

応用プログラミング A 第 5 回演習問題 クラスの概要 (2)

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) 仮引数を受け取るコンストラクタ

第 3 回演習で作成した三角形を表すクラス `Triangle` に、仮引数として 3 辺の長さを受け取るコンストラクタを追加する。また、3 辺の長さから面積を計算して戻り値として返すメンバ関数 `area()` を追加する。このクラスをテストするために、以下の例のような `main()` 関数で三角形の 3 辺の長さと面積を出力せよ。3 辺の長さはコンソールから入力せず、プログラム中で直接設定する。

```
class Triangle {
    double a, b, c;
public:
    Triangle(double u, double v, double w); ←
    void set_a(double x);
    void set_b(double x);
    void set_c(double x);
    double get_a();
    double get_b();
    double get_c();
    double area(); ←
};

Triangle::Triangle(double u, double v, double w) {
    ???????
}

double Triangle::area() {
    return ???;
}
```

第 3 回演習 問 2 の
クラスに追加

```
int main() {
    Triangle obj(3.0, 4.0, 5.0);

    cout << "辺a = " << ??????????;
    cout << "辺b = " << ??????????;
    cout << "辺c = " << ??????????;
    cout << "面積 = " << ??????????;

    return 0;
}
```

```
辺a = 3
辺b = 4
辺c = 5
面積 = 6
続行するには何かキーを押してください . . .
```

問題 2 (必須) 継承

教科書の練習問題 2.3 をやってみること。そのうえで、以下の `area_cl2` クラスを基本クラスとして継承する `rectangle` クラスと `isosceles` クラスを同様に作成せよ。

ヒント 練習問題 2.3 と異なり、`height` と `width` は非公開メンバなので「`height=h;`」のように直接設定することができない。`rectangle` と `isosceles` クラスのコンストラクタ関数内で `set_h()` と `set_w()` を使って設定する。値の呼び出しには `get_h()` と `get_w()` を使う。練習問題 2.3 の解答は付録に載っているので、そのコードを利用すると良い。

```
class area_cl2 {
    double height;
    double width;
public:
    void set_h(double h); // height に値を設定する関数
    void set_w(double w); // width に値を設定する関数
    double get_h();       // height を返す関数
    double get_w();       // width を返す関数
};
```

```
長方形 高さ:10 幅:5 面積:50
三角形 高さ:4 幅:6 面積:12
続行するには何かキーを押してください .
```

```

int main() {
    rectangle b(10.0, 5.0);
    isosceles i(4.0, 6.0);

    cout << "長方形 高さ:" << ?????????? << " 幅:" << ?????????? << " 面積:" << ?????????? << "¥n";
    cout << "三角形 高さ:" << ?????????? << " 幅:" << ?????????? << " 面積:" << ?????????? << "¥n";

    return 0;
}

```

← get_h()を使う
← get_w()を使う

問題3 オブジェクトポインタ

問題 2 のオブジェクトをポインタで操作するように変更する。
main関数を右のように変更し、ポインタ pb および pi を使って値を出力せよ (b や i をつかって出力しない)。

```

長方形 高さ:10 幅:5 面積:50
三角形 高さ:4 幅:6 面積:12
続行するには何かキーを押してください . . .

```

```

int main() {
    rectangle b(10.0, 5.0);
    isosceles i(4.0, 6.0);
    rectangle *pb;
    isosceles *pi;

    pb = &b;
    pi = &i;

    cout << ??????????????????
    cout << ??????????????????

    return 0;
}

```

問題4 共用体

教科書の例 2.5 の 2 を参考に、コンソールから入力された int 型の値のバイナリビットパターンをバイト単位で出力するプログラムを作成せよ。

```

整数を入力してください:123456
バイト単位のビットパターン 3: 00000000
バイト単位のビットパターン 2: 00000001
バイト単位のビットパターン 1: 11100010
バイト単位のビットパターン 0: 01000000
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) すごろくシミュレータ

すごろくで 20 マス先にあるゴールにたどり着くまでに平均何回サイコロを振るかシミュレーションによって求めるプログラムを作成せよ。ゴールにたどり着くのはちょうどどの数字が出たときだけで、ゴールまでの数を超える値が出た場合は折り返すというルールとする。サイコロの目を乱数で発生させ、ゴールにたどり着くまでにサイコロを振った回数を数える。十分な回数 (少なくとも 1000 回以上) 試行して、その平均回数を出力せよ。

答えを求めることができればコードは自由に書いてよい (C++らしいコードでなくても、例えばクラスを使わないコードでも、答えが求まるのであれば問題ない)。

```

20マス先のゴールにたどり着くのに平均10.427回サイコロを振りました
続行するには何かキーを押してください . . .

```

