2017/5/23

応用プログラミングＡ　第６回演習問題　クラスの詳細

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）　オブジェクトの関数への引き渡しと関数からの返し

三角形を表す下記の Triangle クラスのオブジェクトを扱う関数 max_tri()を作成する。この関数は Triangle 型の引数を２つ受け取り、面積の大きいオブジェクトを戻り値として返す。この関数をテストする main()関数を作り動作を確認せよ。なお、Triangle クラスの定義を追加変更してはならない。

```
class Triangle {
    double a, b, c;
public:
    void set_a(double x) { a = x; }
    void set_b(double x) { b = x; }
    void set_c(double x) { c = x; }
    double get_a() { return a; }
    double get_b() { return b; }
    double get_c() { return c; }
    double area();
    void show();
};

double Triangle::area() {
    double s;

    s = (a+b+c) / 2;
    return sqrt(s*(s-a)*(s-b)*(s-c));
}

void Triangle::show() {
    cout << "辺 a = " << a << "¥n";
    cout << "辺 b = " << b << "¥n";
    cout << "辺 c = " << c << "¥n";
    cout << "面積 = " << area() << "¥n";
}
```

```
int main() {
    Triangle obj1, obj2, obj3;

    obj1.set_a(3.0);
    obj1.set_b(4.0);
    obj1.set_c(5.0);
    obj2.set_a(5.0);
    obj2.set_b(12.0);
    obj2.set_c(13.0);

    cout << "三角形¥n";
    obj1.show();

    cout << "三角形¥n";
    obj2.show();

    obj3 = ?????????????
    cout << "大きい三角形¥n";
    obj3.show();

    return 0;
}
```

大きい三角形を代入するオブジェクト

３辺の値はこのように設定

関数を使って大きい三角形を求める

実行結果

```
三角形1
辺a = 3
辺b = 4
辺c = 5
面積 = 6
三角形2
辺a = 5
辺b = 12
辺c = 13
面積 = 30
大きい三角形
辺a = 5
辺b = 12
辺c = 13
面積 = 30
続行するには何かキーを押してください
```

ヒント　max_tri()は Triangle オブジェクトを返す関数なので型は Triangle となる。2 つの仮引数の型も Triangle なので関数宣言は以下のようになる。

```
Triangle max_tri(Triangle ob1, Triangle ob2);
```

後は、ob1 と ob2 の面積を比較して、大きいオブジェクトを return で返すという処理を記述する。

```
Triangle max_tri(Triangle ob1, Triangle ob2) {
    if (?????????????????????????)
        return ???;
    else
        return ???;
}
```

問題２（必須）　オブジェクトのアドレスの関数への引き渡し

問題１と同じ三角形を表す Triangle クラスのオブジェクトを扱う関数 mul_tri()を作成する。この関数は、Triangle クラスのオブジェクトのアドレスと double 型の引数を受け取り、そのオブジェクトの３辺の長さを受け取った double 型の値倍にする。戻り値を返さないのでこの関数の型は void 型である。下リストの main()関数に必要なコードを付け足して結果のような出力が得られることを確認せよ。

```
int main() {
    Triangle obj;
    double a;

    obj.set_a(3.0);
    obj.set_b(4.0);
    obj.set_c(5.0);
    obj.show();

    cout << "何倍にしますか？:";
    cin >> a;
    ????????????????

    obj.show();

    return 0;
}
```

実行結果

```
辺a = 3
辺b = 4
辺c = 5
面積 = 6
何倍にしますか？:1.5
辺a = 4.5
辺b = 6
辺c = 7.5
面積 = 13.5
続行するには何かキーを押してください
```

ヒント　関数宣言は以下のようになる。

```
void mul_tri(Triangle *ob, double x);
```

ポインタを使えば引数のオブジェクトの値を変更することができる。関数内ではポインタを使ってオブジェクトを操作するのでアロー演算子を使用する。教科書の例 3.2 の 3 を参考にすると良い。

```
void mul_tri(Triangle *ob, double x) {
    ????????????????????????????
    ????????????????????????????
    ????????????????????????????
}
```

