

Учреждение образования
“Белорусский государственный университет информатики и
радиоэлектроники”

ОТЧЕТ

По Task_6

Выполнил:

Студент ФКП

Дашко И.Д.

Гр. 114301

Проверил:

Кабариха В.А.

Минск 2021

До изменений ввод и вывод были реализованы в функциях:

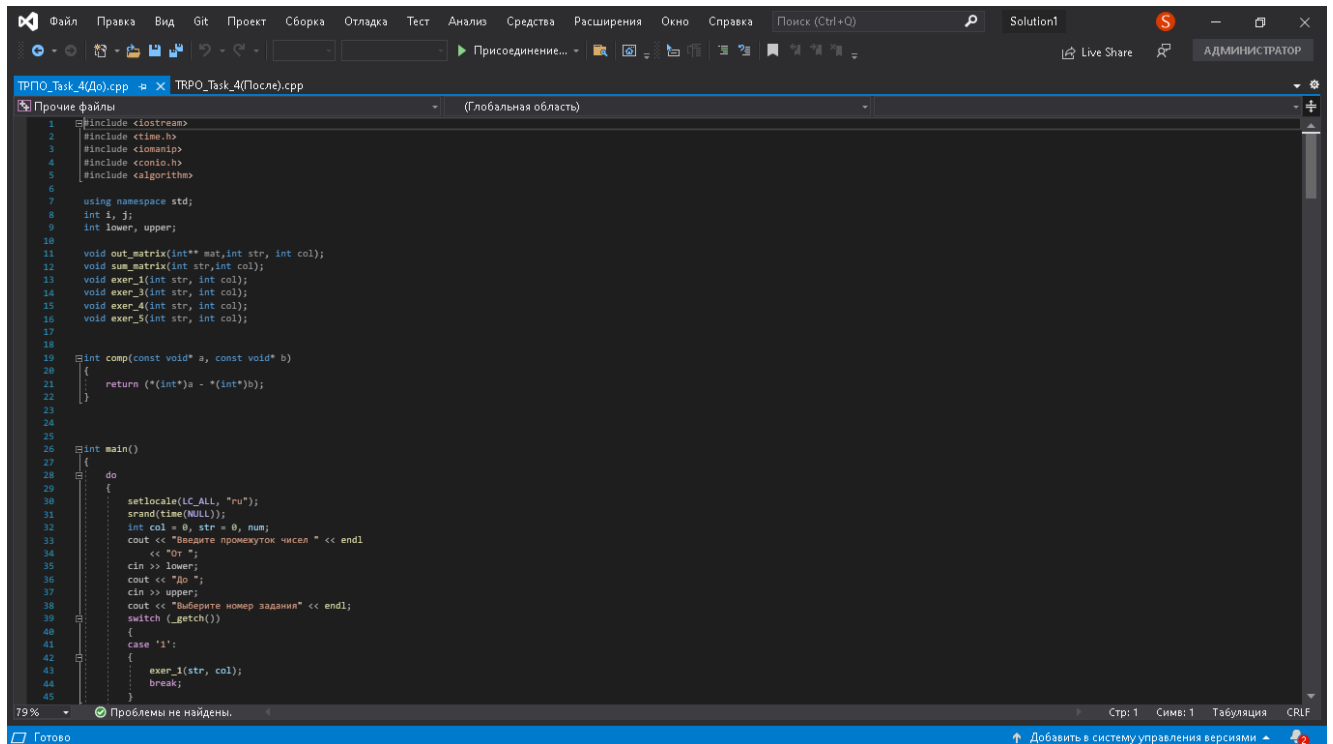
```
25 int main()
26 {
27     do
28     {
29         setlocale(LC_ALL, "ru");
30         srand(time(NULL));
31         int col = 0, str = 0, num;
32         cout << "Введите промежуток чисел " << endl;
33         << "0-1 ";
34         cin >> lower;
35         cout << "0-1 ";
36         cin >> upper;
37         cout << "Введите номер задания" << endl;
38         switch (_getch())
39         {
40             case '1':
41             {
42                 exer_1(str, col);
43                 break;
44             }
45             case '2':
46             {
47                 sum_matrix(str, col);
48                 break;
49             }
50             case '3':
51             {
52                 exer_3(str, col);
53                 break;
54             }
55             case '4':
56             {
57                 exer_4(str, col);
58                 break;
59             }
60             case '5':
61             {
62                 exer_5(str, col);
63                 break;
64             }
65             default:
66             {
67                 cout << "Такого задания нет" << endl;
68                 break;
69             }
70         }
71         cout << "Нажмите 1 для продолжения работы с программой" << endl;
72         while (_getch() != '1');
73     }
74 }
75
76 void exer_1(int str, int col)
```

