

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«Сибирский государственный университет науки и технологий  
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Языки программирования

Лабораторная работа №3 Шаблонные классы

---

Руководитель

\_\_\_\_\_  
подпись, дата

А.В. Проскурин

\_\_\_\_\_  
инициалы, фамилия

Обучающийся БПИ23-02, 23151451

\_\_\_\_\_  
номер группы, зачетной книжки

\_\_\_\_\_  
подпись, дата

С.А. Черкашин

\_\_\_\_\_  
инициалы, фамилия

Красноярск 2024 г.

## **ЦЕЛЬ РАБОТЫ**

*Получение практических навыков разработки и отладки программ, с использованием шаблонов функций и классов.*

## ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомиться с общей постановкой задачи.
2. Ознакомиться с вариантом задания – соответствует вашему номеру в списке группы (при нехватке заданий вариант задания вычисляется как *номер\_в\_списке\_группы - количество\_заданий*).
3. Разработать классы согласно варианту задания.
4. Написать и отладить программу на подготовленных наборах тестовых данных.
5. Подготовить отчет по лабораторной работе. Отчет должен включать в себя:
  - a. титульный лист;
  - b. цель лабораторной работы;
  - c. постановку задачи;
  - d. текст программы с комментариями;
  - e. демонстрацию работы программы (*Снимки экрана при выполнении действий программы с описанием*).
  - f. краткие ответы на контрольные вопросы;
  - g. выводы по лабораторной работе.
6. Защитить лабораторную работу перед преподавателем.

## ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать программу, в которой будет реализован шаблонный класс, выполняющий обработку массива данных согласно варианту задания.

В зависимости от оценки, на которую вы претендуете, необходимо выполнить

следующие задания (Для каждой следующей оценки нужно выполнить **ВСЕ** предыдущие задания, если обратное не указано явно):

Оценка	Общее задание
Удовлетворительно	1. Определить шаблонный класс для обработки массива данных согласно заданию.
	2. Написать конструкторы: — по умолчанию; — копирования (получающий параметр «ссылка на класс»). Реализовать деструктор класса.
	3. Написать методы ввода-вывода данных.
	4. Реализовать метод, выполняющий одно из двух действий вашего варианта задания.
Хорошо	5. Реализовать метод, выполняющий второе действие вашего варианта задания
	6. Реализовать перегрузку операций >> и << с использованием методов для ввода и вывода значений полей.
	7. В функции main заполнять объект шаблона с помощью текстового файла. Выводить данные также в файл.
Отлично	8. Написать метод <b>GenerateRandomValues()</b> для генерации случайных данных в полях.
	9. Модификация элемента (выбирает пользователь) массива в шаблонном классе: <ul style="list-style-type: none"> <li>• через консольный ввод;</li> <li>• при помощи метода</li> </ul>

	<b>SetRandomValue()</b> , генерирующего случайное значение.
	10. Для реализации сравнения и присваивания необходимо реализовать перегрузку соответствующих операторов.

### Вариант №24.

Напишите программу, в которой создайте шаблонный класс **TMatrix**, реализующий динамический двумерный массив и функции-методы:

- вычисления суммы элементов главной диагонали;
- нахождения наименьшего значения среди элементов массива, больших среднеарифметического значения по всему массиву.

Протестируйте вашу программу для работы с массивами различных типов: **int** и класса **дробное число** (которое представлено двумя компонентами). Этот класс необходимо создать.

