**Пояснительная записка к курсовой работе.**

**Тема: «Распределение очереди на два потока»**

Выполнила: Блохина Ангелина

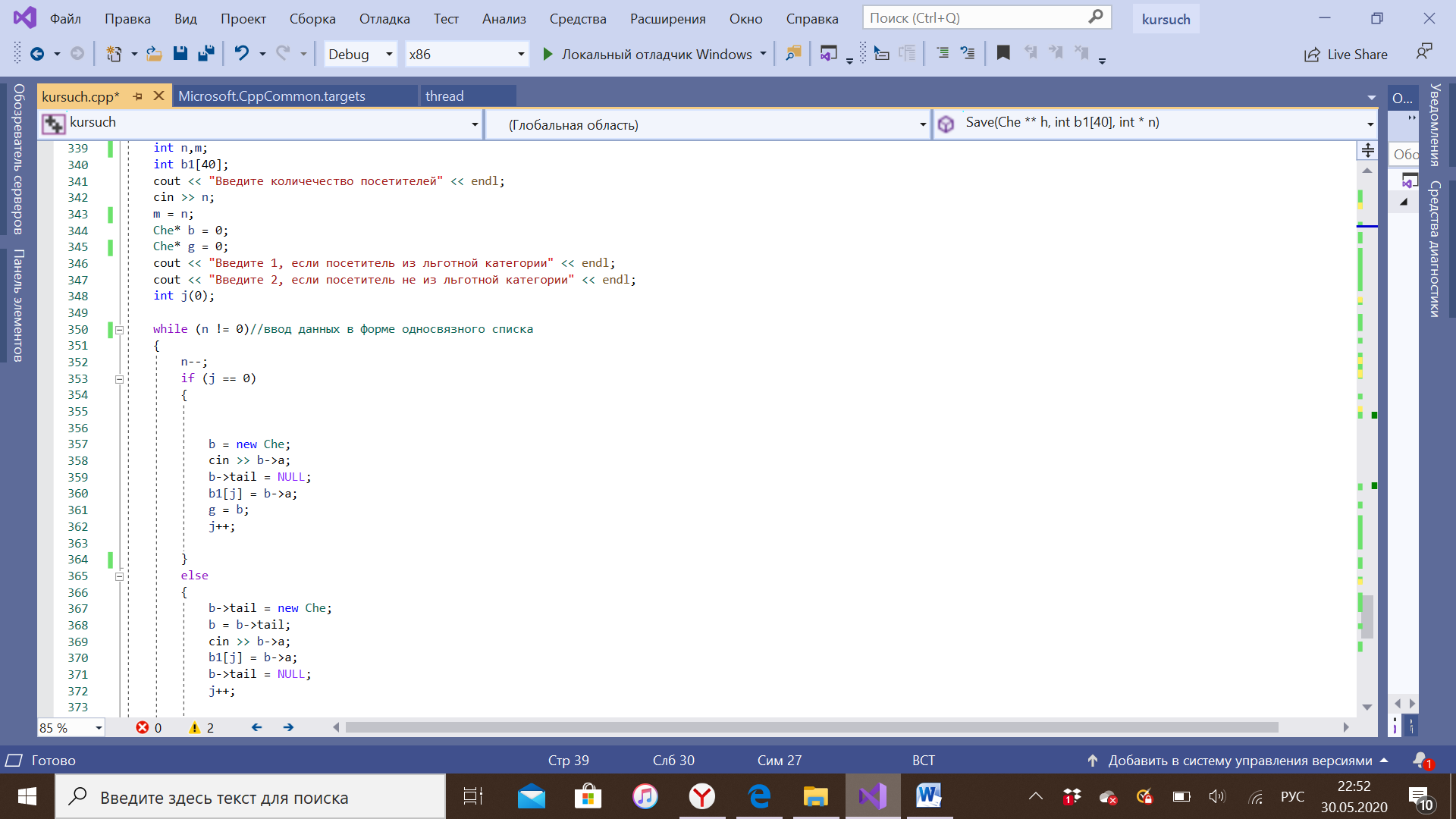
Группа: 9373

Цель работы:

