

C++プログラミングII

第1回 演習

演習 1

- ▶ TimeData5 クラスを使った例を入力し、コンパイルと実行を行ってみよ。
- ▶ ファイルが3個あることに注意せよ。
- ▶ コンパイルと実行方法

```
$ g++ -std=c++17 main.cpp timedata-impl.cpp  
$ ./a.out  
3:50  
5:05
```

演習 2

- ▶ `timedata.hpp` の `TimeData5` を利用して、標準入力から読み込んだ時間情報を合計して出力するプログラムを作成せよ。ファイル名は `main2.cpp` とする。
- ▶ 以下の内容のファイル `elapse.txt` を想定する。

ヘッダファイル : `timedata.hpp`

`elapse.txt`

```
#include <string>
class TimeData5 {
    int min{};
    int sec{};
public:
    void add(int m, int s);
    std::string str() const;
};
```

```
10:10
3:05
4:20
15:28
18:19
```

- ▶ ヒント: `cin >> m >> c >> s` の形式が利用できる

演習 3

TimeData5 を以下の条件で作り返えて TimeData6 とせよ。

- ▶ timedata6.hpp に以下のクラス定義とメンバ関数の本体を入れる。
 - ▶ データメンバを `int sec;` のみ。
 - ▶ メンバ関数の引数や戻り値は変更しない。
- ▶ main2.cpp から main3.cpp を作り以下を修正する。
 - ▶ ヘッダファイルを `timerdata6.hpp` とする
 - ▶ `main()` 関数内の変数 `total` の型名を `TimeData6`

ソースコード 2: timedata6.hpp

```
class TimeData6 {  
    int sec{};  
public:  
    void add(int m, int s);  
    std::string str() const;  
};
```

演習 4

分数を扱う `Frac` クラスとそのプログラムを作成せよ。
以下を条件とする。

- ▶ データメンバは整数で `nume` と `deno` とする
 - ▶ 分子: `numerator`, 分母: `denominator` の省略である。
- ▶ `Frac` 変数の初期値は $\frac{0}{1}$ とする
- ▶ `Frac` 変数 `a` は `a.set(1, 2)` で $\frac{1}{2}$ に設定できる
 - ▶ ただし分母分子は既約分数の数にする
 - ▶ `<numeric>` の `std::gcd(int, int)` を用いてよい
 - ▶ マイナスの場合は分子に符号を与える
- ▶ `a.value()` で分数の値を実数で取り出せる
 - ▶ `int` 変数 `n` を `double` に変換するには `static_cast<double>(n)` とする
- ▶ `a.str()` で $\frac{1}{2}$ のような文字列が取り出せる
- ▶ 次のページのプログラムが実行できるようにする

演習 3 (続き)

```
// インクルード
// Frac クラスの定義

void print(const Frac& f)
{
    std::cout << f.str() << " = " << f.value() << "\n";
}

int main()
{
    Frac a;
    print(a);           // 0/1 = 0
    a.set(2, 4);
    print(a);           // 1/2 = 0.5
    a.set(32, -128);
    print(a);           // -1/4 = -0.25
    a.set(-3, -9);
    print(a);           // 1/3 = 0.333333
}
```