## ХОД РАБОТЫ

### TMatrix.h

```
Lab3 > src > lib > h TMatrix.h > deleteMas()
1  #pragma once
2  #include "/Users/kalmar4ic/Documents/VSCodeC++/Lab3/src/lib/Fraction.h"
   Codeium: Refactor | Explain
3  template <class T>
4  class TMatrix
5  {
6  private:
7      T** matrix;
8      int rows;
9      int cols;
10
11 public:
12     TMatrix();
13     TMatrix(const TMatrix& TM);
14     ~TMatrix();
15
16     istream& In(istream& stream);
17     ostream& Out(ostream& stream) const;
18
19     void SetRandomValue(int min, int max, int rows, int cols);
20     void generateRandomValues(int min, int max);
21     void deleteMas();
22     T mainDiagonal();
23     T findMinNumber();
24 };
25
   Codeium: Refactor | Explain | Generate Function Comment | X
26 template<class T>
27 inline TMatrix<T>::TMatrix()
28 {
29     rows = 0;
30     cols = 0;
31     matrix = nullptr;
32 }
33
   Codeium: Refactor | Explain | Generate Function Comment | X
34 template<class T>
35 inline TMatrix<T>::TMatrix(const TMatrix& TM)
36 {
37     rows = TM.rows;
38     cols = TM.cols;
39     matrix = new T * [rows];
40     for (int i = 0; i < rows; i++)
41     {
```

```

40     for (int i = 0; i < rows; i++)
41     {
42         matrix[i] = new T[cols];
43     }
44
45     for (int i = 0; i < rows; i++)
46     {
47         for (int j = 0; j < cols; j++)
48         {
49             matrix[i][j] = TM.matrix[i][j];
50         }
51     }
52 }
53
54 Codeium: Refactor | Explain | Generate Function Comment | X
55 template<class T>
56 inline TMatrix<T>::~TMatrix()
57 {
58     deleteMas();
59 }
60
61 template<class T>
62 inline istream& TMatrix<T>::In(istream& stream)
63 {
64     deleteMas();
65     if (&stream == &cin)
66     {
67         cout << "enter the number of rows and cols ";
68     }
69     stream >> rows >> cols;
70     matrix = new T * [rows];
71     for (int i = 0; i < rows; i++)
72     {
73         matrix[i] = new T[cols];
74     }
75     for (int j = 0; j < rows; j++)

```

```

74
75     for (int j = 0; j < rows; j++)
76     {
77         for (int k = 0; k < cols; k++)
78         {
79             stream >> matrix[j][k];
80         }
81     }
82     return stream;
83 }
84
85 template<class T>
86 inline ostream& TMatrix<T>::Out(ostream& stream) const
87 {
88     if (&stream != &cout)
89     {
90         stream << rows << " " << cols << endl;
91     }
92     for (int i = 0; i < rows; i++)
93     {
94         for (int j = 0; j < cols; j++)
95         {
96             stream << matrix[i][j] << " ";
97         }
98         stream << endl;
99     }
100     return stream;
101 }
102

```

Codeium: Refactor | Explain | Generate Function Comment | X

```

103 template<class T>
104 inline void TMatrix<T>::SetRandomValue(int min, int max, int rows, int cols)
105 {
106     matrix[rows][cols] = rand() % (max - min + min);
107 }
108

```

Codeium: Refactor | Explain | Generate Function Comment | X

```

109 template<>
110 inline void TMatrix<Fraction>::SetRandomValue(int min, int max, int rows, int cols)
111 {
112     int denominator = rand() % (max - min + min);

```



```

112     int denominator = rand() % (max - min + min);
113     int numerator = rand() % (max - min + min);
114     Fraction F(numerator, denominator);
115     matrix[rows][cols] = F;
116 }
117
Codeium: Refactor | Explain | Generate Function Comment | X
118 template<class T>
119 inline void TMatrix<T>::generateRandomValues(int min, int max)
120 {
121     for (int i = 0; i < rows; i++)
122     {
123         for (int j = 0; j < cols; j++)
124         {
125             matrix[i][j] = rand() % (max - min + min);
126         }
127     }
128 }
129
Codeium: Refactor | Explain | Generate Function Comment | X
130 template <class T>
131 inline void TMatrix<T>::deleteMas()
132 {
133     for (int i = 0; i < rows; ++i)
134     {
135         delete[] matrix[i];
136     }
137     delete[] matrix;
138     rows = 0;
139     cols = 0;
140     matrix = nullptr;
141 }
142
Codeium: Refactor | Explain | Generate Function Comment | X
143 template<>
144 inline void TMatrix<Fraction>::generateRandomValues(int min, int max)
145 {
146     for (int i = 0; i < rows; i++)
147     {
148         for (int j = 0; j < cols; j++)
149         {

