Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский химико-технологический университет имени Д.И. Менделеева»

Кафедра информационных компьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

Вариант 3

Выполнил студент группы КС-30 (Ноль Эльвира Гарриевна)

Ссылка на репозиторий: (https://github.com/Elviranng/Nol-Elvira-KS-30/tree/master/lab5.sem2)

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Дата сдачи: (05.06.21)

Оглавление

[Описание задачи. 2](#_Toc69722552)

[Описание алгоритма. 3](#_Toc69722553)

[Выполнение задачи. 4](#_Toc69722554)

[Заключение. 19](#_Toc69722555)

# Описание задачи.

В рамках лабораторной работы необходимо реализовать генератор случайных графов, генератор должен содержать следующие параметры:

* Максимальное/Минимальное количество генерируемых вершин
* Максимальное/Минимальное количество генерируемых ребер
* Максимальное количество ребер связанных с одной вершины
* Генерируется ли направленный граф
* Максимальное количество входящих и выходящих ребер
* Ребра графа содержат веса
* Максимальное/Минимальное значения веса
* Могут ли существовать в графе недоступные области

Сгенерированный граф должен быть описан в рамках одного класса (этот класс не должен заниматься генерацией), и должен обладать обязательно следующими методами:

* Выдача матрицы смежности
* Выдача матрицы инцидентности
* Выдача список смежности
* Выдача списка ребер

Поиск длиннейшего пути в графе от точки А до точки Б

# Описание структуры.

***Граф*** – совокупность точек, соединенных линиями. Точки называются ***вершинами***, или ***узлами***, а линии – ребрами, или дугами.

***Степень входа вершины*** – количество входящих в нее ребер, ***степень выхода*** – количество исходящих ребер.  
Граф, содержащий ребра между всеми парами вершин, является ***полным***  
Встречаются такие графы, ребрам которых поставлено в соответствие конкретное числовое значение, они называются ***взвешенными графами***, а это значение – ***весом ребра***.  
Когда у ребра оба конца совпадают, т.е. оно выходит из вершины и входит в нее, то такое ребро называется ***петлей***.

Графы делятся на

* связные
* несвязные

В связном графе между любой парой вершин существует как минимум один путь.  
В несвязном графе существует хотя бы одна вершина, не связанная с другими  
Графы также подразделяются на

* ориентированные
* неориентированные
* смешанные.

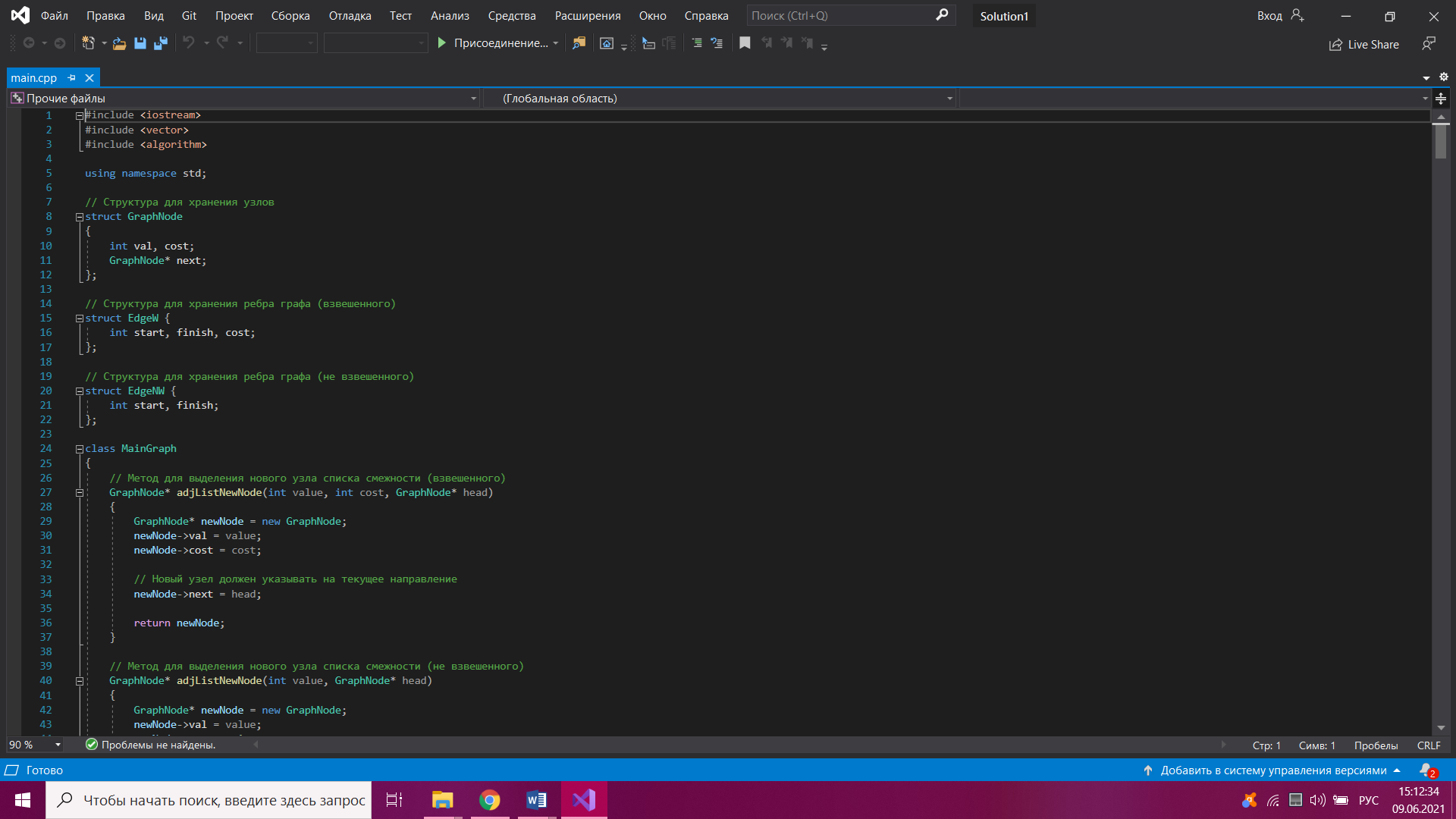
В ориентированном графе ребра являются направленными, т.е. существует только одно доступное направление между двумя связными вершинами.  
В неориентированном графе по каждому из ребер можно осуществлять переход в обоих направлениях.  
Частный случай двух этих видов – смешанный граф. Он характерен наличием как ориентированных, так и неориентированных ребер.

Граф может быть представлен (сохранен) несколькими способами:

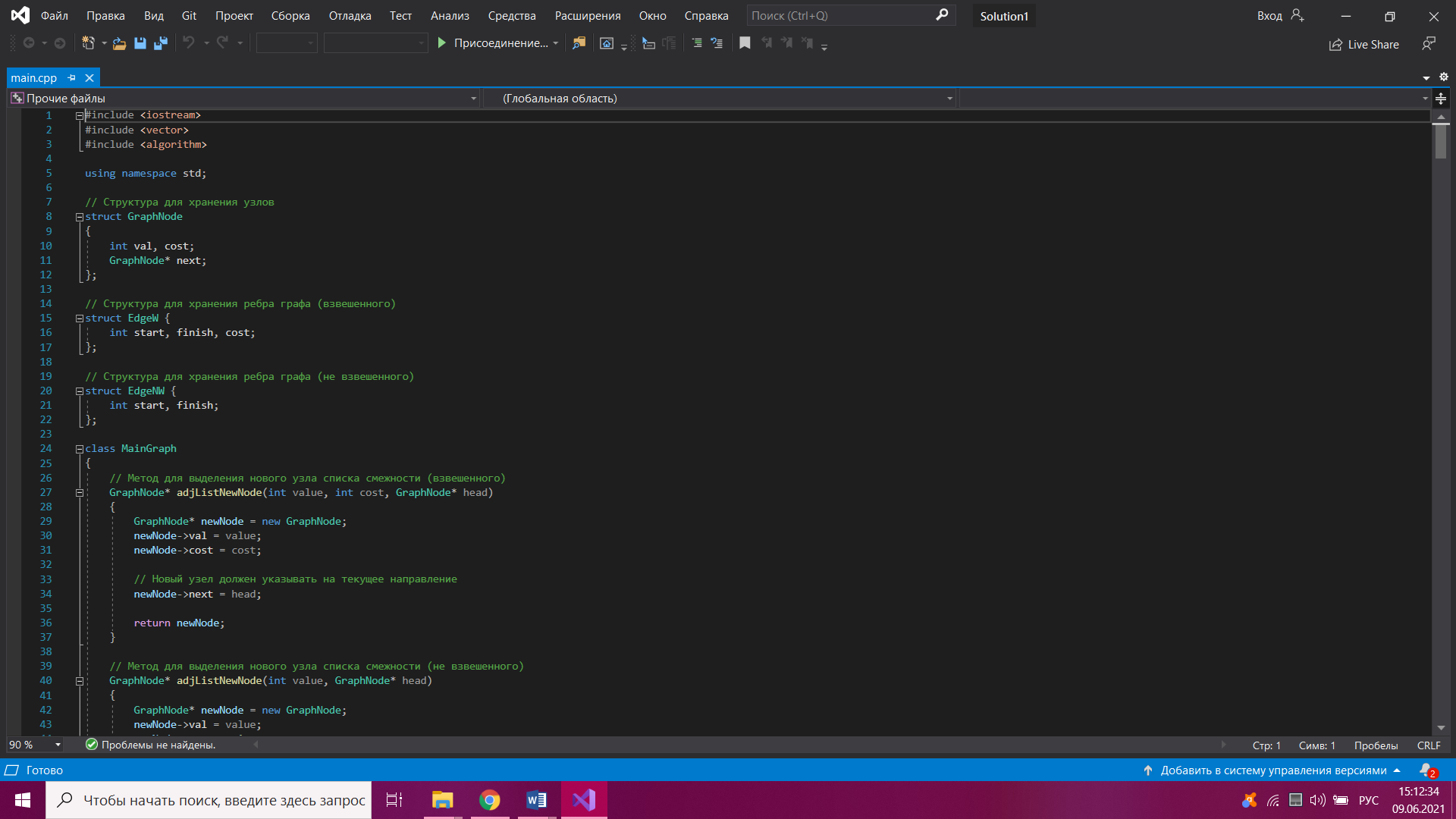
* матрица смежности;
* матрица инцидентности;
* список смежности (инцидентности);
* список ребер.