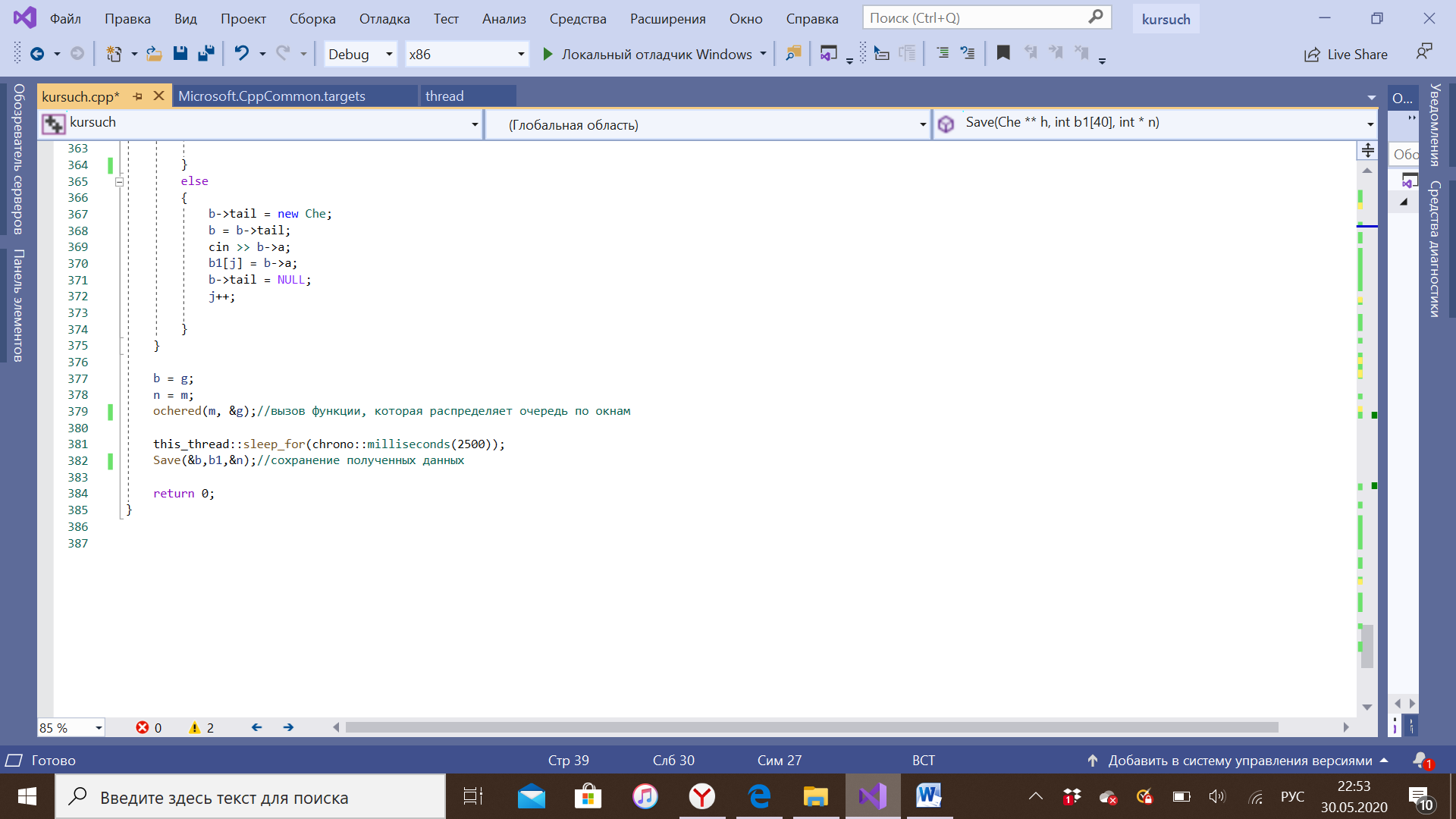
Распределить очередь по окнам в зависимости от порядка следования, учитывая время оформления документа и категории посетителя (льготная/нельготная). Записать полученные результаты в файл. Вывести на кран запись из файла.

Алгоритм работы программы

Пользователь вводит количество посетителей(int n) и их категорию, на основе этих данных создается односвязный список. Одновременно массив(int b1[40]) заполняется получаемыми данными, чтобы в дальнейшем использовать в создание файла.