```

```

148         for (int j = 0; j < cols; j++)
149         {
150             int denominator = rand() % (max - min + min);
151             int numerator = rand() % (max - min + min);
152             Fraction F(numerator, denominator);
153             matrix[i][j] = F;
154         }
155     }
156 }
157
Codeium: Refactor | Explain | Generate Function Comment | X
158 template<class T>
159 inline T TMatrix<T>::mainDiagonal()
160 {
161     T count = matrix[0][0];
162     for (int i = 0, j = 0; i < rows && i < cols; i++, j++)
163     {
164         count += matrix[i][j];
165     }
166     count -= matrix[0][0];
167     //cout << "Main Diagonal: " << count << endl;
168     return count;
169 }
170
Codeium: Refactor | Explain | Generate Function Comment | X
171 template<class T>
172 inline T TMatrix<T>::findMinNumber()
173 {
174     T averageValue = matrix[0][0];
175     T min;
176     T max = matrix[0][0];
177     for (int i = 0; i < rows; i++)
178     {
179         for (int j = 0; j < cols; j++)
180         {
181             averageValue += matrix[i][j];
182             if (matrix[i][j] > max)
183             {
184                 max = matrix[i][j];

```

```

Lab3 > src > lib > h TMatrix.h > deleteMas()
172 inline T TMatrix<T>::findMinNumber()
177     for (int i = 0; i < rows; i++)
179         for (int j = 0; j < cols; j++)
182             if (matrix[i][j] > max)
183             {
184                 max = matrix[i][j];
185             }
186         }
187     }
188     averageValue -= matrix[0][0];
189     averageValue /= (rows + cols);
190     min = max;
191
192     for (int i = 0; i < rows; i++)
193     {
194         for (int j = 0; j < cols; j++)
195         {
196             if (matrix[i][j] > averageValue)
197             {
198                 if (matrix[i][j] < min)
199                 {
200                     min = matrix[i][j];
201                 }
202             }
203         }
204     }
205     //cout << "Min number: " << min << endl;
206     return min;
207 }
208
209 template<class T>
210 istream& operator>>(istream& stream, TMatrix<T>& TM)
211 {
212     return TM.In(stream);;
213 }
214
215 template<class T>
216 ostream& operator<<(ostream& stream, const TMatrix<T>& TM)
217 {
218     TM.Out(stream);
219     return stream;
220 }
221

```

## Fraction.h

Lab3 > src > lib > h Fraction.h > ...

```
2  class Fraction
3  {
4  private:
5      int numerator;
6      int denominator;
7
8  public:
9      Fraction();
10     Fraction(int numerator, int denominator);
11     Fraction(const Fraction& F);
12     ~Fraction();
13
14     friend istream& operator>> (istream& stream, Fraction& F);
15     friend ostream& operator<< (ostream& stream, const Fraction& F);
16
17     bool operator==(const Fraction& F) const;
18     bool operator>(const Fraction& F) const;
19     bool operator<(const Fraction& F) const;
20     bool operator>=(const Fraction& F) const;
21     bool operator<=(const Fraction& F) const;
22
23     Fraction& operator=(const Fraction& F);
24     Fraction& operator+=(const Fraction& F);
25     Fraction& operator/=(const int num);
26     Fraction& operator-=(const Fraction& F);
27
28 };
```

