

第2回 C++ プログラミング実験 II

出題日：2022 年 4 月 18 日（月）

課題 1（提出期限：4 月 21 日（木） 24:00）

前回（第 1 回）の実験課題で作成した分数を扱う Frac クラスを、コンストラクタを使って書き換えてみよう。

まず、下に示すソースコード ex02-1-main.cpp と frac2.hpp を作成せよ。

次に、frac2.hpp の Frac クラスにコンストラクタを作成して、プログラムを完成させよ。Frac クラスのコンストラクタは、引数の値をメンバ変数にセットする（引数なしの場合は $\frac{0}{1}$ になるようセットする）ものとする。完成したプログラムをコンパイルし、実行例と同じ結果になるか確認せよ。

注意事項

ex02-1-main.cpp は下のソースコードから変更せず、frac2.hpp だけを書き換えて完成させること。

また、コンストラクタでセットされる値は既約分数にすること。

ソースコード 1 ex02-1-main.cpp

```
1 // 分数クラスの使用
2 #include <iostream>
3 #include "frac2.hpp"
4
5 void print(const Frac& f)
6 {
7     std::cout << f.str() << " = " << f.value() << "\n";
8 }
9
10 Frac add(const Frac& a, const Frac& b)
11 {
12     Frac c;
13     c.set(a.getn()*b.getd()+b.getn()*a.getd(), a.getd()*b.getd());
14     return c;
15 }
16
17 int main()// 分数クラスの使用
18 {
19     Frac a{1, 3};
20     Frac b{8, 10};
21     Frac c;
22     print(a);           // 1/3 = 0.333333
23     print(b);           // 4/5 = 0.8
24     print(c);           // 0/1 = 0
25     c = add(a, b);
26     print(c);           // 17/15 = 1.13333
27     print(add(a, a));   // 2/3 = 0.666667
28 }
```

```
1 // 分数クラス
2 #include <sstream>
3 #include <numeric>
4
5 class Frac {    // Fraction(分数)
6     int nume, deno; // nume:numerator(分子), deno:denominator(分母)
7 public:
8     void set(int n, int d) {
9         if (d < 0) {
10             n = -n;
11             d = -d;
12         }
13         int gcd {std::gcd(n, d)};
14         nume = n/gcd;
15         deno = d/gcd;
16     }
17     double value() const { return static_cast<double>(nume)/deno; }
18     std::string str() const {
19         std::ostringstream o;
20         o << nume << "/" << deno;
21         return o.str();
22     }
23     int getn() const { return nume; }
24     int getd() const { return deno; }
25 };
```

実行例 1

```
./ex02-1-main
1/3 = 0.333333
4/5 = 0.8
0/1 = 0
17/15 = 1.13333
2/3 = 0.666667
```

課題 2 (提出期限 4 月 21 日 (木) 24:00)

複素数の足し算を行うプログラムを考える。CoursePower から complex1.hpp と ex02-2-main.cpp をダウンロードし、complex1.hpp と ex02-2-main.cpp の空欄を埋めることで、プログラムを完成させなさい。

注意事項

複素数は $a + bi$ のような形式で表示させるが、実行例のように、虚部が負の値のときは $a - bi$ と表示されたり、実部や虚部が 0 のときには $(0 + 2i$ ではなく) $2i$ のように表示されたりするように工夫しなさい。

ソースコード 1 complex1.hpp

```
1 // 複素数を扱うクラス Complex
2
3 class Complex{
4     // 実部と虚部をそれぞれ表す整数 re と im
5     int re{}, im{};
6 public:
7     // コンストラクタの作成
8     Complex(int r=0, int i=0){re=r; im=i;}
9
10    // ゲッター
11    int getr() const{return re;}
12    int geti() const{return im;}
13
14    /*
15     (1) 演算子+をメンバ関数で多重定義せよ
16    */
17
18    // 演算子>>のフレンド関数での多重定義(実部と虚部を順番に入力)
19    friend std::istream& operator>>(std::istream& in, Complex& c){
20        return in >> c.re >> c.im;
21    }
22 };
```

ソースコード 2 ex02-2-main.cpp

```

1 // 複素数クラスを使用
2 #include <iostream>
3 #include "complex1.hpp"
4
5
6 /*
7 (2) 演算子<<を一般関数で多重定義せよ(出力が実行例と同じになるように工夫せよ)
8 */
9
10
11 int main()
12 {
13     Complex A{1,3}, B{1}, C{1,-5}, D{}, E{};
14
15     // 複素数Dの値を入力
16     std::cin >> D;
17
18     // 様々な計算結果を出力
19     std::cout << "A+B=" << A+B << ", A+C=" << A+C
20         << ", A+D=" << A+D << ", E=" << E << "\n";
21
22     return 0;
23 }

```

実行例

```

$ g++ -std=c++17 ex02-2-main.cpp -o ex02-2
$ ./ex02-2
-1 -1 (← 自分で入力)
A+B=2+3i, A+C=2-2i, A+D=2i, E=0

```

課題3 (提出期限 4月21日(木) 24:00)

課題2の結果を利用して、さらに複素数の絶対値(modulus)を求めて比較する。CoursePowerからcomplex2.hppとex02-3-main.cppをダウンロードし、実行例を参考にcomplex2.hppとex02-3-main.cppの空欄を埋めることで、プログラムを完成させなさい。

ソースコード1 complex2.hpp

```
1  #include <cmath> //平方根の計算にstd::sqrt()を使う
2
3  class Complex{
4      int re{}, im{};
5  public:
6      //コンストラクタの作成
7      Complex(int r=0, int i=0){re=r; im=i;}
8
9      //ゲッター
10     int getr() const{return re;}
11     int geti() const{return im;}
12
13     /*
14     (1) 絶対値を計算する関数を作ろう
15     */
16
17     //演算子>>のフレンド関数での多重定義
18     friend std::istream& operator>>(std::istream& in, Complex& c){
19         return in >> c.re >> c.im;
20     }
21
22     /*
23     (2) 演算子<をメンバ関数で多重定義せよ
24     */
25
26     };
```

ソースコード 2 ex02-3-main.cpp

```

1  #include <iostream>
2  #include "complex2.hpp"
3
4
5  /*
6  (0) 課題2で作成したものと同様に演算子<<を一般関数で多重定義せよ
7  */
8
9  int main()
10 {
11     Complex A{1,3}, B{1}, C{1,-5}, D{}, E{};
12
13     std::cin >> D;
14
15     std::cout << "A=" << A << ",D=" << D << "\n";
16     std::cout << "The modulus of A is " << A.getModulus() << "\n";
17     std::cout << "The modulus of D is " << D.getModulus() << "\n";
18
19     // 絶対値を比較する
20     if(A < D){
21         std::cout<<"The modulus of A is smaller than the modulus of D."<<"\n";
22     }
23
24
25     return 0;
26 }

```

実行例 1

```

$ g++ -std=c++17 ex02-3-main.cpp -o ex02-3
$ ./ex02-3
3 -18 (← 自分で入力)
A=1+3i, D=3-18i
The modulus of A is 3.16228
The modulus of D is 18.2483
The modulus of A is smaller than the modulus of D.

```

実行例 2

```

$ g++ -std=c++17 ex02-3-main.cpp -o ex02-3
$ ./ex02-3
1 -1 (← 自分で入力)
A=1+3i, D=1-i
The modulus of A is 3.16228
The modulus of D is 1.41421

```