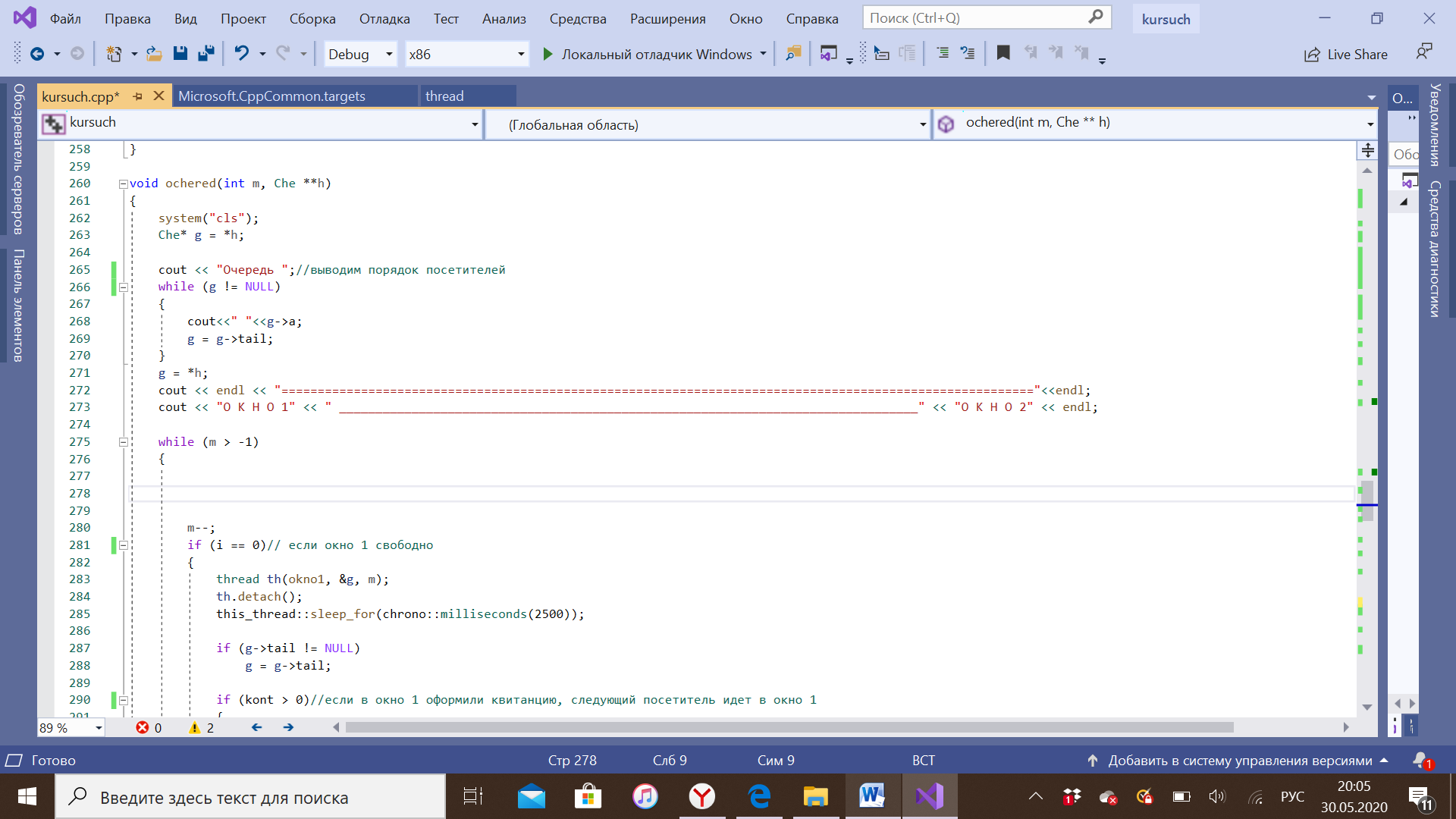


Вызываем функцию cohered(), передавая в нее данные о количестве посетителей и их категориях.



