

課題11

Q01: Point クラス と Rectangle クラスの設計

(プログラム名: **ex11_1.cpp**)

C++ を用いて、点を表すクラス Point と、長方形を表すクラス Rectangle を定義し、以下のメンバや操作を実装しなさい。

1. クラス Point

- データメンバ
 - `float x;`
 - `float y;`
 - コンストラクタ
 - `Point();`
 - 引数なしコンストラクタ。点を `(0.0f, 0.0f)` に初期化する。
 - `Point(float x, float y);`
 - 引数つきコンストラクタ。与えられた座標 `x, y` で点オブジェクトを初期化する。
-

2. クラス Rectangle

- プライベートデータメンバ
 - `Point pt1;` // 左下の頂点
 - `Point pt2;` // 右上の頂点
- コンストラクタ
 - `Rectangle();`
 - 引数なしコンストラクタ。デフォルトの長方形を生成する:
`pt1 = (0.0f, 0.0f)`、`pt2 = (1.0f, 1.0f)` とする。
 - `Rectangle(const Point& p1, const Point& p2);`
 - 引数付きコンストラクタ: **p1, p2** の位置関係に関わらず、**pt1=左下の頂点, pt2=右上の頂点**になるよう正規化。
- メンバ関数
 1. `float area() const;`
 - 長方形の面積を計算し、その値を返す。
 2. `float perimeter() const;`
 - 長方形の周の長さ(周長)を計算し、その値を返す。
- 演算子のオーバーロード

1. `friend std::ostream& operator<<(std::ostream& os, const Rectangle& r);`
 - 挿入演算子 `<<`:
長方形 `r` の情報を、
`[(pt1.x, pt1.y), (pt2.x, pt2.y)], area = 長方形の面積`
のような形式で出力ストリーム `os` に書き出す。
2. `friend std::istream& operator>>(std::istream& is, Rectangle& r);`
 - 抽出演算子 `>>`: 入力形式: `x1 y1 x2 y2`
入力ストリーム `is` から 2 点の座標を読み込み、左下と右上の座標を計算し、`r.pt1, r.pt2` を設定する。

【プログラム】

今回の課題では、コードを複数のファイルに分割する必要はありません。

下に示したサンプルコードを参考にして、同じファイル内に必要な処理を追記し、プログラムを完成させてください。

```
C/C++
#include <iostream>

class Point
{
    // ↓ ここを補完
}

class Rectangle
{
private:
    // 常に pt1: 左下の頂点, pt2: 右上の頂点 になるように保持する
    Point pt1; // 左下
    Point pt2; // 右上
public:
    // ↓ ここを補完

    // デフォルトコンストラクタ: pt1(0,0), pt2(1,1)
    Rectangle()
    {
    }

    // 引数付きコンストラクタ
    // 注意: p1, p2 の位置関係に関わらず、pt1=左下, pt2=右上 になるよう正規化
    Rectangle(const Point &p1, const Point &p2)
```

```

{
}

// 面積
float area() const
{
}

// 周長
float perimeter() const
{
}

// 挿入演算子 <<
friend std::ostream& operator<<(std::ostream &os, const Rectangle &r)
{
}

// 抽出演算子 >>
// 入力形式 : x1 y1 x2 y2
friend std::istream& operator>>(std::istream &is, Rectangle &r)
{
}
}

// 動作確認用 main、この部分は変更しないでください
int main()
{
    using namespace std;
    Rectangle r1;
    Rectangle r2(Point(3.0f, 2.0f), Point(0.6f, 0.5f));

    Rectangle r3;
    cout << "矩形の座標を、x1 y1 x2 y2 の形式で入力してください：" ;
    cin >> r3;

    cout << "r1 = " << r1 << "\n";
    cout << "r2 = " << r2 << "\n";
    cout << "r3 = " << r3 << "\n";
}

```

【実行例1】

入力

None

矩形の座標を、x1 y1 x2 y2 の形式で入力してください : 1 2 3 4

出力

None

```
r1 = [(0, 0), (1, 1)], area = 1
r2 = [(0.6, 0.5), (3, 2)], area = 3.6
r3 = [(1, 2), (3, 4)], area = 4
```

【実行例2】

入力

None

矩形の座標を、x1 y1 x2 y2 の形式で入力してください : 3 4 1 2

出力

None

```
r1 = [(0, 0), (1, 1)], area = 1
r2 = [(0.6, 0.5), (3, 2)], area = 3.6
r3 = [(1, 2), (3, 4)], area = 4
```

【実行例3】

入力

None

矩形の座標を、x1 y1 x2 y2 の形式で入力してください : 3 1 3 2

出力

None

```
r1 = [(0, 0), (1, 1)], area = 1
r2 = [(0.6, 0.5), (3, 2)], area = 3.6
r3 = [(3, 1), (3, 2)], area = 0
```

