

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 №7**  
Реализация классов на языке C++  
**Вариант №20**

Выполнил  
студент группы ИУ9-21Б  
Лисов Алексей

Москва, 2023

## 1 Условие

Приобретение навыков использования библиотеки `swing`. В варианте номер 20 необходимо было реализовать Тетраэдр Рубика, на каждом ребре которого размещается  $n$  элементов, в изометрической проекции. По выбору пользователя тетраэдр может быть или не быть раскрашен случайными цветами. Условие задачи, исходный код и пример работы программы необходимо прислать в формате  $\text{\LaTeX}$ .

## 2 Код решения

Файл `main.cpp`

```
#include <iostream>
#include <vector>
#include <algorithm>
#include "Fract.h"

using namespace std;

int main() {
    Fract a(2, 5);
    for (auto i : a.getFiniteContinuedFraction(a.n, a.d)) {
        cout << i << " ";
    }
    cout << "\n";

    int& b = a.get(0);
    b = 12;

    return 0;
}
```

Файл `Fract.h`

```
#pragma once

#include <vector>
#include <iostream>
```

```

using namespace std;

class Fract {
public:
    vector<int> f;
    int n;
    int d;
    Fract(int n_, int d_) {
        n = n_;
        d = d_;
    };

    vector<int> getFiniteContinuedFraction(int n, int d) {
        std::vector<int> continuedFraction;

        int r;
        while (d != 0) {
            r = n % d;
            continuedFraction.push_back((n - r) / d);
            n = d;
            d = r;
        }

        f = continuedFraction;
        return continuedFraction;
    }

    void getRationalNumberFromContinuedFraction() {
        int n = f[f.size() - 1];
        int d = 1;

        int r;
        int k = f.size();
        while (k > 1) {
            r = d;
            d = n;
            k--;
            n = f[k - 1] * n + r;
        }

        cout << n << "/" << d << "\n";
    }
};

```

```

    }

    void add(int d_) {
        f.push_back(d_);
    }

    int& get(int i) {
        return f[i];
    }
};

```

### 3 Скриншоты

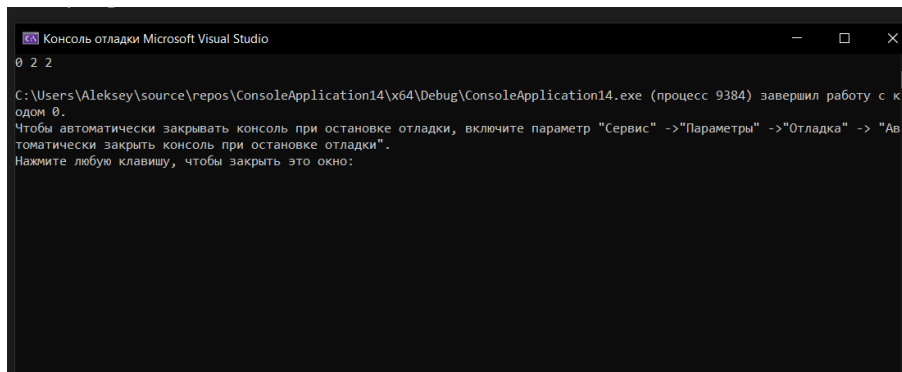


Рис. 1: Вывод программы

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include "Fract.h"
5
6  using namespace std;
7
8  int main() {
9      Fract a(2, 5);
10     for (auto i : a.getFiniteContinuedFraction(a.n, a.d)) {
11         cout << i << " ";
12     }
13     cout << "\n";
14
15     int& b = a.get(0);
16     b = 12;
17
18     return 0;
19 }
20

```

Рис. 2: Вывод программы

```

1  #pragma once
2
3  #include <vector>
4  #include <iostream>
5
6  using namespace std;
7
8  class Fract {
9  public:
10     vector<int> f;
11     int n;
12     int d;
13     Fract(int n_, int d_) {
14         n = n_;
15         d = d_;
16     };
17
18     vector<int> getFiniteContinuedFraction(int n, int d) {
19         std::vector<int> continuedFraction;
20
21         int r;
22
23     };
24 };
25

```

Рис. 3: Вывод программы