問題３　関数とオブジェクト

３次元空間の点を表す右リストの Coord クラスを用いて、次のような関数を作成する。ひとつは、２つの Coord オブジェクトを引数として受け取り、その中点を表すオブジェクトを戻り値として返す関数 mid_Coord()。もうひとつは、２つの Coord オブジェクトを引数として受け取り、その２点間の距離を戻り値（double 型）として返す関数 dist_Coord()である。関数内で値を出力するものではないので注意すること。main()関数で点 A、点 B を表すオブジェクトを作成し（座標は下の結果を参照）、mid_Coord()と dist_Coord()を使って中点の座標と距離を出力せよ。

```
class Coord {
    double x;
    double y;
    double z;
public:
    void set_x(double u) { x = u; }
    void set_y(double v) { y = v; }
    void set_z(double w) { z = w; }
    double get_x() { return x; }
    double get_y() { return y; }
    double get_z() { return z; }
    void show();
};

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

Coord mid_Coord(Coord ob1, Coord ob2);

double dist_Coord(Coord ob1, Coord ob2);
```

```
点A(1,6.5,-10)
点B(-1,1.5,4)
ABの中点(0,4,-3)
AB間の距離は15です
続行するには何かキーを押してくださし
```

問題４（チャレンジ）　索敵ゲーム

次のようなゲームを考える。4×4 マスのマップにプレイヤーユニット、敵ユニット、ターゲットユニットがいる。毎フェイズ、ユニットは 1 マス動くことができる。ターゲットと同じマスにプレイヤーが入ると勝ち、敵と同じマスにプレイヤーが入ると負けである。プレイヤーには敵、および、ターゲットの位置は直接分からないとする。そのかわり、近くにユニットがいるかいないかは分かるものとする。ユニットを表すクラスを作成し、プレイヤー、敵、ターゲットをオブジェクトとして扱う。このクラスは座標を表すメンバ変数を持つ。ユニット間の距離は、2 つのオブジェクトを引数として受け取り距離を戻り値として返す関数を作って判定するとよい。このようなゲームを作成せよ。細かな仕様については適当に定めてよい。コンソール入出力やメッセージに関しても自由に設定してよい（図はあくまで一例である)。

```
あなたは(1,3)にいます
ターゲットは近くにいません
敵1は近くにいます
敵2は近くにいません
どちらに移動しますか？（下-2、左-4、右-6、上-8、移動しない-5）:8
あなたは(1,2)にいます
ターゲットは近くにいます
敵1は近くにいます
敵2は近くにいます
どちらに移動しますか？（下-2、左-4、右-6、上-8、移動しない-5）:6
あなたは(2,2)にいます
ターゲットは近くにいます
敵1は近くにいます
敵2は近くにいません
どちらに移動しますか？（下-2、左-4、右-6、上-8、移動しない-5）:6
おめでとう！
ターゲットを捕まえました
続行するには何かキーを押してください . . .
```

テキストメッセージのみで作成した例

```
+－+－+－+－+
|　|　|　|　|
+－+－+－+－+
|　|　|　|　|
+－+－+－+－+
|　|　|　|　|
+－+－+－+－+
|　|　|　|Ｐ|
+－+－+－+－+
ターゲットは近くにいません
敵1は近くにいません
敵2は近くにいません
　　８
　　↑
４←５→６
　　↓
　　２
どちらに移動しますか？
```

```
+－+－+－+－+
|　|　|E2|　|
+－+－+－+－+
|　|　|　|　|
+－+－+－+－+
|Ｐ|　|　|E1|
+－+－+－+－+
|　|　|　|　|
+－+－+－+－+
おめでとう！
ターゲットを捕まえました
続行するには何かキーを押してください
```

グラフィカルな出力を用いた例