# Выполнение задачи.

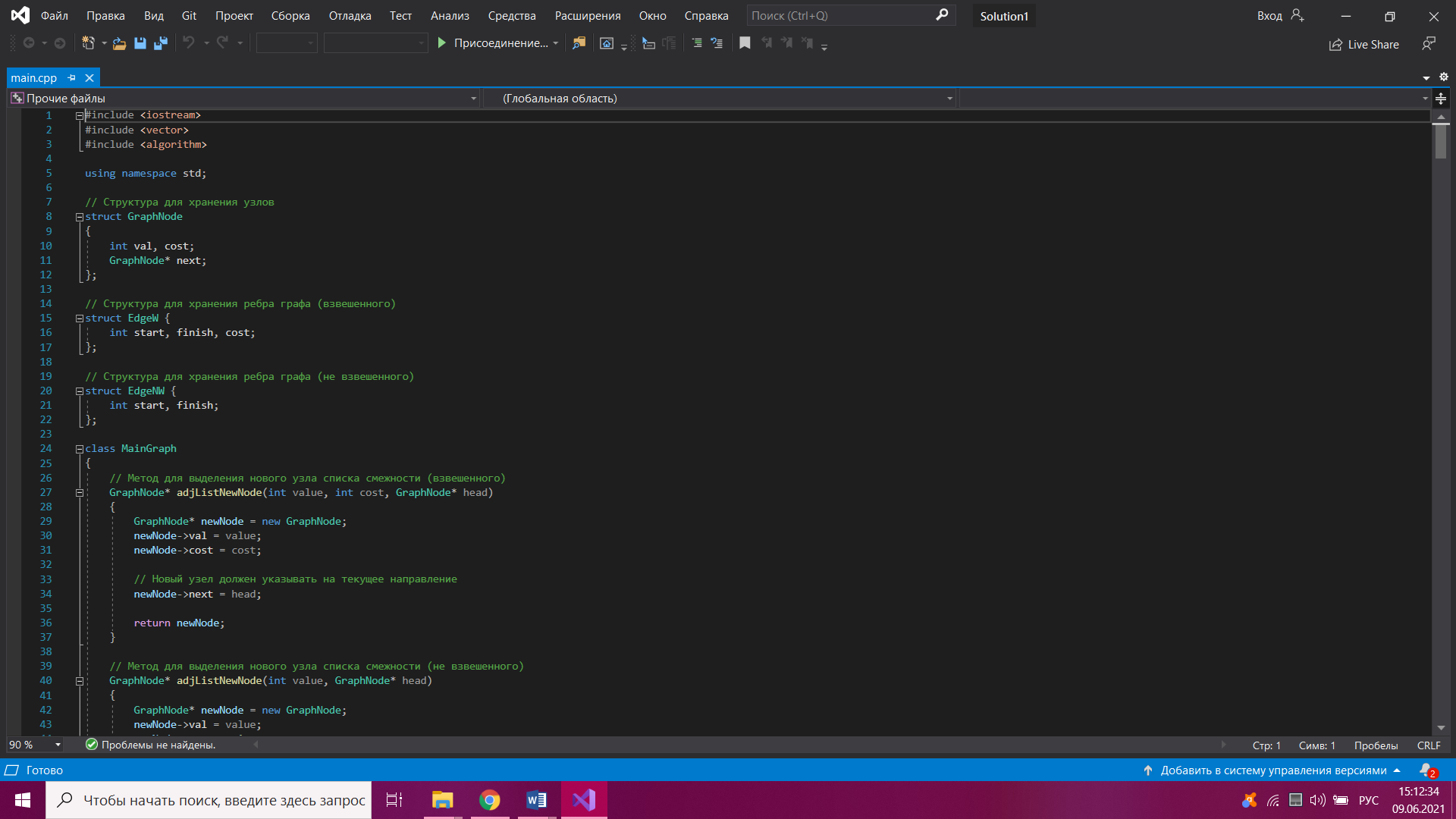
Структура для хранения узлов



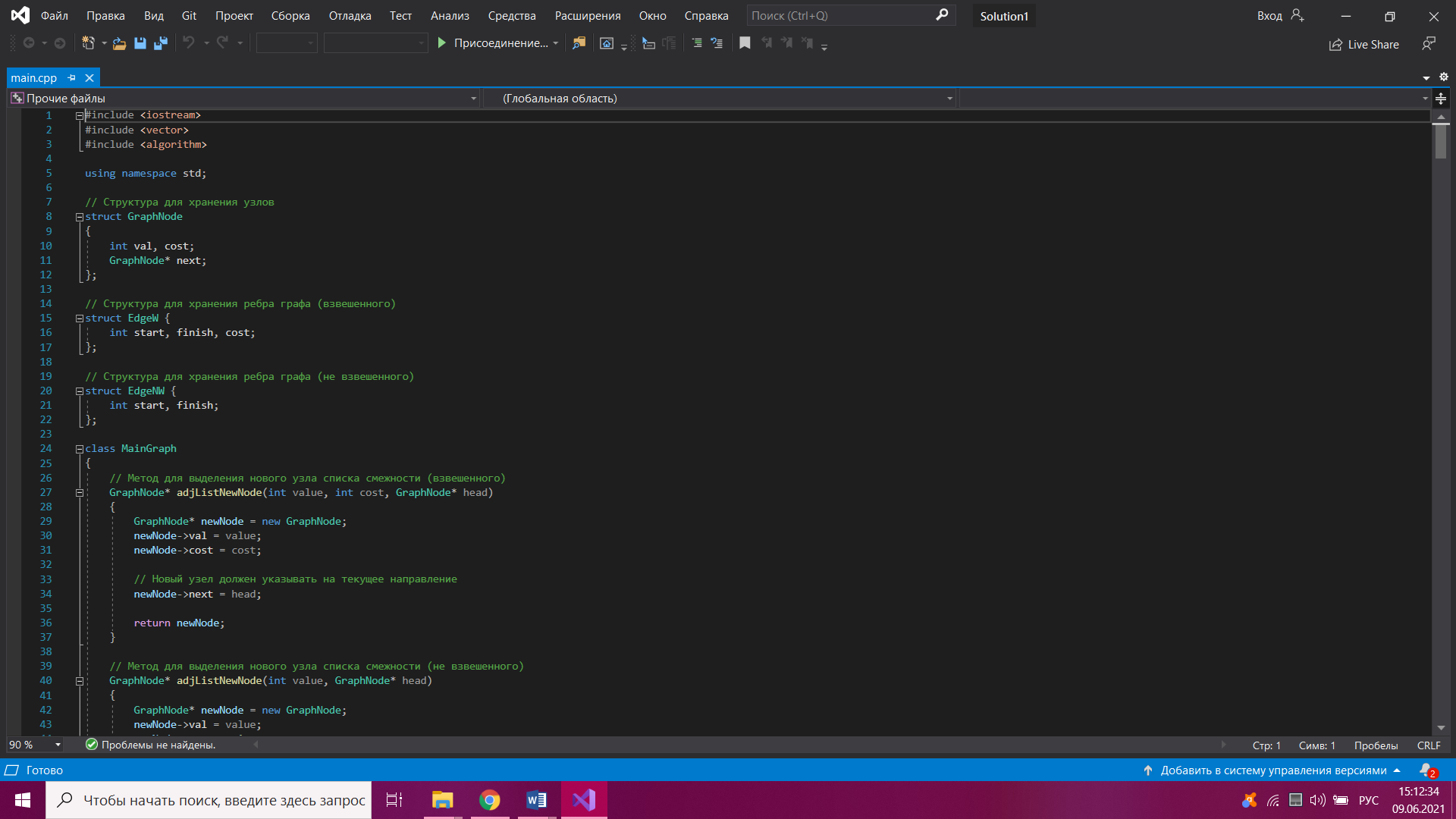
Структура для хранения ребра графа (взвешенного)