**Функция ochered().**

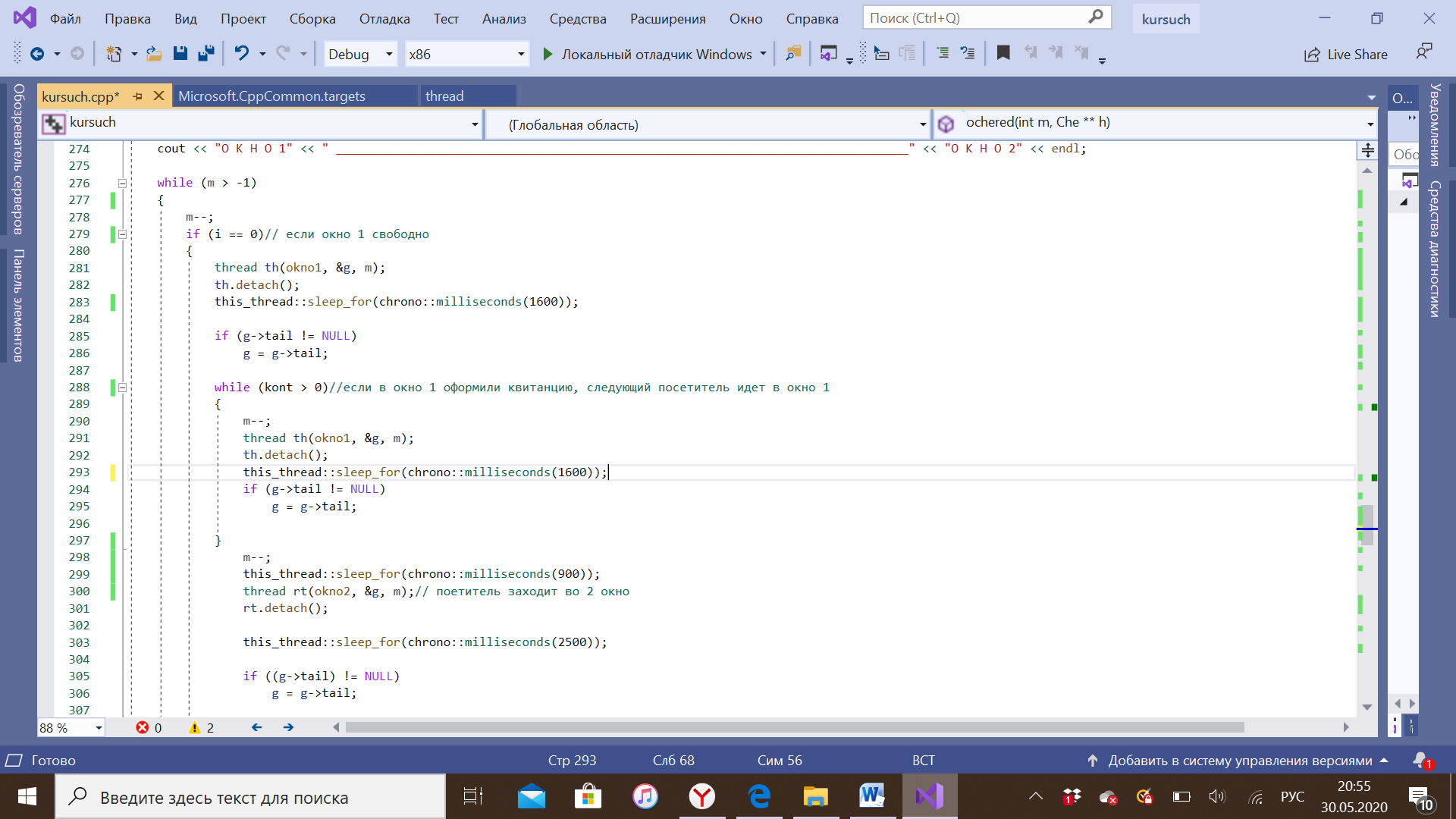
Выводим порядок посетителей, который ввел пользователь. Снова присваиваем переменной g значение первого элемента списка.