После изменений всё было перенесено в функцию main()

```
61 case '1':
62 {
63     cout << "Задайте размеры матрицы" << endl;
64     cin >> str >> col;
65     int** mat = createMatrix(mat, str, col);
66     fillMatrix(mat, str, col);
67     out_matrix(mat, str, col);
68     deleteMatrix(str, mat);
69     break;
70 }
71 case '2':
72 {
73     cout << "Задайте размеры матрицы" << endl;
74     cin >> str >> col;
75     int** mat = createMatrix(mat, str, col);
76     fillMatrix(mat, str, col);
77     out_matrix(mat, str, col);
78     cout << "Сумма равно " << sum_matrix(mat, str, col) << endl;
79     cout << "Среднее арифметическое равно " << sum_matrix(mat, str, col) / (col * str) << endl;
80     deleteMatrix(str, mat);
81     break;
82 }
83 case '3':
84 {
85     cout << "Задайте размеры матрицы" << endl;
86     cin >> str >> col;
87     int** mat = createMatrix(mat, str, col);
88     fillMatrix(mat, str, col);
89     cout << "Введите размеры второй матрицы (количество столбцов первой матрицы должно быть равно количеству строк первой матрицы)" << endl;
90     int str_1, col_1;
91     do
92     {
93         cin >> str_1 >> col_1;
94         if (col_1 != str)
95         {
96             cout << "Количество столбцов первой матрицы не равно количеству строк первой матрицы" << endl;
97         }
98     } while (col_1 != str_1);
99     int** mat_1 = createMatrix(mat_1, str_1, col_1);
100     fillMatrix(mat_1, str_1, col_1);
101     cout << "Первая ";
102     out_matrix(mat, str, col);
103     cout << "Вторая ";
104     out_matrix(mat_1, str_1, col_1);
105     int** work_mat = createMatrix(work_mat, str, col_1);
106     work_mat[0][0] = mat[0][0] + mat_1[0][0];
107     out_matrix(work_mat, str, col_1);
108     deleteMatrix(str, mat);
109     deleteMatrix(str_1, mat_1);
110     break;
111 }
```