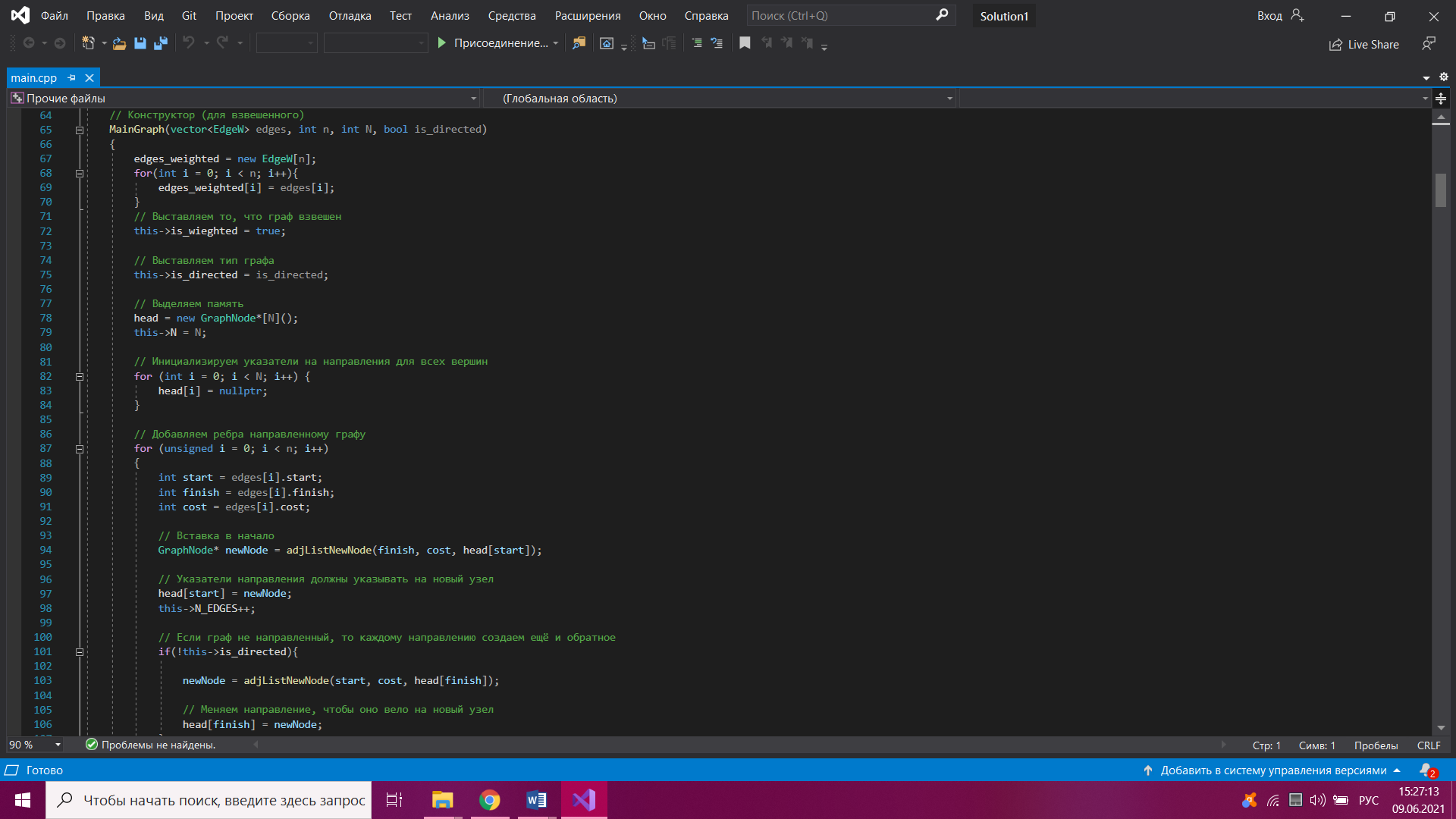


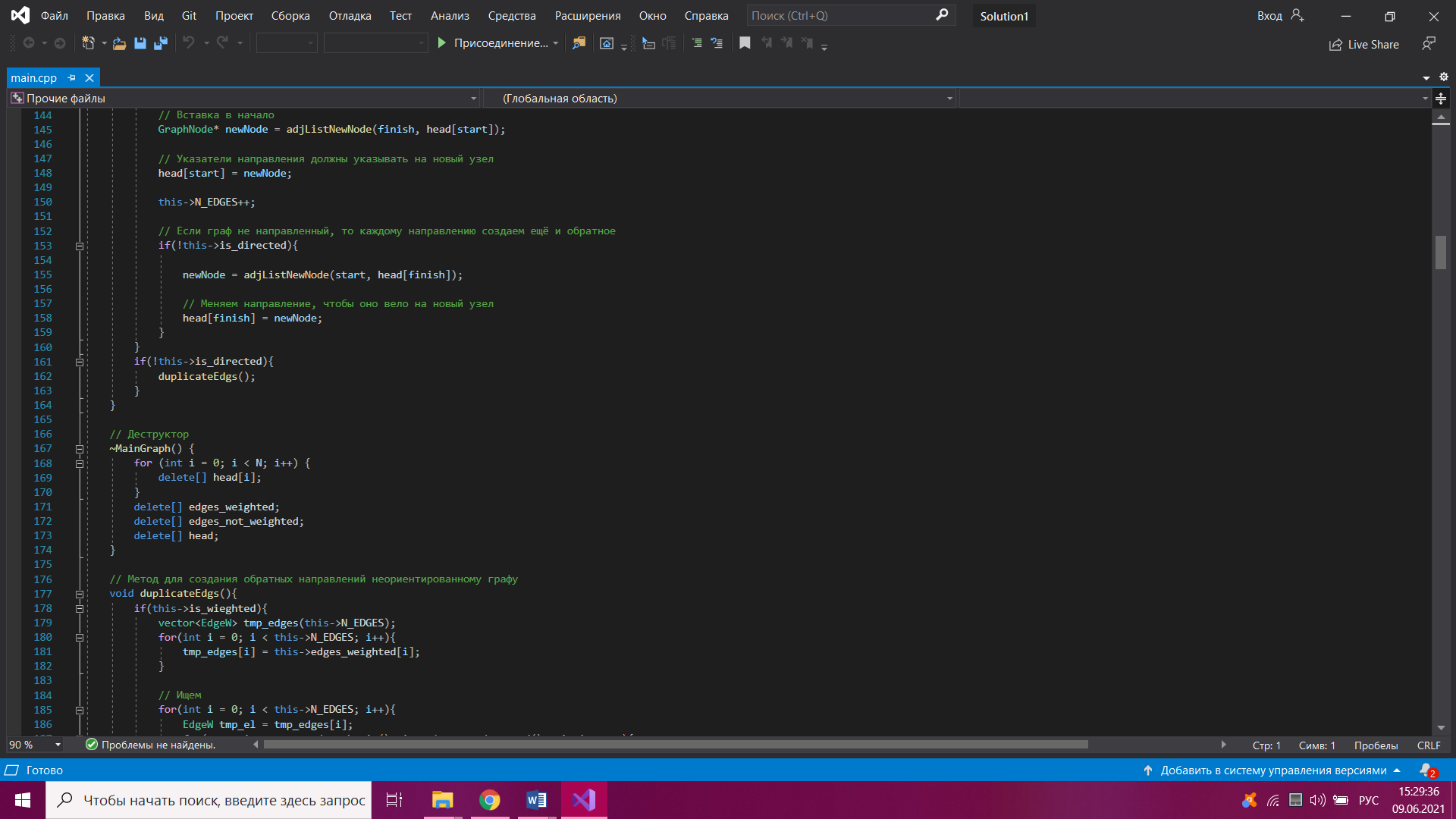
Структура для хранения ребра графа (не взвешенного)



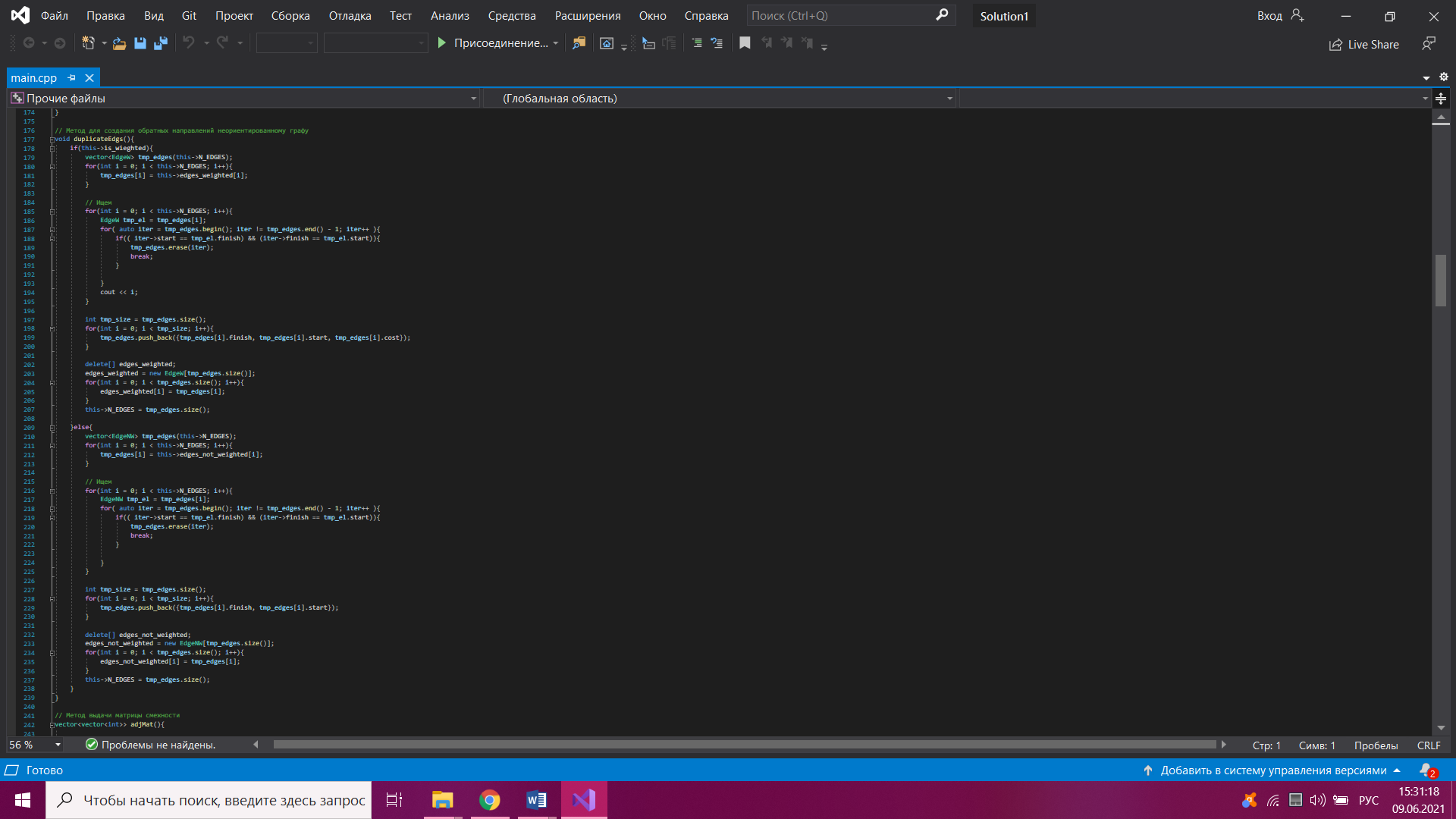
В этом классе ещё несколько свойств:



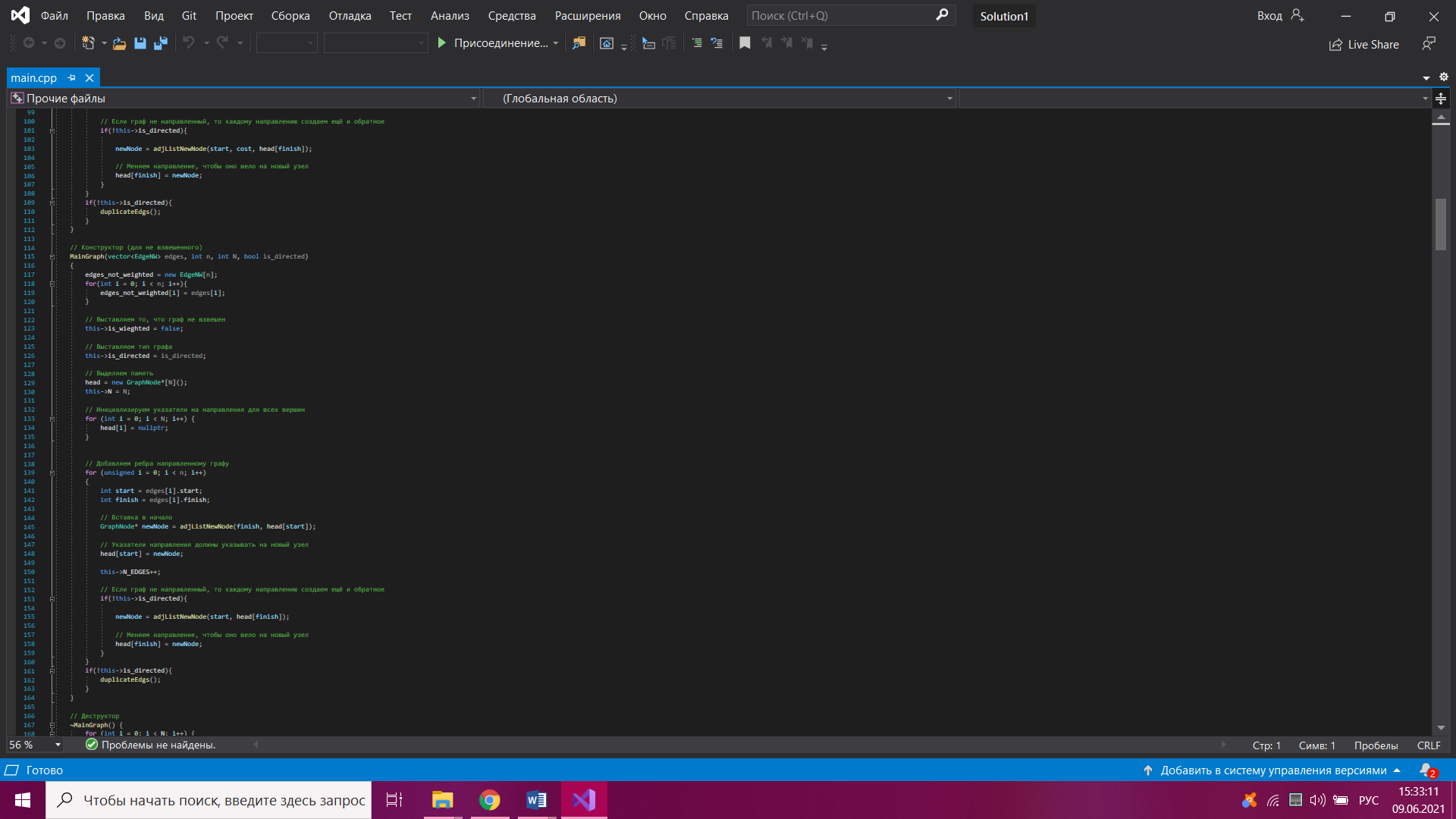


 Метод для создания обратных направлений неориентированному графу

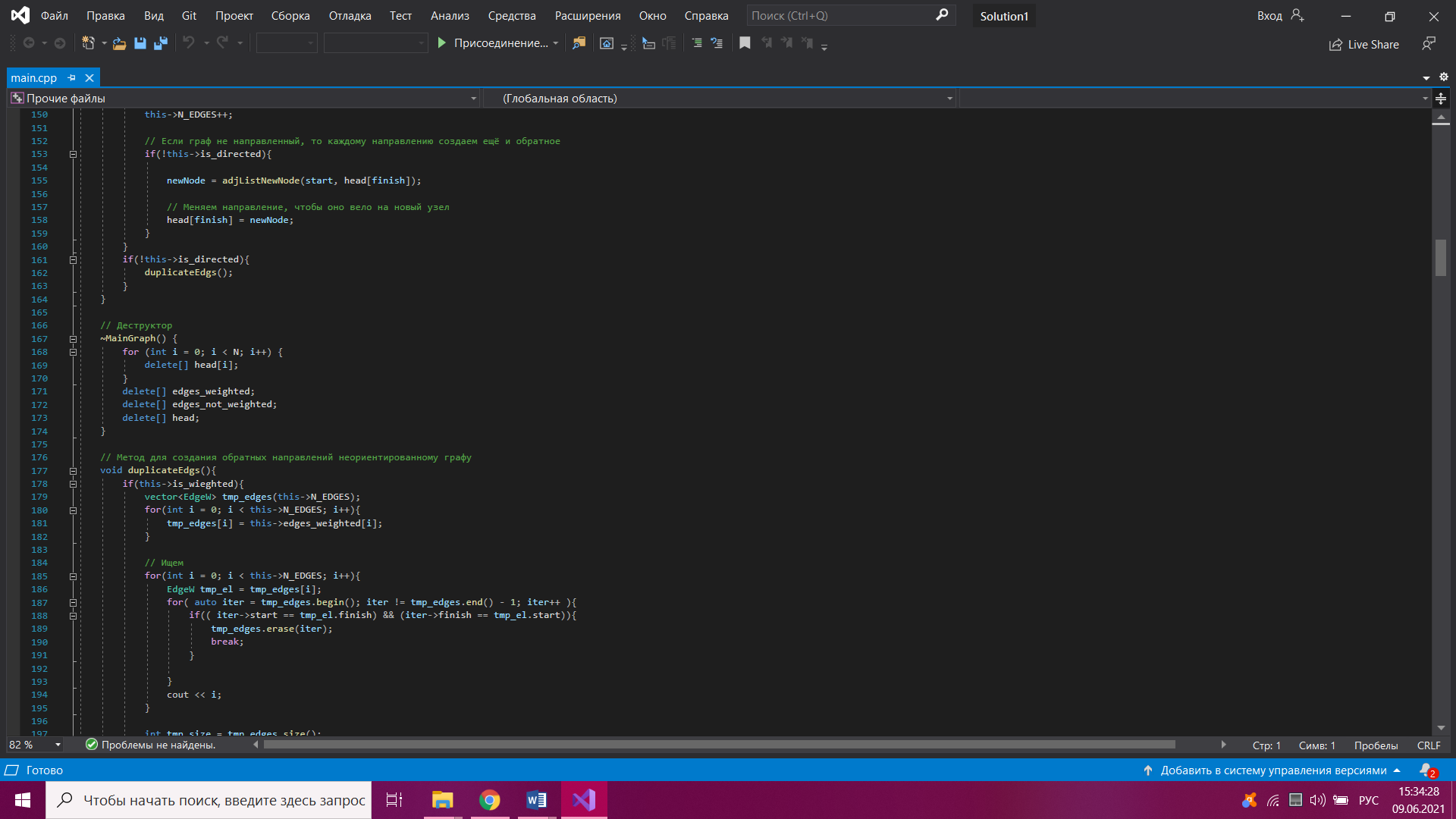
    void duplicateEdgs(){



Конструктор (для не взвешенного)