## Fraction.cpp

Lab3 > src > lib > C++ Fraction.cpp > ...

```
1  #include <iostream>
2  #include <ctime>
3  using namespace std;
4  #include "/Users/kalmar4ic/Documents/VSCodeC++/Lab3/src/lib/Fraction.h"
5
6  Codeium: Refactor | Explain | Generate Function Comment | X
7  Fraction::Fraction()
8  {
9      numerator = 0;
10     denominator = 1;
11 }
12
13 Codeium: Refactor | Explain | Generate Function Comment | X
14 Fraction::Fraction(int numerator, int denominator)
15 {
16     this->numerator = numerator;
17     this->denominator = denominator;
18 }
19
20 Codeium: Refactor | Explain | Generate Function Comment | X
21 Fraction::Fraction(const Fraction& F)
22 {
23     numerator = F.numerator;
24     denominator = F.denominator;
25 }
```

```

23      Codeium: Refactor | Explain | Generate Function Comment | X
24      Fraction::~~Fraction()
25      {
26      }
27      }
28
29      istream& operator>>(istream& stream, Fraction& F)
30      {
31          char slash;
32          stream >> F.numerator >> slash >> F.denominator;
33          return stream;
34      }
35
36      ostream& operator<<(ostream& stream, const Fraction& F)
37      {
38          stream << F.numerator << "/" << F.denominator;
39          return stream;
40      }
41
42      Codeium: Refactor | Explain | Generate Function Comment | X
43      bool Fraction::operator==(const Fraction& F) const
44      {
45          return numerator * F.denominator == F.numerator * denominator;
46      }
47
48      Codeium: Refactor | Explain | Generate Function Comment | X
49      bool Fraction::operator>(const Fraction& F) const
50      {
51          return numerator * F.denominator > F.numerator * denominator;
52      }
53
54      Codeium: Refactor | Explain | Generate Function Comment | X
55      bool Fraction::operator<(const Fraction& F) const
56      {
57          return numerator * F.denominator < F.numerator * denominator;
58      }

```

```

56 Codeium: Refactor | Explain | Generate Function Comment | X
57 bool Fraction::operator>=(const Fraction& F) const
58 {
59     return numerator * F.denominator >= F.numerator * denominator;
60 }
61
62 Codeium: Refactor | Explain | Generate Function Comment | X
63 bool Fraction::operator<=(const Fraction& F) const
64 {
65     return numerator * F.denominator <= F.numerator * denominator;
66 }
67
68 Fraction& Fraction::operator=(const Fraction& F)
69 {
70     numerator = F.numerator;
71     denominator = F.denominator;
72     return *this;
73 }
74
75 Fraction& Fraction::operator+=(const Fraction& F)
76 {
77     numerator = numerator * F.denominator + F.numerator * denominator;
78     denominator = denominator * F.denominator;
79
80     return *this;
81 }
82
83 Fraction& Fraction::operator/=(const int num)
84 {
85     numerator = numerator * num;
86     return *this;
87 }
88
89 Fraction& Fraction::operator-=(const Fraction &F)
90 {
91     numerator = numerator * F.denominator - F.numerator * denominator;
92     denominator = denominator * F.denominator;
93
94     return *this;
95 }