Распределение посетителей по окнам. Пока не закончатся посетители вополняем следующий алгоритм:

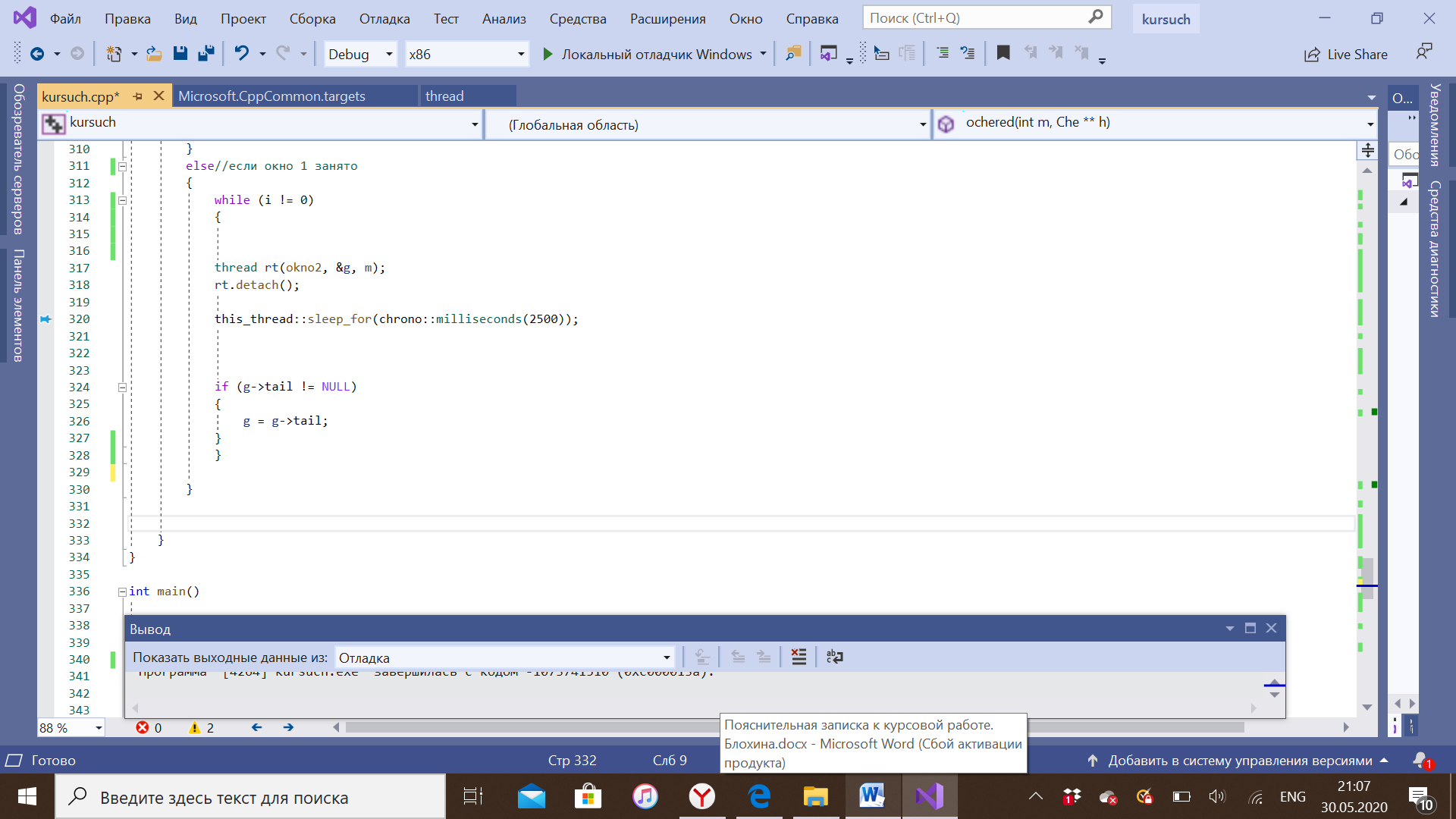
* если первое окно свободно:

1. туда заходит посетитель;
2. ожидает 1600 миллисекунд, пока выполняются действия в функции okno1();
3. переходит к следующему посетителю;
4. если в первом окне оформили квитанцию, то следующий посетитель идет в первое окно, так как время оформления квитанции меньше разницы времени ожидания первого окна посетителем (2500 миллисекунды), поэтому окно 1 становится свободно раньше;
5. если пункт 4 не выполняется, то посетитель идет в окно 2;
6. после чего ждет 2500 миллисекнд и переходит к следующему посетителю;



* если первое окно занято (работает функция договор, время выполнения которой больше, чем время между клиентами):