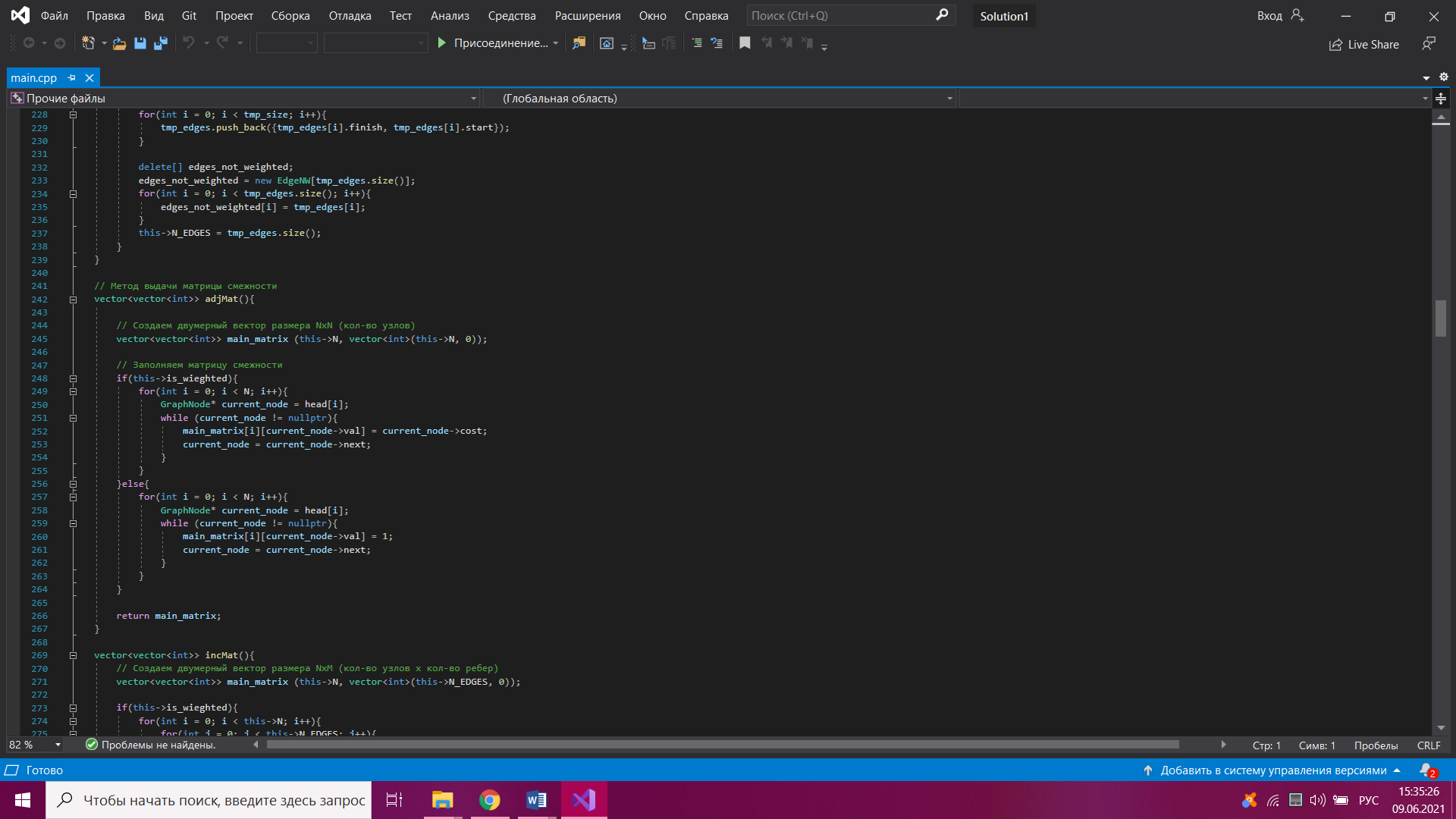


Деструктор

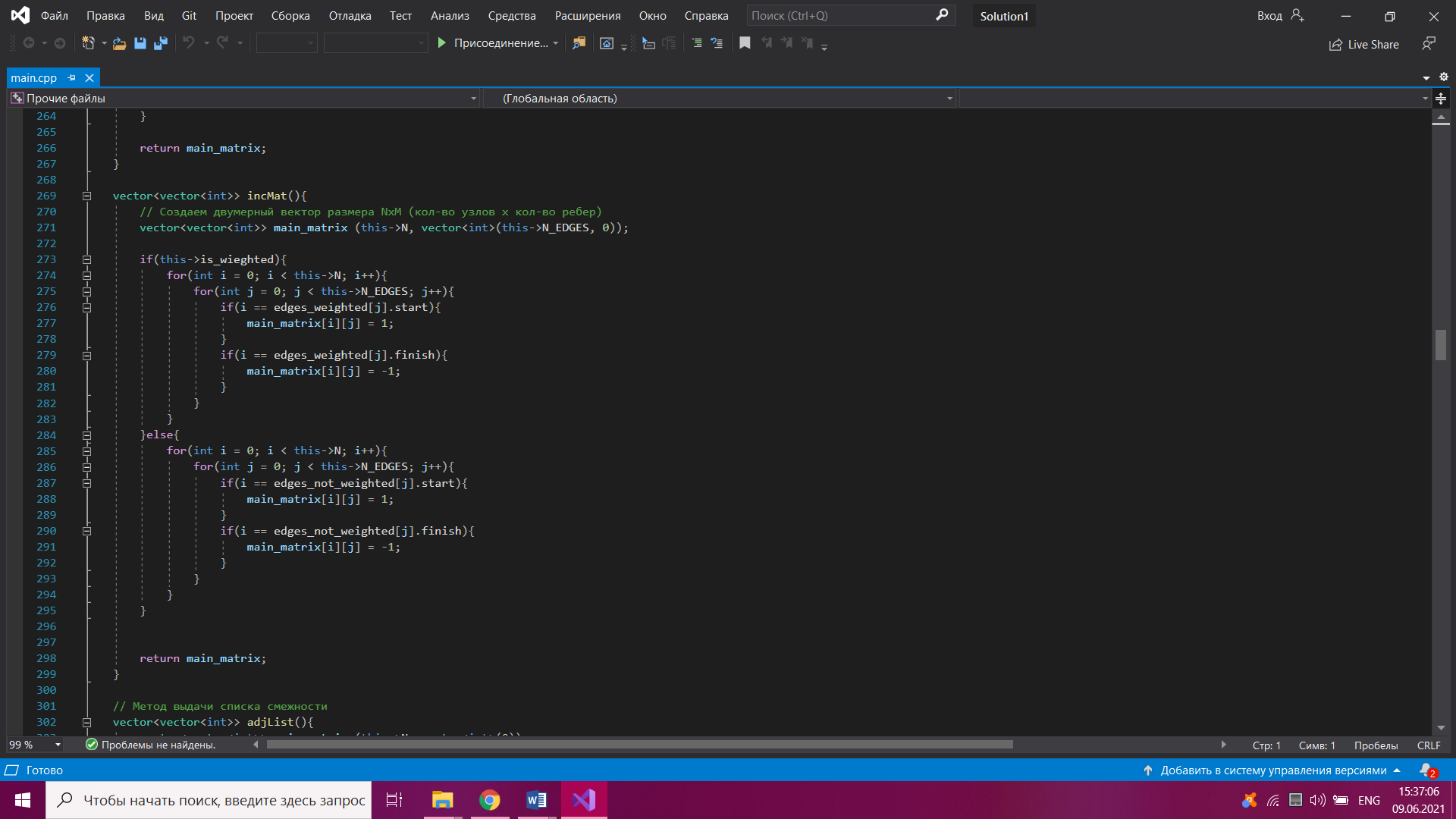


Обязательные методы:

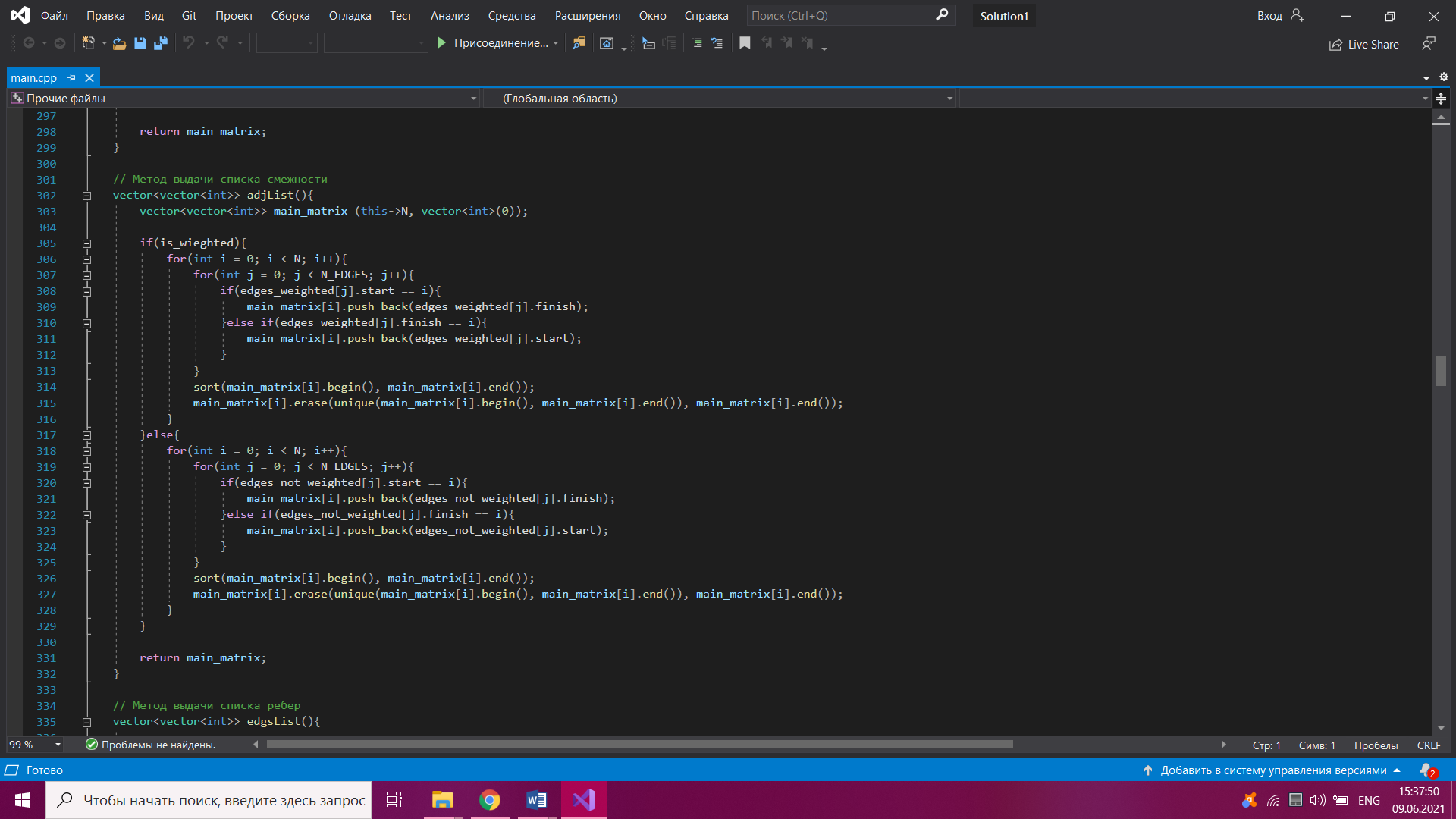
* + 1. **Матрица смежности-** это квадратная матрица, в которой каждый элемент принимает одно из двух значений: 0 или 1.



* + 1. **Матрица инцидентности-** это матрица, количество строк в которой соответствует числу вершин, а количество столбцов – числу рёбер. В ней указываются связи между инцидентными элементами графа (ребро(дуга) и вершина).

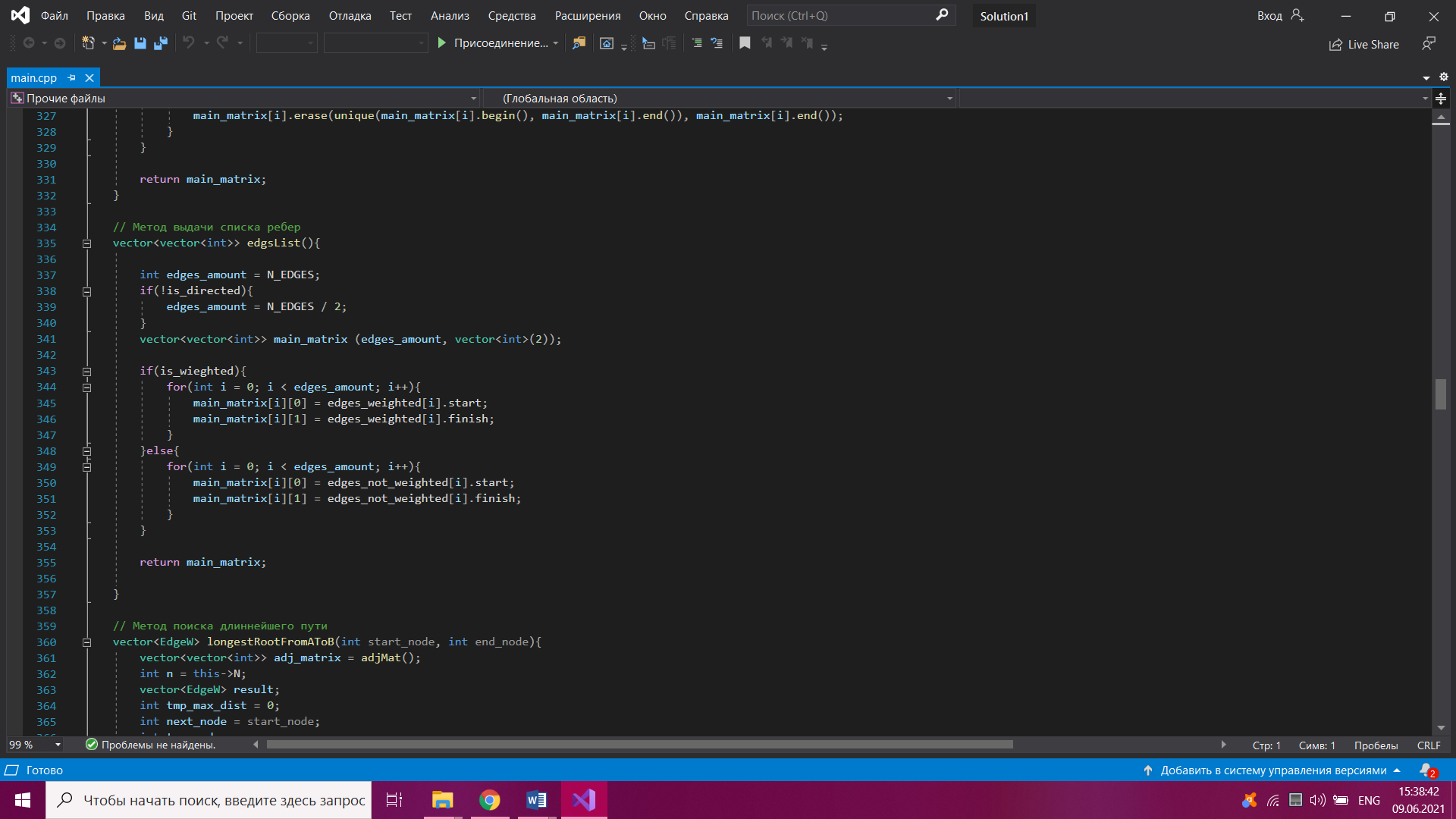


* + 1. **Список смежности:**– Если количество ребер графа по сравнению с количеством вершин невелико, то значения большинства элементов матрицы смежности будут равны 0.



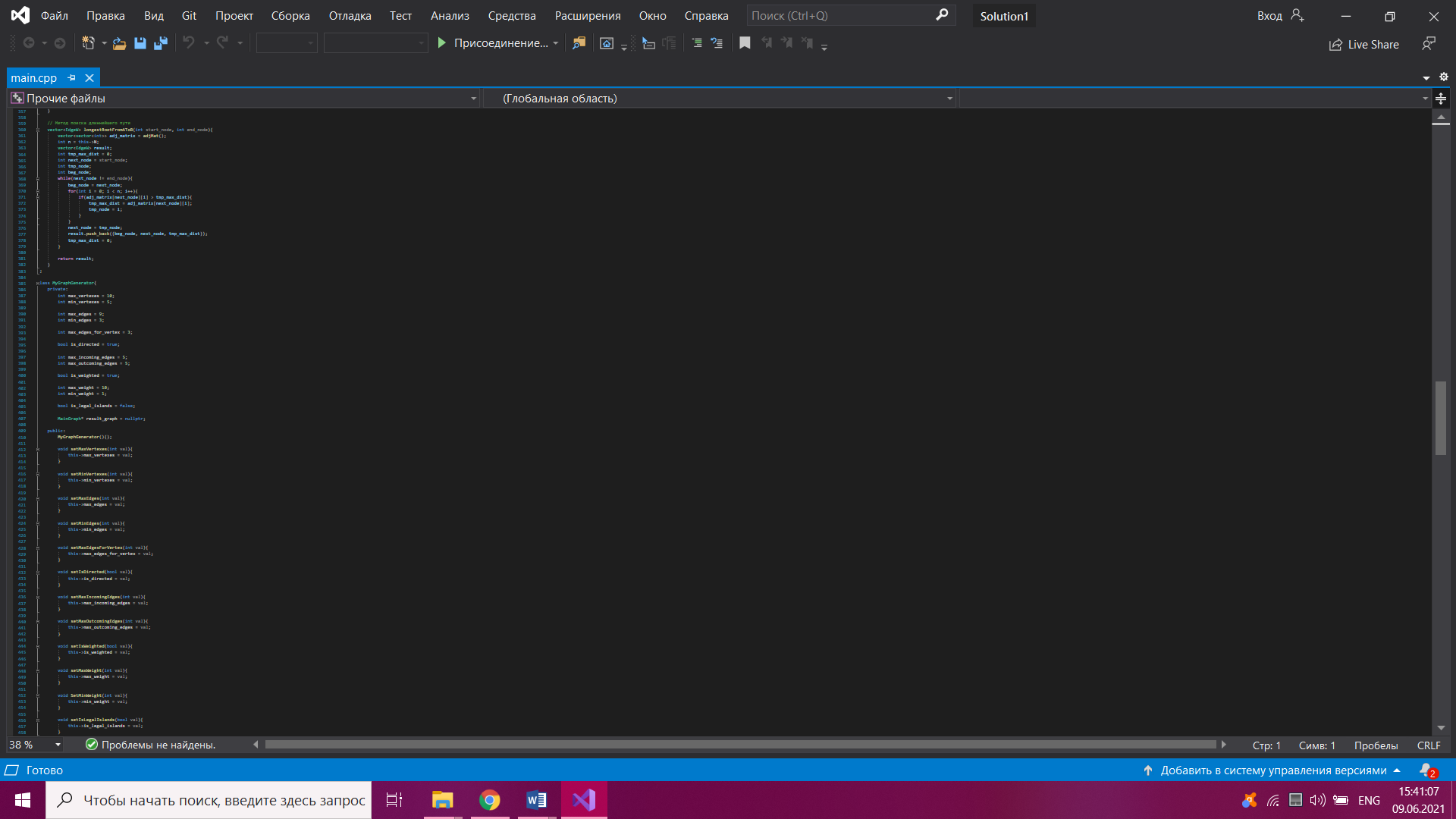
* + 1. **Список ребер.** В списке рёбер в каждой строке записываются две смежные вершины и вес соединяющего их ребра (для взвешенного графа).