```

## Main.cpp

```
Lab3 > src > C++ main.cpp > main()
1  #include <iostream>
2  #include <fstream>
3  #include <ctime>
4  #include <string>
5  using namespace std;
6  #include "/Users/kalmar4ic/Documents/VSCodeC++/Lab3/src/lib/TMatrix.h"
7  #include "/Users/kalmar4ic/Documents/VSCodeC++/Lab3/src/lib/Fraction.h"
8
Codeium: Refactor | Explain | Generate Function Comment | X
9  int main()
10 {
11     srand(time(0));
12     ifstream Fromfile;
13     ofstream Tofile;
14     string filename;
15     int choice;
16     int choice2;
17     int max, min;
18     cout << "Нажмите 1 - чтобы использовать значения int" << endl;
19     cout << "Нажмите 2 - чтобы использовать значения Fraction" << endl;
20     cin >> choice;
21     cin.ignore();
22     if (choice == 1)
23     {
24         TMatrix<int> matr;
25         cout << "Нажмите 1 - чтобы ввести значения с клавиатуры" << endl;
26         cout << "Нажмите 2 - чтобы ввести значения из файла" << endl;
27         cin >> choice2;
28         if (choice2 == 1)
29         {
30             matr.In(cin);
31         }
32         else if (choice2 == 2)
33         {
34             cout << "Введите название файла: ";
35             cin >> filename;
36             Fromfile.open(filename);
37             matr.In(Fromfile);
38             Fromfile.close();
39         }
40         matr.Out(cout);
41         cout << "Главная диагональ: " << matr.mainDiagonal() << endl;
42         cout << "Минимальное число: " << matr.findMinNumber() << endl;
43     }
```

```

41     cout << "Главная диагональ: " << matr.mainDiagonal() << endl;
42     cout << "Минимальное число: " << matr.findMinNumber() << endl;
43
44     cout << "Если вы хотите задать значения рандомно – нажмите 3" << endl;
45     cout << "Если вы хотите записать значения в файл – нажмите 4" << endl;
46     cout << "Если вы хотите задать значения закончить программу – нажмите 5" << endl;
47     cin >> choice2;
48     cin.ignore();
49     if (choice2 == 3)
50     {
51         cout << "Введите максимальное и минимальное значение: ";
52         cin >> max >> min;
53         matr.generateRandomValues(min, max);
54         matr.Out(cout);
55     }
56     else if(choice2 == 4)
57     {
58         cout << "Введите название файла: ";
59         cin >> filename;
60         Tofile.open(filename);
61         matr.Out(Tofile);
62         Tofile.close();
63     }
64
65 }
66 else if (choice == 2)
67 {
68     TMatrix<Fraction> matr;
69     cout << "Нажмите 1 – чтобы ввести значения с клавиатуры" << endl;
70     cout << "Нажмите 2 – чтобы ввести значения из файла" << endl;
71     cin >> choice2;
72     if (choice2 == 1)
73     {
74         matr.In(cin);
75     }
76     else if (choice2 == 2)
77     {
78         cout << "Введите название файла: ";
79         cin >> filename;
80         Fromfile.open(filename);
81         matr.In(Fromfile);

```

```

82         Fromfile.close();
83     }
84     matr.Out(cout);
85     cout << "Главная диагональ: " << matr.mainDiagonal() << endl;
86     cout << "Минимальное число: " << matr.findMinNumber() << endl;
87     cout << "Если вы хотите задать значения рандомно – нажмите 3" << endl;
88     cout << "Если вы хотите записать значения в файл – нажмите 4" << endl;
89     cout << "Если вы хотите задать значения закончить программу – нажмите 5" << endl;
90     cin >> choice2;
91     cin.ignore();
92     if (choice2 == 3)
93     {
94         cout << "Введите максимальное и минимальное значение: ";
95         cin >> max >> min;
96         matr.generateRandomValues(min, max);
97         matr.Out(cout);
98     }
99     else if(choice2 == 4)
100    {
101        cout << "Введите название файла: ";
102        cin >> filename;
103        Tofile.open(filename);
104        matr.Out(Tofile);
105        Tofile.close();
106    }
107    }
108    return 0;
109 }

```



Демонстрация работы программы:

Ввод с файла тип int (Файл: 1.txt)

```
Lab3 > 1.txt
1 2 2
2 1 2
3 3 4
4
```

Результат программы:

```
Нажмите 2 - чтобы использовать значения Fraction
1
Нажмите 1 - чтобы ввести значения с клавиатуры
Нажмите 2 - чтобы ввести значения из файла
2
Введите название файла: 1.txt
1 2
3 4
Главная диагональ: 5
Минимальное число: 3
Если вы хотите задать значения вручную - нажмите 1
```

Ввод из файла типа Fraction (Файл 2.txt)

```
Lab3 > 2.txt
1 2 2
2 1/4 2/5
3 33/97 15/6
4
```

Результат программы:

```
Нажмите 2 - чтобы использовать значения Fraction
2
Нажмите 1 - чтобы ввести значения с клавиатуры
Нажмите 2 - чтобы ввести значения из файла
2
Введите название файла: 2.txt
1/4 2/5
33/97 15/6
Главная диагональ: 1056/384
Минимальное число: 15/6
Если вы хотите задать значения вручную - нажмите 1
```

