



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 10
по курсу «Языки и методы программирования»
«Реализация итераторов на языке C++»
Вариант 4

Студент группы ИУ9-21Б Шиятов Н.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Требуется составить контейнерный класс и итератор для перебора содержимого объектов этого класса.

Последовательность простых дробей с однонаправленным итератором по суммам соседних дробей. Обращение к элементам последовательности должно осуществляться с помощью перегруженной операции «[]». Изменение суммы двух соседних дробей посредством итератора должно отражаться на величине первой дроби.

2 Результаты

Исходный код программы представлен в листингах ??– ??.

Результат запуска представлен на рисунке ??.

Листинг 1 — Код класса Fraction

```
1 #include <iostream>
2 #include <utility>
3 #include <vector>
4
5 using namespace std;
6
7 class Fraction {
8 private:
9     int numerator, denominator;
10
11     void simplify() {
12         int t = GCD(numerator, denominator);
13         this->numerator = numerator / t;
14         this->denominator = denominator / t;
15     };
16 public:
17     Fraction() : numerator(1), denominator(1) {};
18
19     Fraction(int numerator, int denominator) : numerator(numerator),
20     denominator(denominator) {
21         simplify();
22     }
23
24     static int GCD(int a, int b) {
25         return b ? GCD(b, a % b) : a;
26     }
27
28     static int LCM(int a, int b) {
29         return a / GCD(a, b) * b;
30     }
31
32     friend Fraction sum(Fraction a, Fraction b) {
33         int numerator1 = a.numerator, denominator1 = a.denominator;
34         int numerator2 = b.numerator, denominator2 = b.denominator;
35         int lcm = LCM(denominator1, denominator2);
36         numerator1 *= lcm / denominator1;
37         numerator2 *= lcm / denominator2;
38         int x = numerator1 + numerator2;
39         return {x, lcm};
40     }
41
42     friend Fraction operator+(const Fraction &a, const Fraction &b) {
43         return sum(a, b);
44     }
45
46     friend ostream &operator<<(ostream &output, const Fraction &frac) {
47         output << frac.numerator << "/" << frac.denominator;
48         return output;
49     }
50
51     friend istream &operator>>(istream &in, Fraction &frac) {
52         in >> frac.numerator >> frac.denominator;
53         frac.simplify();
54         return in;
55     }
56 };
57
```

Листинг 2 — Код класса FractionSequence

```

1 class FractionSequence {
2 private:
3     vector<Fraction> sequence;
4
5     class Iterator {
6         vector<Fraction>::iterator cur;
7     public:
8         Iterator(vector<Fraction>::iterator first) : cur(first) {}
9
10        Iterator operator++(int) {
11            Iterator temp = *this;
12            cur++;
13            return temp;
14        }
15
16        bool operator!=(const Iterator &it) {
17            return cur != it.cur;
18        }
19
20        bool operator==(const Iterator &it) {
21            return cur == it.cur;
22        }
23
24        Iterator operator+=(Fraction x) {
25            *cur = *cur + x;
26            return *this;
27        }
28
29        Fraction operator*() {
30            return *cur + *(cur + 1);
31        }
32    };
33
34 public:
35     explicit FractionSequence(vector<Fraction> s) : sequence(std::move(s)) {}
36
37     void add(int numerator, int denominator) {
38         sequence.emplace_back(numerator, denominator);
39     }
40
41     Fraction &operator[](int index) {
42         return sequence[index];
43     }
44
45     Iterator begin() {
46         return {sequence.begin()};
47     }
48
49     Iterator end() {
50         return {sequence.end() - 1};
51     }
52 };

```

Листинг 3 — Код функции main

```
1 #include "FractionSequence.cpp"
2 #include <iostream>
3
4 using namespace std;
5
6 int main() {
7     FractionSequence sequence({
8         Fraction(1, 2),
9         Fraction(1, 3),
10    });
11    sequence.add(3, 4);
12
13    for (int i = 0; i < 3; i++){
14        cout << sequence[i] << " ";
15    }
16    cout << endl;
17
18
19    for (auto it = sequence.begin(); it != sequence.end(); it++) {
20        cout << *it << " ";
21    }
22    cout << endl;
23
24
25    for (auto it = sequence.begin(); it != sequence.end(); it++){
26        it += {1, 3};
27        cout << *it << " ";
28    }
29    cout << endl;
30
31
32    for (int i = 0; i < 3; i++) {
33        cout << sequence[i] << " ";
34    }
35
36    return 0;
37 }
```

```
/media/hdd/Education/Univer/Learning/2_s  
e-build-debug/FractionIterator  
1/2  1/3  3/4  
5/6  13/12  
7/6  17/12  
5/6  2/3  3/4  
Process finished with exit code 0
```

Рис. 1 — Результат работы