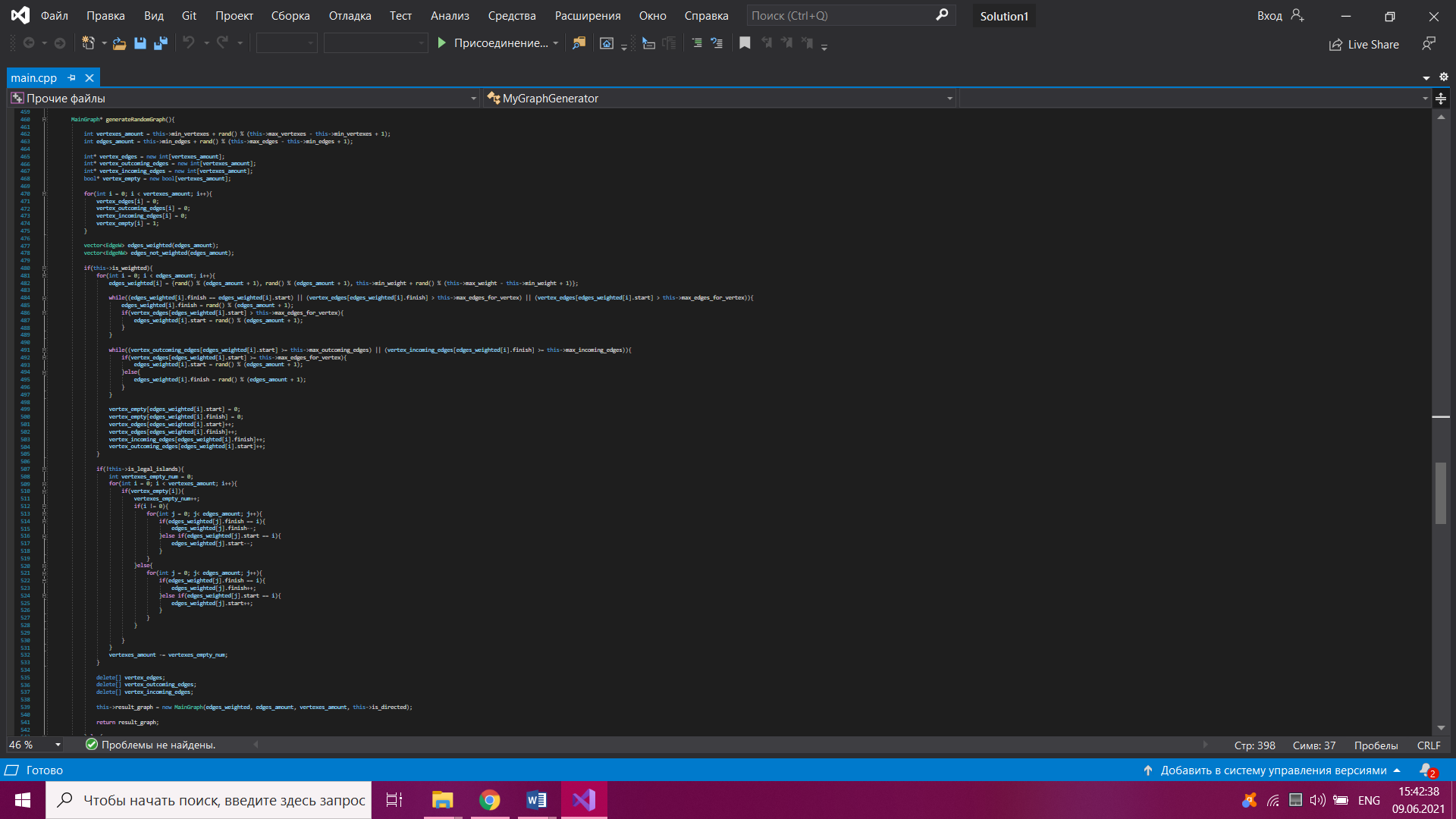
Количество строк в списке ребер всегда должно быть равно величине, получающейся в результате сложения ориентированных рёбер с удвоенным количеством неориентированных рёбер.