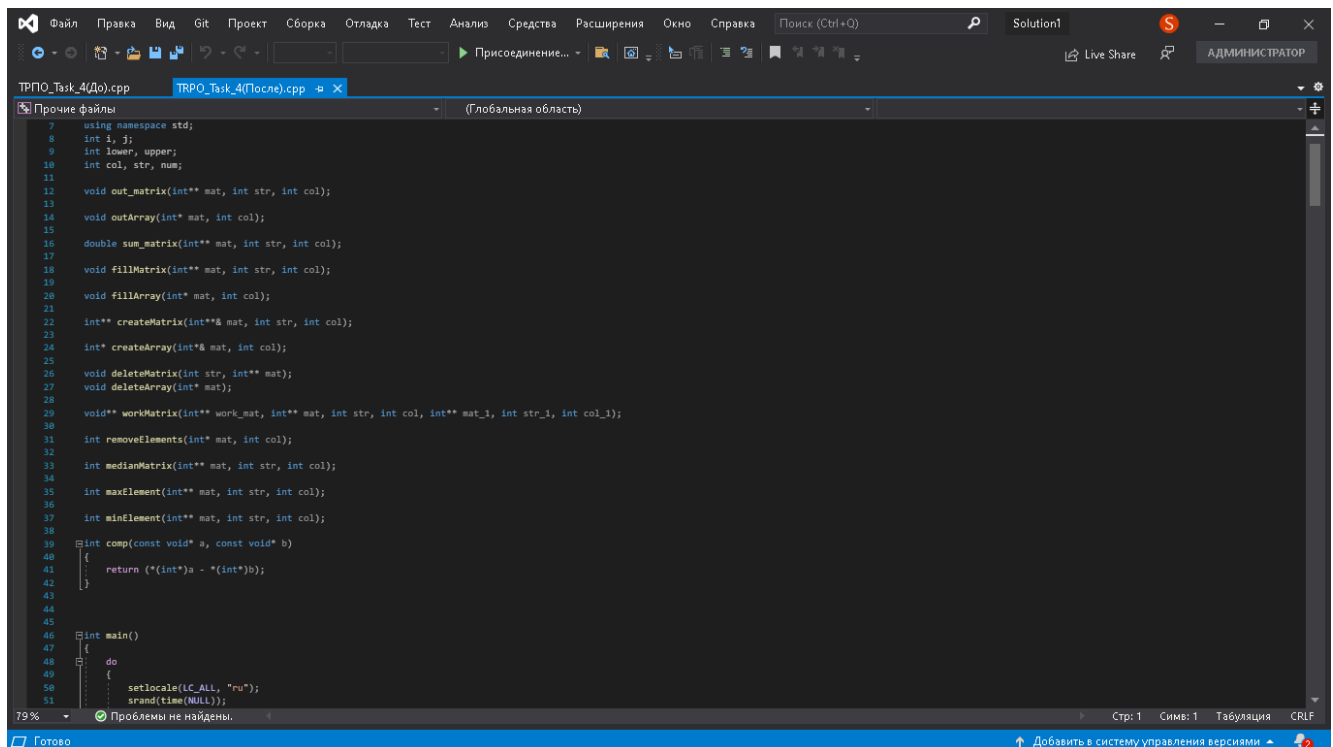
Ещё одним из изменений стало выделение новых функций и изменение названий старых

До:



```
1 #include <iostream>
2 #include <time.h>
3 #include <iomanip>
4 #include <conio.h>
5 #include <algorithm>
6
7 using namespace std;
8 int i, j;
9 int lower, upper;
10
11 void out_matrix(int** mat, int str, int col);
12 void sum_matrix(int str, int col);
13 void exer_1(int str, int col);
14 void exer_3(int str, int col);
15 void exer_4(int str, int col);
16 void exer_5(int str, int col);
17
18
19 int comp(const void* a, const void* b)
20 {
21     return (*(int*)a - *(int*)b);
22 }
23
24
25
26 int main()
27 {
28     do
29     {
30         setlocale(LC_ALL, "ru");
31         srand(time(NULL));
32         int col = 0, str = 0, num;
33         cout << "Введите промежутки чисел " << endl;
34         cout << "0: ";
35         cin >> lower;
36         cout << "до ";
37         cin >> upper;
38         cout << "Выберите номер задания" << endl;
39         switch (_getch())
40         {
41             case '1':
42             {
43                 exer_1(str, col);
44                 break;
45             }
46         }
47     }
48 }
```