Ввод значений с клавиатуры тип int:

```
./out/main
Нажмите 1 – чтобы использовать значения int
Нажмите 2 – чтобы использовать значения Fraction
1
Нажмите 1 – чтобы ввести значения с клавиатуры
Нажмите 2 – чтобы ввести значения из файла
1
enter the number of rows and cols 3 3
1 2 3 4 5 6 7 8 9
1 2 3
4 5 6
7 8 9
Главная диагональ: 15
Минимальное число: 8
```

Ввод значений с клавиатуры тип Fraction:

```
kalmar4ic@MacBook-Air-Sergej Lab3 % make run
./out/main
Нажмите 1 – чтобы использовать значения int
Нажмите 2 – чтобы использовать значения Fraction
2
Нажмите 1 – чтобы ввести значения с клавиатуры
Нажмите 2 – чтобы ввести значения из файла
1
enter the number of rows and cols 2 2
12/56 14/55 18/20 44/50
12/56 14/55
18/20 44/50
Главная диагональ: 9608704/8780800
Минимальное число: 12/56
Если вы хотите задать значения вручную – нажмите 2
```

Генератор случайных чисел для Int:

```
kalmar4ic@MacBook-Air-Sergej Lab3 % make run
./out/main
Нажмите 1 – чтобы использовать значения int
Нажмите 2 – чтобы использовать значения Fraction
1
Нажмите 1 – чтобы ввести значения с клавиатуры
Нажмите 2 – чтобы ввести значения из файла
2
Введите название файла: 1.txt
1 2
3 4
Главная диагональ: 5
Минимальное число: 3
Если вы хотите задать значения случайно – нажмите 3
Если вы хотите записать значения в файл – нажмите 4
Если вы хотите задать значения закончить программу – нажмите 5
3
Введите максимальное и минимальное значение: 50 20
28 43
29 27
kalmar4ic@MacBook-Air-Sergej Lab3 %
```

## Генератор случайных чисел типа Fraction:

```
.70017main
Нажмите 1 – чтобы использовать значения int
Нажмите 2 – чтобы использовать значения Fraction
2
Нажмите 1 – чтобы ввести значения с клавиатуры
Нажмите 2 – чтобы ввести значения из файла
2
Введите название файла: 2.txt
1/4 2/5
33/97 15/6
Главная диагональ: 1056/384
Минимальное число: 15/6
Если вы хотите задать значения случайно – нажмите 3
Если вы хотите записать значения в файл – нажмите 4
Если вы хотите задать значения закончить программу – нажмите 5
3
Введите максимальное и минимальное значение: 50 10
35/10 23/43
18/24 40/48
kalmar4ic@MacBook-Air-Sergej Lab3 %
```

## Запись в файл типа int:

```
170017main
Нажмите 1 – чтобы использовать значения int
Нажмите 2 – чтобы использовать значения Fraction
1
Нажмите 1 – чтобы ввести значения с клавиатуры
Нажмите 2 – чтобы ввести значения из файла
1
enter the number of rows and cols 2 2
5902 54 4932109 5345
5902 54
4932109 5345
Главная диагональ: 11247
Минимальное число: 4932109
Если вы хотите задать значения случайно – нажмите 3
Если вы хотите записать значения в файл – нажмите 4
Если вы хотите задать значения закончить программу – нажмите 5
4
Введите название файла: 1.txt
o kalmar4ic@MacBook-Air-Sergej Lab3 %
```

