェクト配列の初期化

3次元空間の点を表す以下のような **Coord** クラスのオブジェクトの配列を作成し、その値を出力するプログラムを作成せよ。オブジェクトの数は4つで、それぞれの値は出力結果のようにになっている。このクラスの定義を追加・変更してはならない。

ヒント：この **Coord** クラスには非公開メンバ変数 *x, y, z* に値を設定する関数が用意されていない。そのため、コンストラクタを使って値を設定する必要がある。オブジェクト配列の宣言時に初期化を行うこと。コンストラクタが複数の引数を受け取るオブジェクト配列の初期化については教科書の例 4.1 の4を参考にせよ。座標の出力にはメンバ関数の **show()** を利用する。

```
class Coord {
    double x;
    double y;
    double z;
public:
    Coord(double s, double t, double u) {
        x = s; y = t; z = u;
    }
    double get_x() { return x; }
    double get_y() { return y; }
    double get_z() { return z; }
    void show();
};

void Coord::show() {
    cout << "(" << x << ", " << y << ", " << z << ")";
}
```

```
ob[0]=(1,2,3)
ob[1]=(4,5,6)
ob[2]=(7,8,9)
ob[3]=(-1,-2,-3)
続行するには何かキーを押してください
```

出力結果

## 問題3 オブジェクトの配列

問題1を拡張する。複数の **CName** オブジェクトを、配列を使って作成する。**CName** オブジェクトにホワイト、ブラック、レッド、グリーン、ブルー、シアン、マゼンタ、イエローの8つを用意して **Circle** オブジェクトの色が前述の8つの中にあるかどうかを判定するプログラムを作成せよ。テストのために、**Circle** オブジェクトの色を変えて実行してみる。なお、前述の8つの色の色値は下の実行結果を参照すること。

ヒント：**Circle** オブジェクトの色が *i* 番目の **CName** オブジェクトと一致しているか判定する文をループカウンタ *i* の **for** 文の中に記述する。一致していたら色名を出力してから **break** でループを抜ける。ループの最後まで達した場合は一致している色がなかったということなので「登録されている色にありません」と出力する。

```
円の情報を表示
座標：(2,3)
半径：5
RGB：(255,0,0)

登録してある色一覧
White：(255,255,255)
Black：(0,0,0)
Red：(255,0,0)
Green：(0,255,0)
Blue：(0,0,255)
Cyan：(0,255,255)
Magenta：(255,0,255)
Yellow：(255,255,0)

円の色はRedです
続行するには何かキーを押してください
```

色を変えてテスト

```
円の情報を表示
座標：(2,3)
半径：5
RGB：(255,150,0)

登録してある色一覧
White：(255,255,255)
Black：(0,0,0)
Red：(255,0,0)
Green：(0,255,0)
Blue：(0,0,255)
Cyan：(0,255,255)
Magenta：(255,0,255)
Yellow：(255,255,0)

円の色は登録されている色にありません
続行するには何かキーを押してください
```

#### 問題 4 オブジェクトの配列とポインタ

問題 3 をさらに拡張する。**Circle** オブジェクトを、配列を使って 3 つ作成する。また、**Circle** と **CName** の配列の要素にポインタを使ってアクセスするように変更する ([ ] を使わずに要素にアクセスする)。オブジェクトポインタをインクリメントすると配列の次のオブジェクトを指すようになることを利用して、以下の実行結果のように **Circle** オブジェクトの色を判定するプログラムを作成せよ。ポインタで配列の要素にアクセスする方法は以下のコードと例 4.2 の 1 を参考にせよ。

```
Circle ob[3];
Circle *pc;

// []を使わずにポインタでobの要素にアクセス
pc = ob;
for (i = 0; i < 3; i++) {
    cout << "円" << i + 1 << "の情報を表示\n";
    pc->show();
    pc++;
}
```

```
円1の情報を表示
座標：(2,3)
半径：5
RGB：(255,0,0)
円2の情報を表示
座標：(4,5)
半径：6.5
RGB：(0,255,0)
円3の情報を表示
座標：(1.5,1.5)
半径：2
RGB：(150,150,150)
```

```
登録してある色一覧
White：(255,255,255)
Black：(0,0,0)
Red：(255,0,0)
Green：(0,255,0)
Blue：(0,0,255)
Cyan：(0,255,255)
Magenta：(255,0,255)
Yellow：(255,255,0)
```

```
円1の色はRedです
円2の色はGreenです
円3の色は登録されている色にありません
続行するには何かキーを押してください
```

