

## 応用プログラミング A 第9回演習問題 ポインタ・参照

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） ポインタの基本

学生のデータを管理する StudentList クラスを作成する。このクラスは氏名 (char name[50]) を非公開メンバ変数として持つ。値を設定する void set(char \*n) は strcpy を使ってメンバ変数 name に文字列をコピーする。値を出力する void show() は cout で name を出力する。この 2 つの関数は公開メンバである。このクラスのオブジェクトを 2 つ作り、ポインタを使ってそれぞれのオブジェクトにアクセスし、以下の実行結果となるようなプログラムを作成せよ。

入力

```

1人目のデータを入力します
名前を入力してください:ああああ
2人目のデータを入力します
名前を入力してください:いはいはい
1 氏名:ああああ
2 氏名:いはいはい
  
```

実行結果

```

class StudentList {
    char name[50];
public:
    void set(char *n);
    void show();
};

void StudentList::set(char *n) {
    ?????????????????;
}

void StudentList::show() {
    ?????????????????????????????????????;
}

int main() {
    StudentList s1, s2;
    StudentList *ptr;
    char nm[50];

    ptr = &s1;
    cout << "1人目のデータを入力します\n";
    cout << "名前を入力してください:";
    cin >> nm;
    ptr????????????;
    // s1に直接アクセスせず、
    // ptrを使って値を設定
    // s2についても同様
    // その後、出力
  
```

### 問題 2（必須） ポインタと参照

次のプログラムの関数 f() はポインタ仮引数を使用して値を 2 倍にしている。これを、参照仮引数を使ったコードに書き換えよ。

```

#include <iostream>
using namespace std;

void f(int *n) {
    *n = *n * 2;
}

int main() {
    int i = 10;

    cout << "i = " << i << "\n";
    f(&i);
    cout << "i = " << i << "\n";

    return 0;
}
  
```

実行結果

```

i = 10
i = 20
続行するには何かキーを押してください . . .
  
```

### 問題 3 値渡し、ポインタ渡し、参照渡し

関数にオブジェクトを渡す際に、「値渡し」、「ポインタ渡し」、「参照渡し」といった方法でオブジェクトを渡すことができる。これらの違いを確認するプログラムを作成する。以下のリストでは **Coord** クラスのオブジェクトを渡して値を設定する関数を 3 種類作成している。それらの関数を使って値を設定したのち、オブジェクトの値を表示して設定できているか確認している。このリストの不完全な部分(コメントのみの行) にコードを追加して、プログラムを完成させよ。

```
#include <iostream>
using namespace std;

class Coord {
    double x, y, z;
public:
    void set(double u, double v, double w) {
        x = u; y = v; z = w;
    }
    void show();
};

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

void setCoord1(Coord ob) {    // 値渡し
    double u, v, w;
    cout << "数字を 3 つ入力してください: ";
    cin >> u >> v >> w;
    ob.set(u, v, w);    // 実際には値を設定できない
}

void setCoord2(Coord *ob) {    // ポインタ渡し
    // 上記関数を参考に値を設定するコードを記述
}

void setCoord3(Coord &ob) {    // 参照渡し
    // 上記関数を参考に値を設定するコードを記述
}

int main() {
    Coord obj;

    obj.set(1.1, 2.3, 5.5);
    cout << "オブジェクトの初期値: ";
    obj.show();
    cout << "\n\n";

    cout << "値渡しで関数にオブジェクトを渡します\n";
    setCoord1(obj);
    cout << "オブジェクトの値: ";
    obj.show();
    cout << "\n\n";

    cout << "ポインタ渡しで関数にオブジェクトを渡します\n";
    // 上記を参考に関数呼び出しと値の表示のコードを記述

    cout << "参照渡しで関数にオブジェクトを渡します\n";
    // 上記を参考に関数呼び出しと値の表示のコードを記述

    return 0;
}
```

実行結果

オブジェクトの初期値: (1.1,2.3,5.5)

入力

値渡しで関数にオブジェクトを渡します  
数字を3つ入力してください: 1.0 2.0 3.0  
オブジェクトの値: (1.1,2.3,5.5)

ポインタ渡しで関数にオブジェクトを渡します  
数字を3つ入力してください: 1.0 2.0 3.0  
オブジェクトの値: (1,2,3)

参照渡しで関数にオブジェクトを渡します  
数字を3つ入力してください: 4.5 5.5 6.5  
オブジェクトの値: (4.5,5.5,6.5)  
続行するには何かキーを押してください . . .