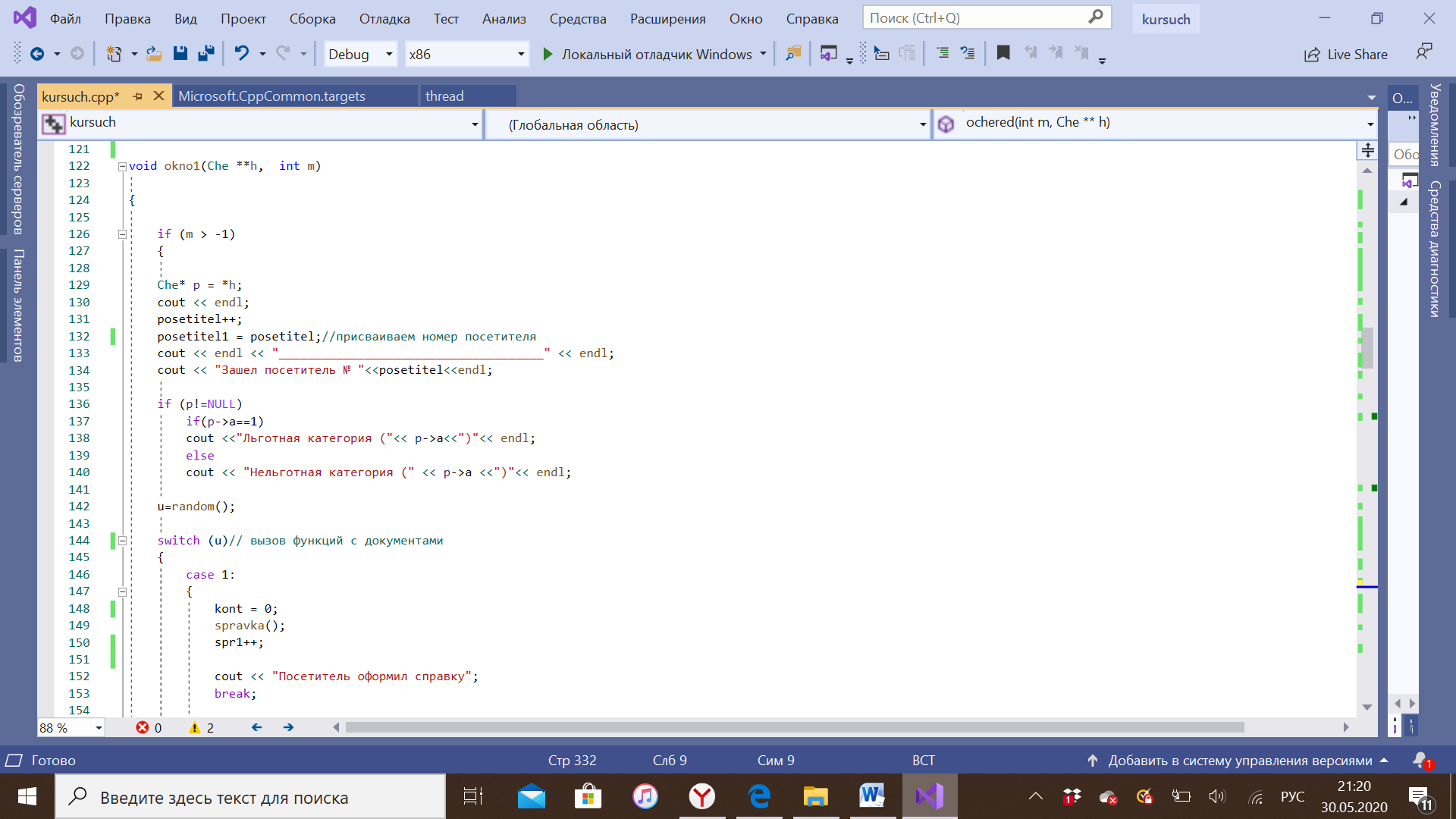
1. посетители идут во второе окно, пока не освободится первое



**Функция okno1()**

Принимает данные о категории клиента и количестве оставшихся клиентов.