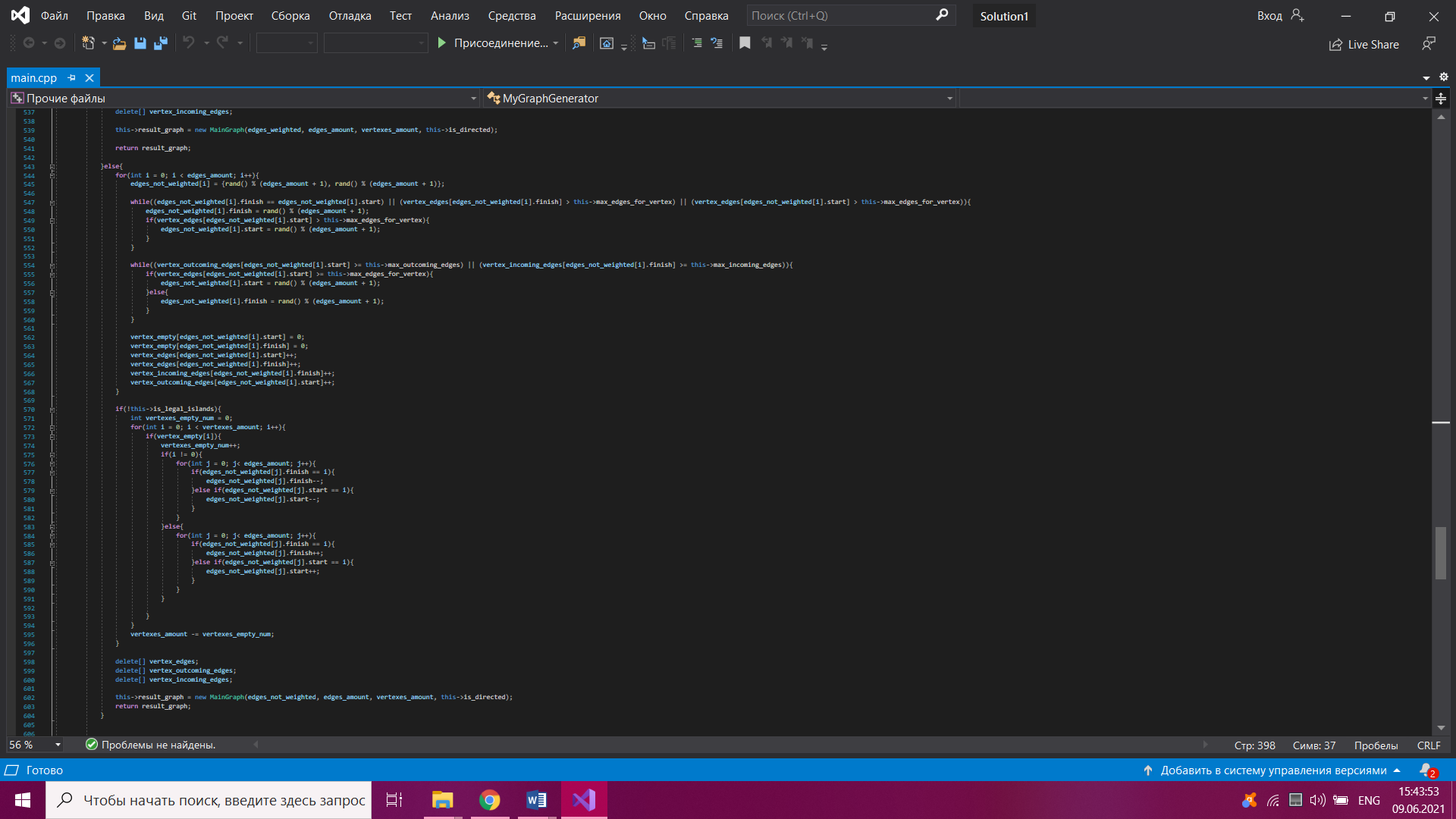


**Задание по варианту. Поиск длиннейшего пути из А в B:**

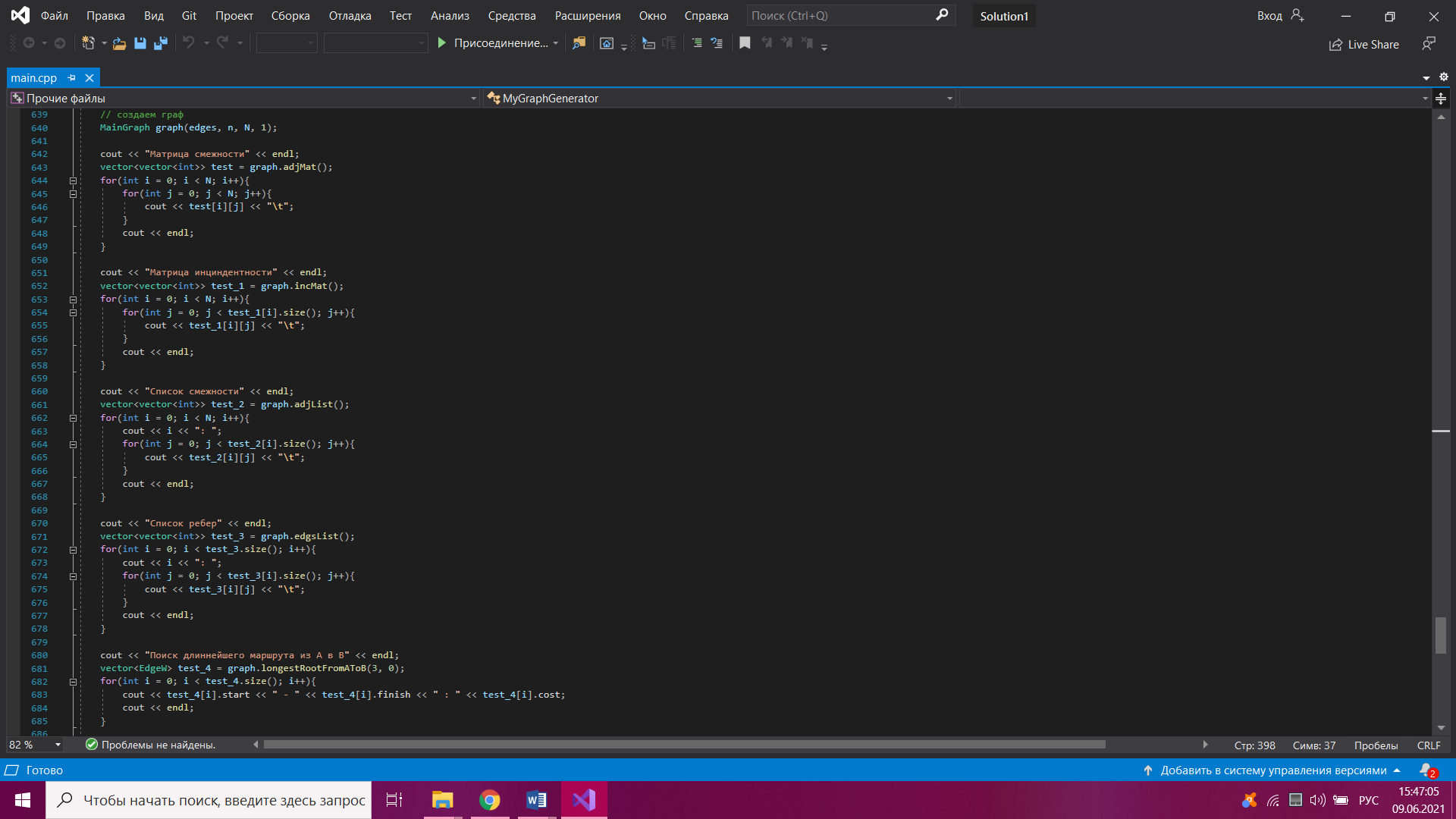
это задача поиска простого пути максимальной длины в заданном графе. Путь называется простым, если в нём нет повторных вершин. Длина пути может быть измерена либо числом рёбер, либо (в случае взвешенных графов) суммой весов его рёбер. В отличие от задачи кратчайшего пути, которая может быть решена за полиномиальное время на графах без циклов с отрицательным весом, задача нахождения самого длинного пути является NP-трудной и не может быть решена за полиномиальное время для произвольных графов, если только не P = NP. Более сильная трудность также известна, показывающая, что задачу трудно аппроксимировать. Однако задача решается за линейное время на ориентированных ациклических графах, которые имеют важное применение в задачах нахождения критического пути в задачах планирования.

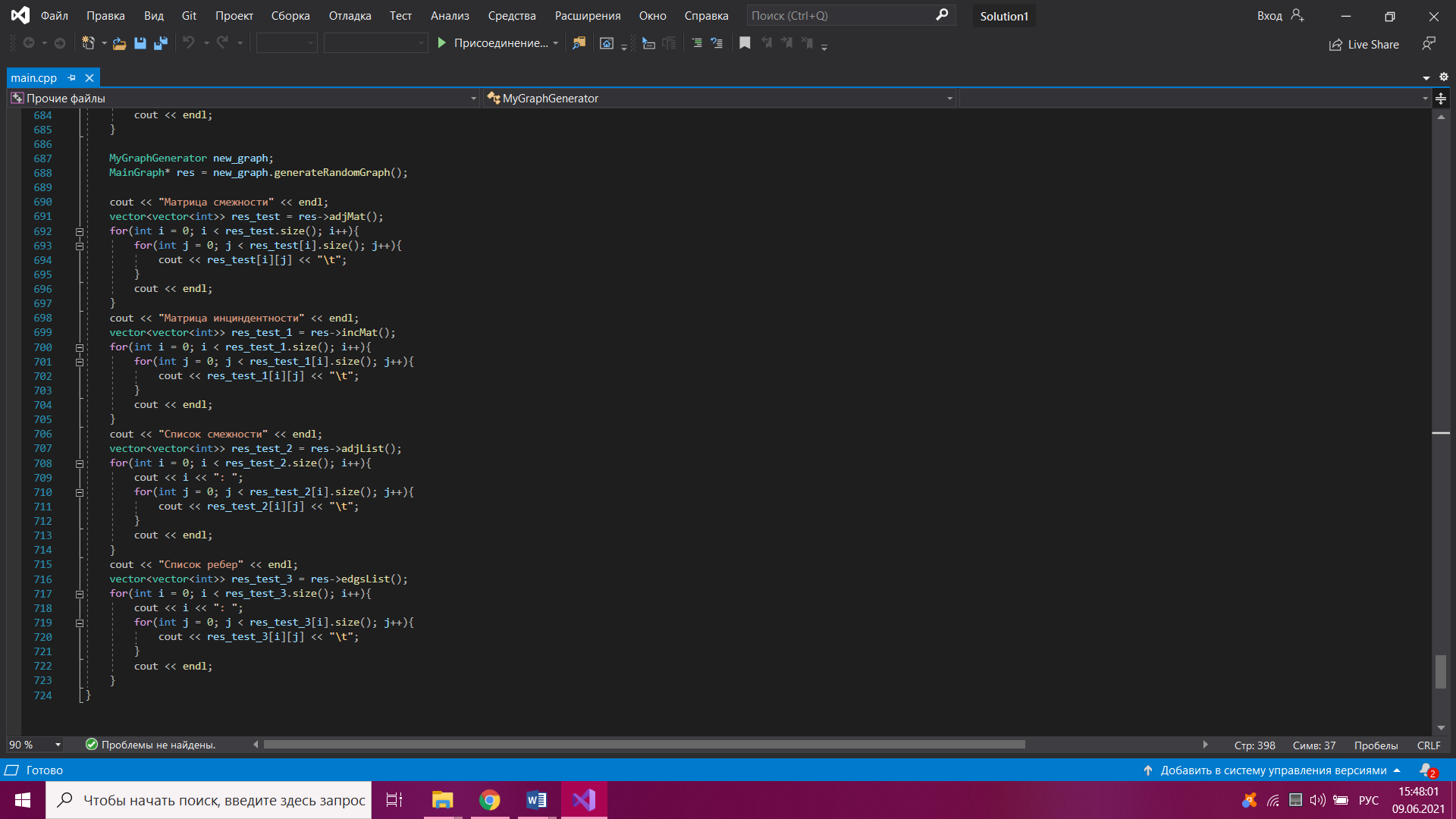






Тесты:





# Заключение.

В ходе данной лабораторной работы реализован генератор случайных графов, генератор, который содержит следующие параметры:

* Максимальное/Минимальное количество генерируемых вершин
* Максимальное/Минимальное количество генерируемых ребер
* Максимальное количество ребер связанных с одной вершины
* Генерируется ли направленный граф
* Максимальное количество входящий и выходящих ребер
* Ребра графа содержат веса
* Максимальное/Минимальное значения веса
* Могут ли существовать в графе недоступные области

Сгенерированный граф описан следующими методами:

* Выдача матрицы смежности
* Выдача матрицы инцидентности
* Выдача список смежности
* Выдача списка ребер

Был осуществлён поиск длиннейшего пути в графе от точки А до точки Б.