値渡しではオブジェクト  
に値を設定できない

ポインタ渡しや参照渡  
しでは設定できる

#### 問題 4 new を使ったメモリの動的確保

以下の C のプログラムを、C++ で書き換えよ。なお、メモリの確保には new 演算子を使うこと。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {
    int i, n;
    char **strlist;
    char buf[256];    // 文字列は最大で255文字とする

    printf("文字列の数:");
    scanf("%d", &n);
    strlist = (char **)malloc(sizeof(char *) * n);
    if (!strlist) {
        printf("メモリが確保できません\n");
        return 1;
    }
    printf("文字列の入力\n");
    for (i = 0; i < n; i++) {
        printf("%dつ目:", i+1);
        scanf("%s", buf);
        strlist[i] = (char *)malloc(sizeof(char) * (strlen(buf)+1));
        if (!strlist[i]) {
            printf("メモリが足りません\n");
            return 1;
        }
        strcpy(strlist[i], buf);
    }
    printf("文字列の出力\n");
    for (i = 0; i < n; i++) {
        printf("%dつ目:", i+1);
        printf("%s\n", strlist[i]);
    }
    return 0;
}
```

```
文字列の数:3
文字列の入力
1つ目:情報工学科
2つ目:情報ネットワークコミュニケーション学科
3つ目:情報メディア学科
文字列の出力
1つ目:情報工学科
2つ目:情報ネットワークコミュニケーション学科
3つ目:情報メディア学科
```

実行結果

## 問題 5 (チャレンジ) 参照渡しを用いた値の交換

トランプのカードを表す PCard クラスを利用したプログラムを作成する。PCard クラスのメンバ変数はスートを表す列挙型 `suit` と数字を表す `int` 型 `number` である。メンバ関数は値を設定する `void set(enum suit st, int nm)`、スートと数字を出力する `void show()` である。メンバ変数は非公開、メンバ関数は公開とする。なお、ジョーカーについては考えない。PCard クラスのオブジェクトを 1 次元配列で 52 個作成し、スペード、ハート、クラブ、ダイヤの 1~13 のカードの値を設定する。そして、以下のアルゴリズムで配列に順番に格納されているカードをシャッフルする。

- 以下の手順を十分な回数繰り返す
  - ① 0~51 の乱数 `a` を発生させる
  - ② 0~51 の乱数 `b` を発生させる
  - ③ 配列の `a` 番目と `b` 番目の値を入れ替える

値の入れ替えには参照仮引数を使用した関数を利用すること。その後、52 枚のカードを 4 つの配列に 13 個ずつコピーする (4 人のプレイヤーにカードを配る操作を想定)。最後に 4 つの配列の値をすべて出力し、シャッフルされたカードが配られていることを確認せよ。なお、全 52 枚のカードをすべて確認するのは困難なため、テストデータとして 4 つのスートの 1~3 までのカード (全 12 枚) をシャッフルできることが確認できればよい。

```
enum suit { SPADE, HEART, CLUB, DIAMOND };

class PCard {
    enum suit suit;
    int number;
public:
    void set(enum suit st, int nm) {
        suit = st;
        number = nm;
    }
    void show();
};

void PCard::show() {
    switch(suit) {
    case SPADE:
        cout << "スペードの";
        break;
    case HEART:
        cout << "ハートの";
        break;
    case CLUB:
        cout << "クラブの";
        break;
    case DIAMOND:
        cout << "ダイヤの";
        break;
    }
    cout << number << "¥n";
}
```

スペードの1  
スペードの2  
スペードの3  
ハートの1  
ハートの2  
ハートの3  
クラブの1  
クラブの2  
クラブの3  
ダイヤの1  
ダイヤの2  
ダイヤの3

最初は  
整列している

シャッフルします

4人のプレイヤーにカードを配ります

プレイヤー1のカード  
ダイヤの1  
ハートの3  
スペードの3

プレイヤー2のカード  
ハートの2  
スペードの1  
クラブの1

プレイヤー3のカード  
クラブの3  
ダイヤの3  
ハートの1

プレイヤー4のカード  
クラブの2  
スペードの2  
ダイヤの2

シャッフルされた  
カードが配られる

続行するには何かキーを押してください . . .