## 問題 5 (チャレンジ) 学生データの検索プログラム

学生データを表す **StudentList** クラスを使って、条件に合った学生だけ表示するプログラムを作成する。**StudentList** クラスの定義は以下のリストのようにになっている。このクラスの配列にあらかじめ 10 人分のデータを格納しておく。取得単位数が一定以上、または、一定以下の学生を検索するか、GPA が一定以上、または、一定以下の学生を検索する。2 つの条件を組み合わせても良いことにする。また、繰り返し検索できるようにする。

表のような学生データが格納されているときに、次の条件に合った学生を表示して下の実行結果のようになることを確認せよ。

- ・ 取得単位数 100 以上
- ・ GPA 1.9 以下
- ・ 取得単位数 90 以上かつ GPA 2.5 以下
- ・ 取得単位数 90 以下かつ GPA 2.5 以上

ID	氏名	取得単位数	GPA
1	指原莉乃	86	1.94
2	柏木由紀	52	1.78
3	渡辺麻友	86	2.05
4	高橋みなみ	106	2.38
5	松井珠理奈	89	2.41
6	山本彩	101	2.89
7	宮脇咲良	86	1.87
8	宮澤佐江	98	2.12
9	島崎遥香	100	2.53
10	横山由依	102	3.14

```
class StudentList {
    int id;
    char name[81];
    int credit; // 取得単位数
    float gpa; // GPA
public:
    StudentList(int n, char *str) {
        id = n;
        strcpy(name, str);
        credit = 0;
        gpa = 0;
    }
    void set_credit(int n) { credit = n; }
    void set_gpa(float n) { gpa = n; }
    int get_id() { return id; }
    char *get_name() { return name; }
    int get_credit() { return credit; }
    float get_gpa() { return gpa; }
    void show() {
        cout << "ID:" << id << "氏名:" << name
        << "取得単位数:" << credit << "GPA:" << gpa
        << "\n";
    }
};
```

```
条件に合う学生を検索します。
取得単位数の条件 0.条件なし 1.一定以上 2.一定以下
1
GPAの条件 0.条件なし 1.一定以上 2.一定以下
0
取得単位数の基準値を入力してください: 100
ID:4 氏名:高橋みなみ 取得単位数:106 GPA:2.38
ID:6 氏名:山本彩 取得単位数:101 GPA:2.89
ID:9 氏名:島崎遥香 取得単位数:100 GPA:2.53
ID:10 氏名:横山由依 取得単位数:102 GPA:3.14
検索を終了しますか? 0.はい 1.いいえ
1
条件に合う学生を検索します。
取得単位数の条件 0.条件なし 1.一定以上 2.一定以下
0
GPAの条件 0.条件なし 1.一定以上 2.一定以下
2
GPAの基準値を入力してください: 1.9
ID:2 氏名:柏木由紀 取得単位数:52 GPA:1.78
ID:7 氏名:宮脇咲良 取得単位数:86 GPA:1.87
検索を終了しますか? 0.はい 1.いいえ
1
条件に合う学生を検索します。
取得単位数の条件 0.条件なし 1.一定以上 2.一定以下
1
GPAの条件 0.条件なし 1.一定以上 2.一定以下
2
取得単位数の基準値を入力してください: 90
GPAの基準値を入力してください: 2.5
ID:4 氏名:高橋みなみ 取得単位数:106 GPA:2.38
ID:8 氏名:宮澤佐江 取得単位数:98 GPA:2.12
検索を終了しますか? 0.はい 1.いいえ
1
条件に合う学生を検索します。
取得単位数の条件 0.条件なし 1.一定以上 2.一定以下
2
GPAの条件 0.条件なし 1.一定以上 2.一定以下
1
取得単位数の基準値を入力してください: 90
GPAの基準値を入力してください: 2.5
該当者なし
検索を終了しますか? 0.はい 1.いいえ
0
続行するには何かキーを押してください . . .
```

実行結果