

第2回 C++プログラミング実験II

出題日 2021年4月16日(金) 15:00

第一提出期限 2021年4月16日(金) 16:40

第二提出期限 2021年4月22日(木) 23:59

実験課題2

以下のプログラムは、二次元座標における座標点に対するいくつかの処理を行っている。以下のプログラムを実行して、実行例のようになるために必要な二次元座標の座標点を表す Point クラスを作成せよ。Point クラスは、座標点の名前 (string), x, y 座標 (double) をメンバーデータとして持つ。

<pre>#include <iostream> #include "point.hpp" // 作成するPointクラスの定義, 関数宣言を記述する int main(){ // (1) 以下を参考にコンストラクタを作成せよ Point a{"A", 1.0, 2.0}, b{"B", 3.4, 1.6}; Point c{}, d{1.1, 2.2}, e{3.3}, m{"M"}; std::cout<<"¥n(1)--- Constructer OK¥n"; std::cout<<"¥n(2)--- Point output¥n"; //(2) 出力表示 コンストラクタの確認 std::cout<<"a: "<<a<<"¥n"; std::cout<<"b: "<<b<<"¥n"; std::cout<<"c: "<<c<<"¥n"; std::cout<<"d: "<<d<<"¥n"; std::cout<<"e: "<<e<<"¥n"; std::cout<<"m: "<<m<<"¥n"; std::cout<<"¥n(3)--- Point input¥n"; //(3)入力 std::cout<<"before: Point c:"<<c<<"¥n"; std::cout << "Input Point c: name, x,y: "; std::cin >> c; // 入力 std::cout<<"after: Point c:"<<c<<"¥n"; std::cout<<"¥n(4)--- Point x,y setting¥n"; //(4)座標の代入設定 b<=c; // b~cの座標の代入 <=は、大なりの比較ではなく、座標代入の演算子として用いている std::cout<<b<<"¥n"; std::cout<<"¥n(5)--- Point addition¥n"; //(5)座標の加算 m <= (a+b); // Pointの座標の加算, 座標の代入 <= std::cout<< a << " + " << b << " = "<< m << "¥n"; std::cout<<"¥n(6)--- Point scalar division¥n"; //(6)座標のスカラーによる除算 m <= (a+b)/2; // 2点 a, bの midpoint mの計算(加算, 除算), 座標の代入 <= std::cout<<"Middle point: "<<m<<"¥n"; std::cout<<"¥n(7)--- Point scalar multiplication¥n"; //(7) 座標のスカラー積 double k=3.4; d = a*k; std::cout<<"d=a*k "<< d<<"¥n"; std::cout<<"¥n(8)--- Scalar Point multiplication¥n"; // (8) 座標のスカラー積2 e = k*a; // (8)はできる人だけでいい std::cout<<"e=k*a "<< e<<"¥n"; }</pre>	prac02-main.cpp
--	-----------------

CoursePower から、`prac02-main.cpp` をダウンロードし、以下の実行に必要な `Point` クラスの定義ファイル `point.hpp` と必要な関数の実装プログラム、`point.cpp` を作成せよ。

まずは、以下の実行例をみて、クラス `Point` には、どのようなメンバー関数や一般関数が必要なのか、考えること。

実際に作成する関数が決まったら、順に作成していく。まず、`main` プログラムの一部をコメントにして、(1) から (7) までは、順に、必要な関数を `point.hpp` や `point.cpp` に定義して加えていくこと。すなわち、最初は、コンストラクタを作成して、`main` の (1) の表示が無事出るか確認せよ。この時、(2) から `main` の最後の部分までコメントアウトしておく。次に (2) の部分の出力関数を作成して、(2) までは正常に動くか確認せよ。この時、(3) 以降はコメントアウトしておく。次に (3) の入力のための関数を作成した場合には、(4) 以降の部分コメントアウトして実行せよ。以下、同様。

以下の実行例を見て、(1) — (8) のそれぞれの部分では、どのような動作が必要か、どう値を設定しているか、プログラムを実際に作成する前に、考えること。

実行結果 (下線部は、入力)

```
% ./prac02
(1)--- Constructor OK
(2)--- Point output
a: (A: 1, 2 )
b: (B: 3.4, 1.6 )
c: (noname: 0, 0 )
d: (noname: 1.1, 2.2 )
e: (noname: 3.3, 0 )
m: (M: 0, 0 )

(3)--- Point input
before: Point c:(noname: 0, 0 )
Input Point c: name, x,y: CCC 2.8 5.6
after: Point c:(CCC: 2.8, 5.6 )

(4)--- Point x,y setting
(B: 2.8, 5.6 )

(5)--- Point addition
(A: 1, 2 ) + (B: 2.8, 5.6 ) = (M: 3.8, 7.6 )

(6)--- Point scalar division
Middle point: (M: 1.9, 3.8 )

(7)--- Point scalar multiplication
d=a*k (A: 3.4, 6.8 )

(8)--- Scalar Point multiplication
e=k*a (A: 3.4, 6.8 )
```