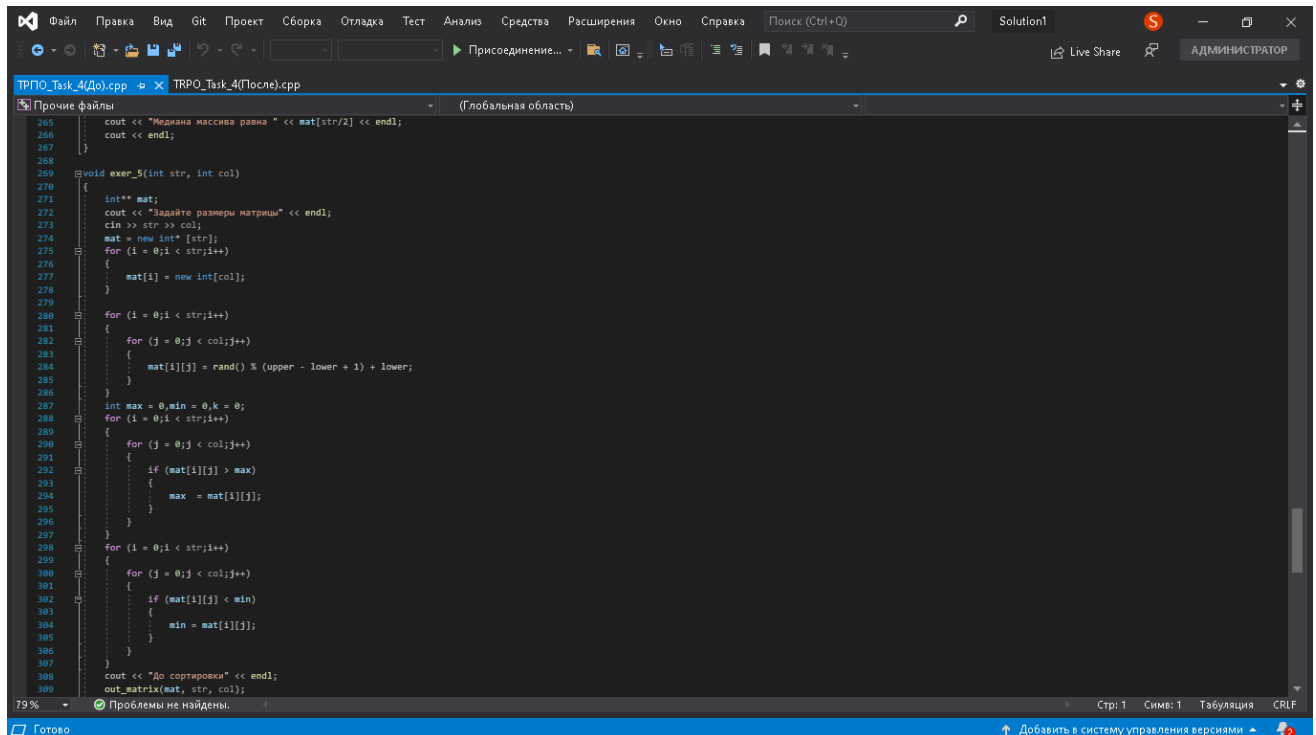
После



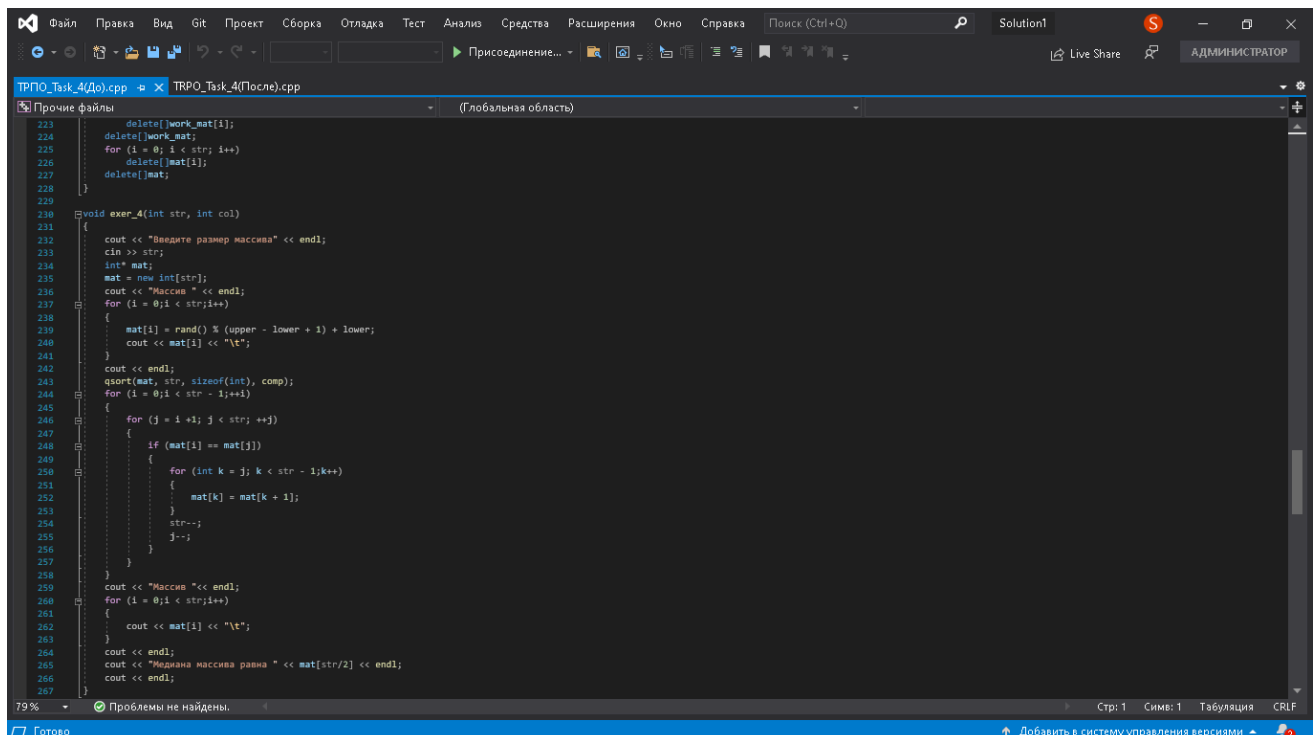
```
7 using namespace std;
8 int i, j;
9 int lower, upper;
10 int col, str, num;
11
12 void out_matrix(int** mat, int str, int col);
13
14 void outArray(int* mat, int col);
15
16 double sum_matrix(int** mat, int str, int col);
17
18 void fillMatrix(int** mat, int str, int col);
19
20 void fillArray(int* mat, int col);
21
22 int** createMatrix(int**& mat, int str, int col);
23
24 int* createArray(int*& mat, int col);
25
26 void deleteMatrix(int str, int** mat);
27 void deleteArray(int* mat);
28
29 void** workMatrix(int** work_mat, int** mat, int str, int col, int** mat_1, int str_1, int col_1);
30
31 int removeElements(int* mat, int col);
32
33 int medianMatrix(int** mat, int str, int col);
34
35 int maxElement(int** mat, int str, int col);
36
37 int minElement(int** mat, int str, int col);
38
39 int comp(const void* a, const void* b)
40 {
41     return (*(int*)a - *(int*)b);
42 }
43
44
45
46 int main()
47 {
48     do
49     {
50         setlocale(LC_ALL, "ru");
51         srand(time(NULL));
52     }
53 }
```

После изменений в функциях выполняется только вычисление(за исключением функции вывода) , до этого там был реализован ВВОД И ВЫВОД :