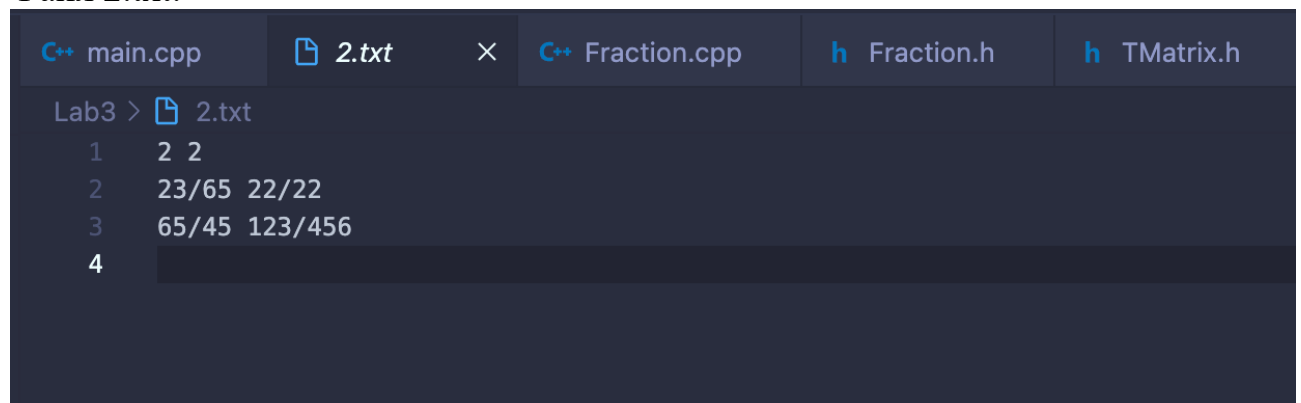
## Файл 1.txt

```
C++ Fraction.cpp 1.txt x h Fraction.h h TMatrix.h C++ ma
Lab3 > 1.txt
1 2 2
2 5902 54
3 4932109 5345
4
```

Запись в файл типа Fraction:

```
● kalmar4ic@MacBook-Air-Sergej Lab3 % make run
./out/main
Нажмите 1 – чтобы использовать значения int
Нажмите 2 – чтобы использовать значения Fraction
2
Нажмите 1 – чтобы ввести значения с клавиатуры
Нажмите 2 – чтобы ввести значения из файла
1
enter the number of rows and cols 2 2
23/65 22/22 65/45 123/456
23/65 22/22
65/45 123/456
Главная диагональ: 78090675/125229000
Минимальное число: 123/456
Если вы хотите задать значения случайно – нажмите 3
Если вы хотите записать значения в файл – нажмите 4
Если вы хотите задать значения закончить программу – нажмите 5
4
Введите название файла: 2.txt
○ kalmar4ic@MacBook-Air-Sergej Lab3 %
```

Файл 2.txt:



```
Lab3 > 2.txt
1 2 2
2 23/65 22/22
3 65/45 123/456
4
```

## ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какой принцип ООП реализуется с помощью шаблонных функций и классов?

*Механизм шаблонов реализует в C++ параметрический полиморфизм.*

2. Когда целесообразно использовать шаблоны функций?

*Когда одна функция может использоваться с несколькими типами данных*

3. Каков общий вид объявления шаблона функции?

*template <typename T>*

*Имя\_функции(аргументы\_функции) {  
}*

4. Каков синтаксис объявления шаблонного класса?

*template <typename T>*

*class ИмяКласса {  
};*

5. В чем особенности определения шаблонного класса?

*Чтобы сделать класс шаблонным, необходимо перед классом написать `template <typename T>` или `<class T>`.*

6. В чем особенности определения метода шаблонного класса вне класса?

*При определении метода вне тела класса необходимо указать, что метод принадлежит шаблонному классу. А также после имени класса необходимо указать тип (имя\_класса<параметры\_шаблона>).*

7. Что такое специализация шаблона?

*Возможность в C++ создать конкретную реализацию шаблона для определённого типа или набора типов данных.*

8. Как объявить объект шаблонного класса?

*имя\_класса<аргумент\_шаблона> имя\_объекта;*

9. Что такое инстанцирование?

*Процесс генерации кода функции или класса по шаблону для конкретных параметров.*

10. Чем в строке **template <class T>** может быть **T**?

**T** может быть любым типом – как встроенным в язык C++, так и определенным программистом.

## **ВЫВОДЫ**

Были получены знания и навыки работы с шаблонами функций и классов.