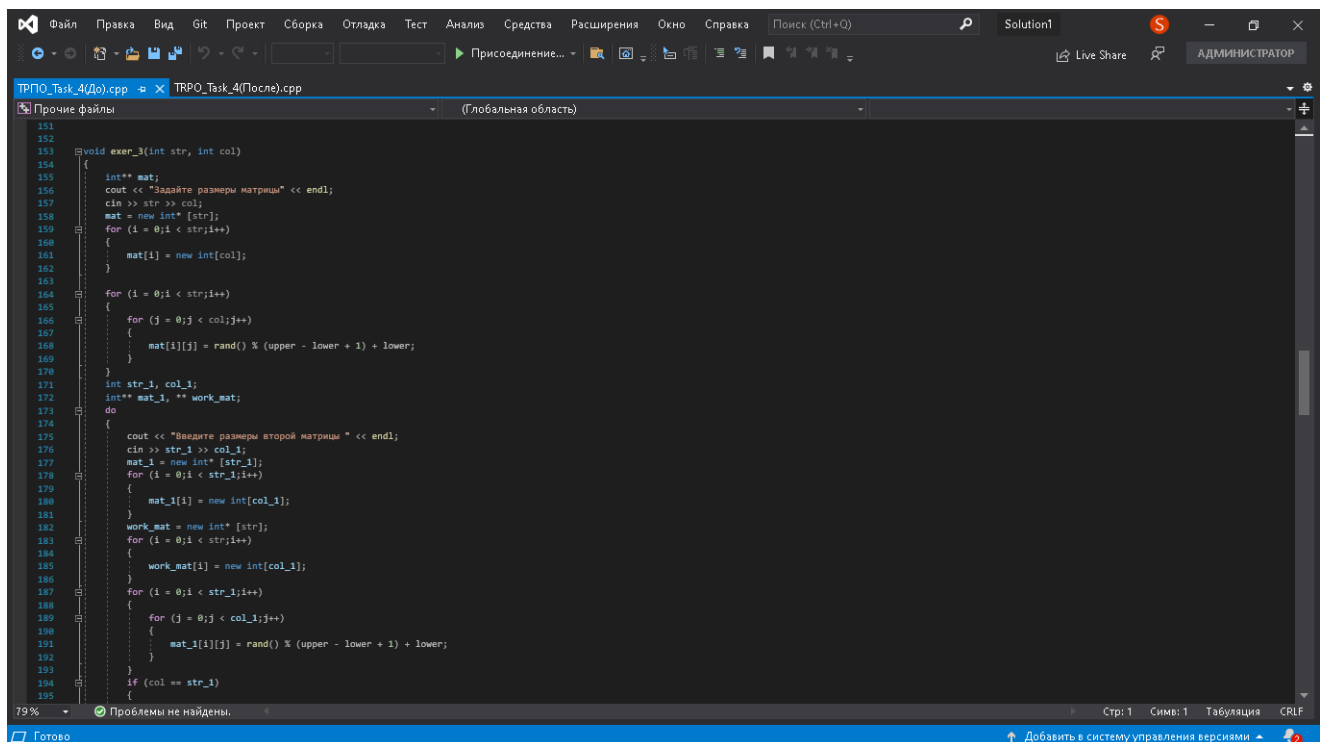
До:



```
265 cout << "Медиана массива равна " << mat[ctr/2] << endl;
266 cout << endl;
267 }
268
269 void exer_5(int str, int col)
270 {
271     int** mat;
272     cout << "Задайте размеры матрицы" << endl;
273     cin >> str >> col;
274     mat = new int* [str];
275     for (i = 0; i < str; i++)
276     {
277         mat[i] = new int[col];
278     }
279
280     for (i = 0; i < str; i++)
281     {
282         for (j = 0; j < col; j++)
283         {
284             mat[i][j] = rand() % (upper - lower + 1) + lower;
285         }
286     }
287     int max = 0, min = 0, k = 0;
288     for (i = 0; i < str; i++)
289     {
290         for (j = 0; j < col; j++)
291         {
292             if (mat[i][j] > max)
293             {
294                 max = mat[i][j];
295             }
296         }
297     }
298     for (i = 0; i < str; i++)
299     {
300         for (j = 0; j < col; j++)
301         {
302             if (mat[i][j] < min)
303             {
304                 min = mat[i][j];
305             }
306         }
307     }
308     cout << "До сортировки" << endl;
309     out_matrix(mat, str, col);
```



```
221 delete[]work_mat[i];
222 delete[]work_mat;
223 for (i = 0; i < str; i++)
224     delete[]mat[i];
225 delete[]mat;
226 }
227
228 void exer_4(int str, int col)
229 {
230     cout << "Введите размер массива" << endl;
231     cin >> str;
232     int* mat;
233     mat = new int[str];
234     cout << "Массив " << endl;
235     for (i = 0; i < str; i++)
236     {
237         mat[i] = rand() % (upper - lower + 1) + lower;
238         cout << mat[i] << "\t";
239     }
240     cout << endl;
241     qsort(mat, str, sizeof(int), comp);
242     for (i = 0; i < str - 1; i++)
243     {
244         for (j = i + 1; j < str; ++j)
245         {
246             if (mat[i] == mat[j])
247             {
248                 for (int k = j; k < str - 1; k++)
249                 {
250                     mat[k] = mat[k + 1];
251                 }
252                 str--;
253                 j--;
254             }
255         }
256     }
257     cout << "Массив " << endl;
258     for (i = 0; i < str; i++)
259     {
260         cout << mat[i] << "\t";
261     }
262     cout << endl;
263     cout << "Медиана массива равна " << mat[ctr/2] << endl;
264     cout << endl;
265 }
```

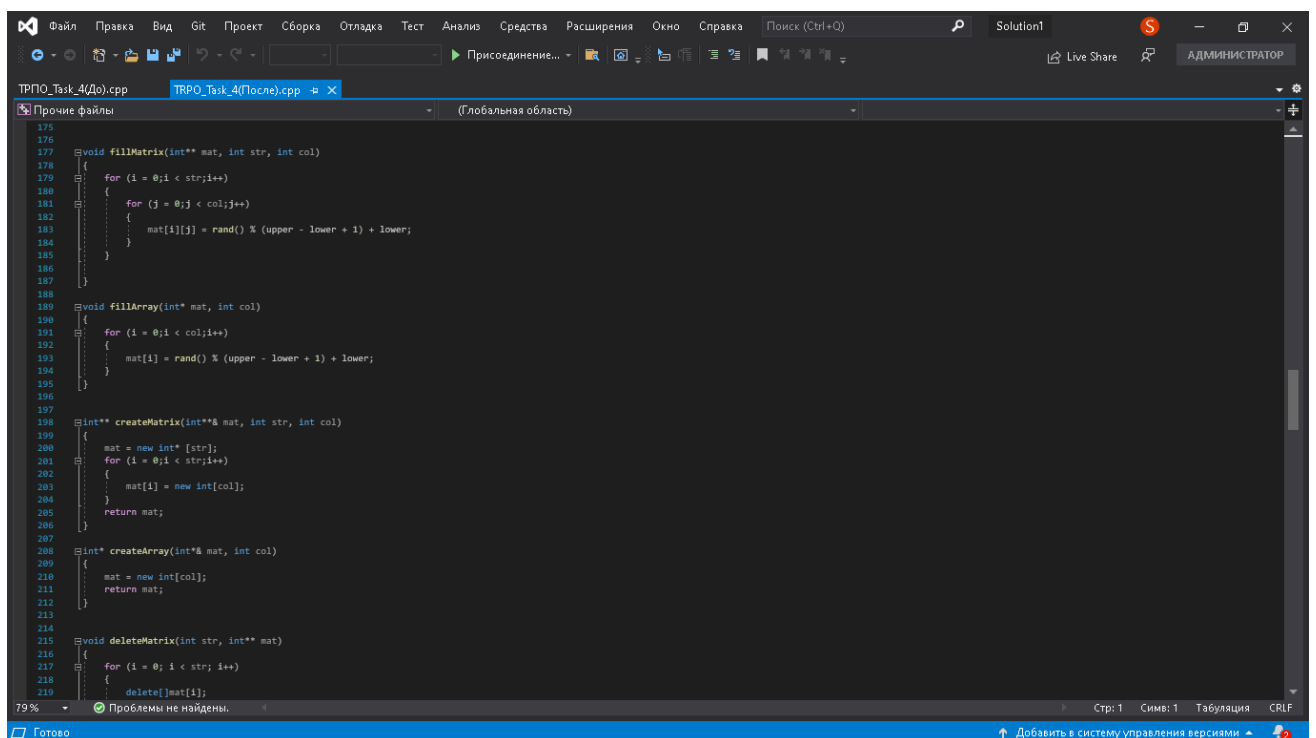


```
151
152
153 void exer_3(int str, int col)
154 {
155     int** mat;
156     cout << "Задайте размеры матрицы" << endl;
157     cin >> str >> col;
158     mat = new int* [str];
159     for (i = 0; i < str; i++)
160     {
161         mat[i] = new int[col];
162     }
163
164     for (i = 0; i < str; i++)
165     {
166         for (j = 0; j < col; j++)
167         {
168             mat[i][j] = rand() % (upper - lower + 1) + lower;
169         }
170     }
171     int str_1, col_1;
172     int** mat_1, ** work_mat;
173     do
174     {
175         cout << "Введите размеры второй матрицы " << endl;
176         cin >> str_1 >> col_1;
177         mat_1 = new int* [str_1];
178         for (i = 0; i < str_1; i++)
179         {
180             mat_1[i] = new int[col_1];
181         }
182         work_mat = new int* [str];
183         for (i = 0; i < str; i++)
184         {
185             work_mat[i] = new int[col_1];
186         }
187         for (i = 0; i < str_1; i++)
188         {
189             for (j = 0; j < col_1; j++)
190             {
191                 mat_1[i][j] = rand() % (upper - lower + 1) + lower;
192             }
193         }
194         if (col == str_1)
195         {
```

79% Проблемы не найдены. Стр: 1 Симв: 1 Табуляция CRLF

Готово

После:



```
175
176
177 void fillMatrix(int** mat, int str, int col)
178 {
179     for (i = 0; i < str; i++)
180     {
181         for (j = 0; j < col; j++)
182         {
183             mat[i][j] = rand() % (upper - lower + 1) + lower;
184         }
185     }
186 }
187
188 void fillArray(int* mat, int col)
189 {
190     for (i = 0; i < col; i++)
191     {
192         mat[i] = rand() % (upper - lower + 1) + lower;
193     }
194 }
195
196 int** createMatrix(int**& mat, int str, int col)
197 {
198     {
199         mat = new int* [str];
200         for (i = 0; i < str; i++)
201         {
202             mat[i] = new int[col];
203         }
204     }
205     return mat;
206 }
207
208 int* createArray(int*& mat, int col)
209 {
210     {
211         mat = new int[col];
212     }
213 }
214
215 void deleteMatrix(int str, int** mat)
216 {
217     for (i = 0; i < str; i++)
218     {
219         delete[] mat[i];
220     }
```

79% Проблемы не найдены. Стр: 1 Симв: 1 Табуляция CRLF

Готово

```
229 void out_matrix(int** mat, int str, int col)
230 {
231     cout << "Матрица " << endl;
232     for (i = 0; i < str; i++)
233     {
234         for (j = 0; j < col; j++)
235         {
236             cout << setw(10) << mat[i][j];
237         }
238         cout << endl;
239     }
240 }
241
242 void outArray(int* mat, int col)
243 {
244     cout << "Массив " << endl;
245     for (i = 0; i < col; i++)
246     {
247         cout << mat[i] << " ";
248     }
249     cout << endl;
250 }
251
252 double sum_matrix(int** mat, int str, int col)
253 {
254     double sum = 0;
255     for (i = 0; i < str; i++)
256     {
257         for (j = 0; j < col; j++)
258         {
259             sum += mat[i][j];
260         }
261     }
262     return sum;
263 }
264
265 void** workMatrix(int** work_mat, int** mat, int str, int col, int** mat_1, int str_1, int col_1)
266 {
267     for (i = 0; i < str; i++)
268     {
269         for (j = 0; j < col_1; j++)
```

```
283 }
284
285 int removeElements(int* mat, int col)
286 {
287     qsort(mat, col, sizeof(int), comp);
288     for (i = 0; i < col - 1; i++)
289     {
290         for (j = i + 1; j < col; j++)
291         {
292             if (mat[i] == mat[j])
293             {
294                 for (int k = j; k < col - 1; k++)
295                 {
296                     mat[k] = mat[k + 1];
297                 }
298                 col--;
299                 j--;
300             }
301         }
302     }
303     return col;
304 }
305
306 int medianMatrix(int** mat, int str, int col)
307 {
308     int k = 0;
309     int* mat_5;
310     mat_5 = new int[str * col];
311     for (i = 0; i < str; i++)
312     {
313         for (j = 0; j < col; j++)
314         {
315             mat_5[k] = mat[i][j];
316             k++;
317         }
318     }
319     qsort(mat_5, col * str, sizeof(int), comp);
320     k = 0;
321     for (i = 0; i < str; i++)
322     {
323         for (j = 0; j < col; j++)
324         {
325             mat[i][j] = mat_5[k];
326             k++;
327         }
```

В результате проделанной работы код стал более читабельным функции стали чистыми и количество строк изменилось с 338 на 364.