* выводит на экран, какой по счету посетитель зашел;
* определяет, к какой категории относится посетитель, и выводит ее на экран;
* случайным образом определяет, что будет оформлять посетитель;



* если u=1:

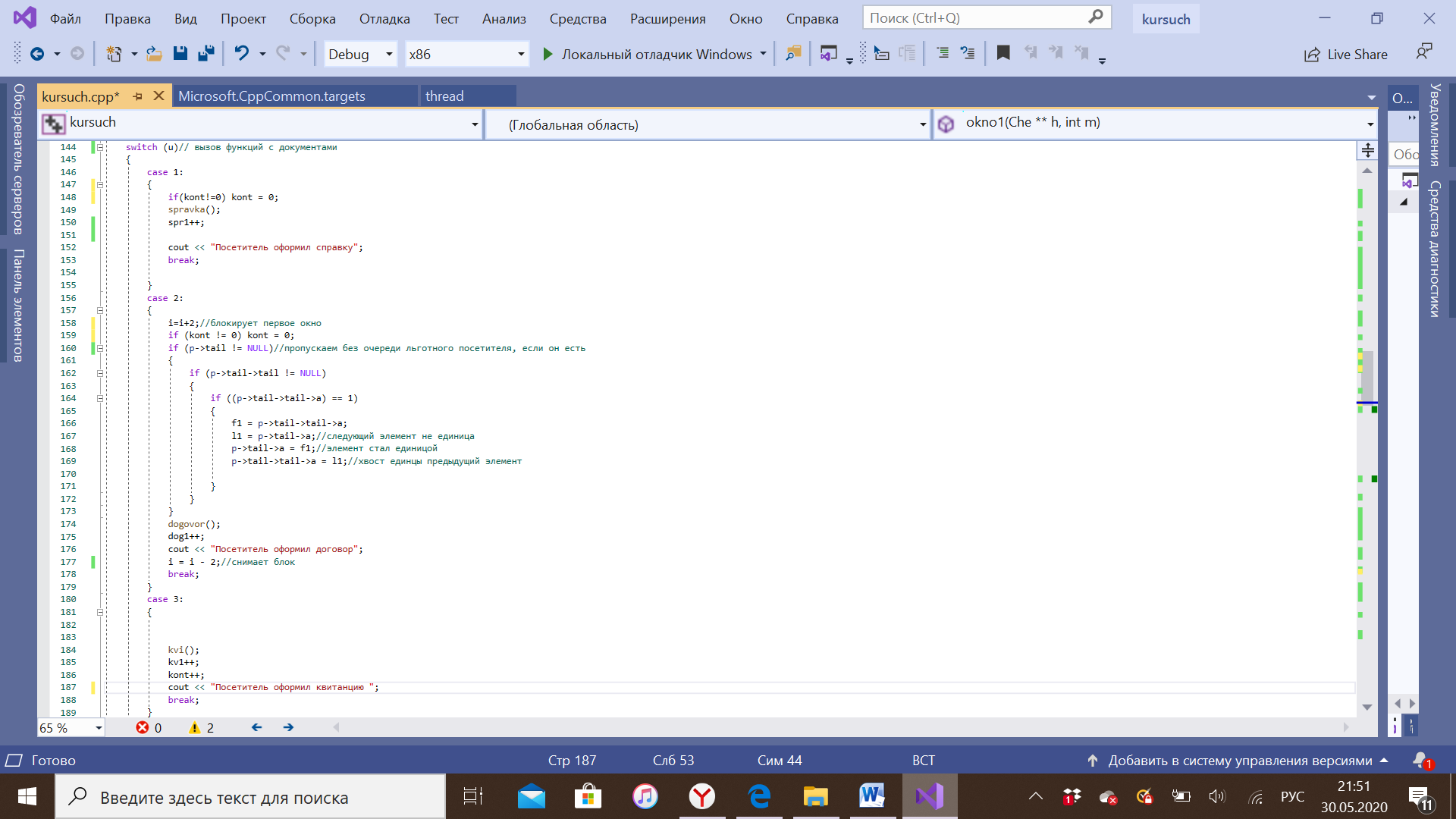
1. если в предыдущее посещение оформляли квитанцию, то обнуляем значение kont;
2. вызываем функцию spravka();
3. выводим на экран, что посетитель оформил справку;

* если u=2:

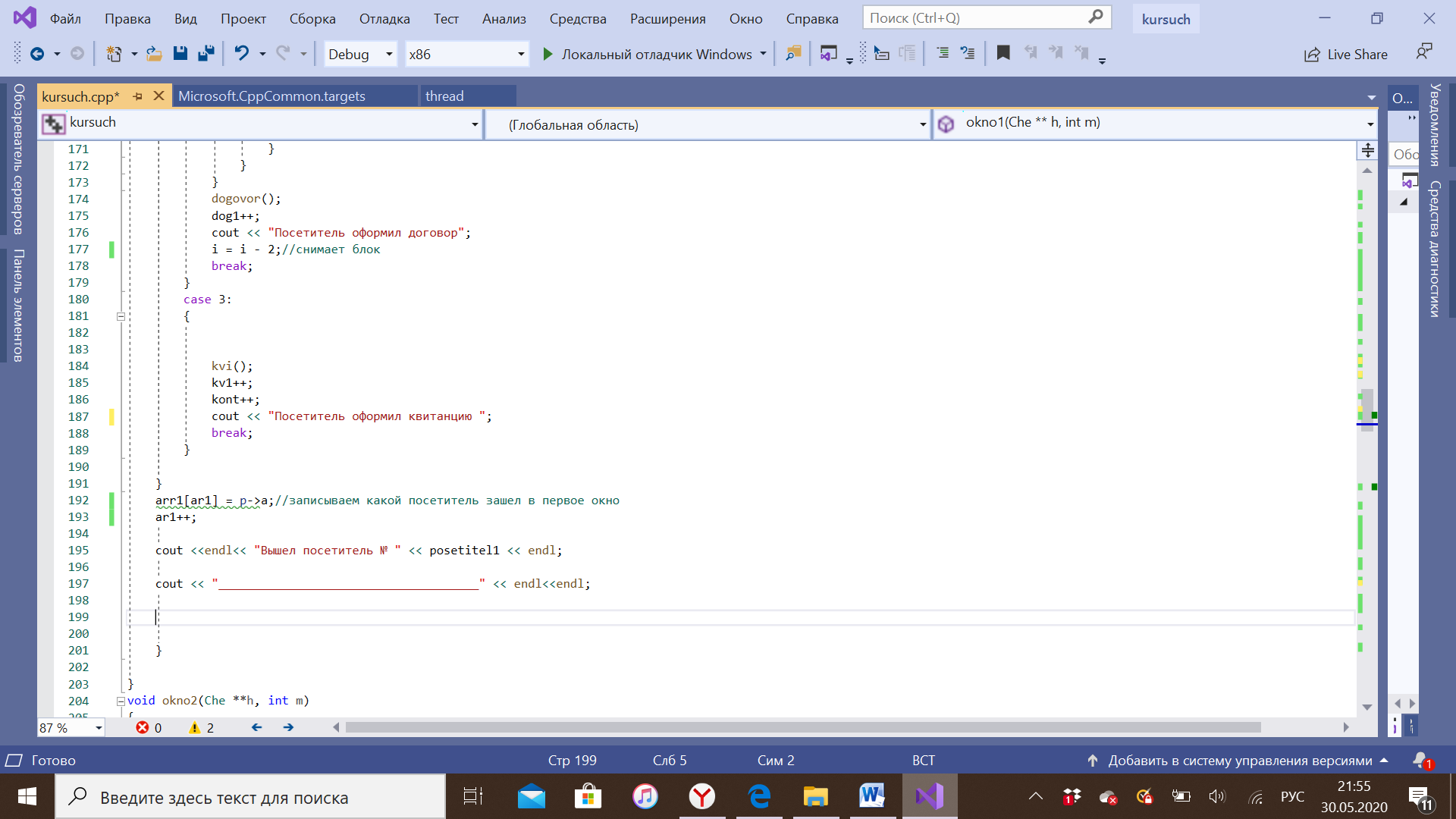
1. блокируем вход в окно 1;
2. если в предыдущее посещение оформляли квитанцию, то обнуляем значение kont;
3. если 2 посетитель от посетителя, который офрмляет договор относится к льготной категории, то пропускаем его без очереди в следующее окно;
4. вызываем функцию dogovor();
5. выводим на экран, что посетитель оформил договор;
6. открываем вход окно 1;

* если u=3:

1. вызываем функцию kvi();
2. увеличиваем переменную kont, чтобы следующий посетитель снова зашел в окно1;
3. выводим на экран, что посетитель оформил квитанцию;

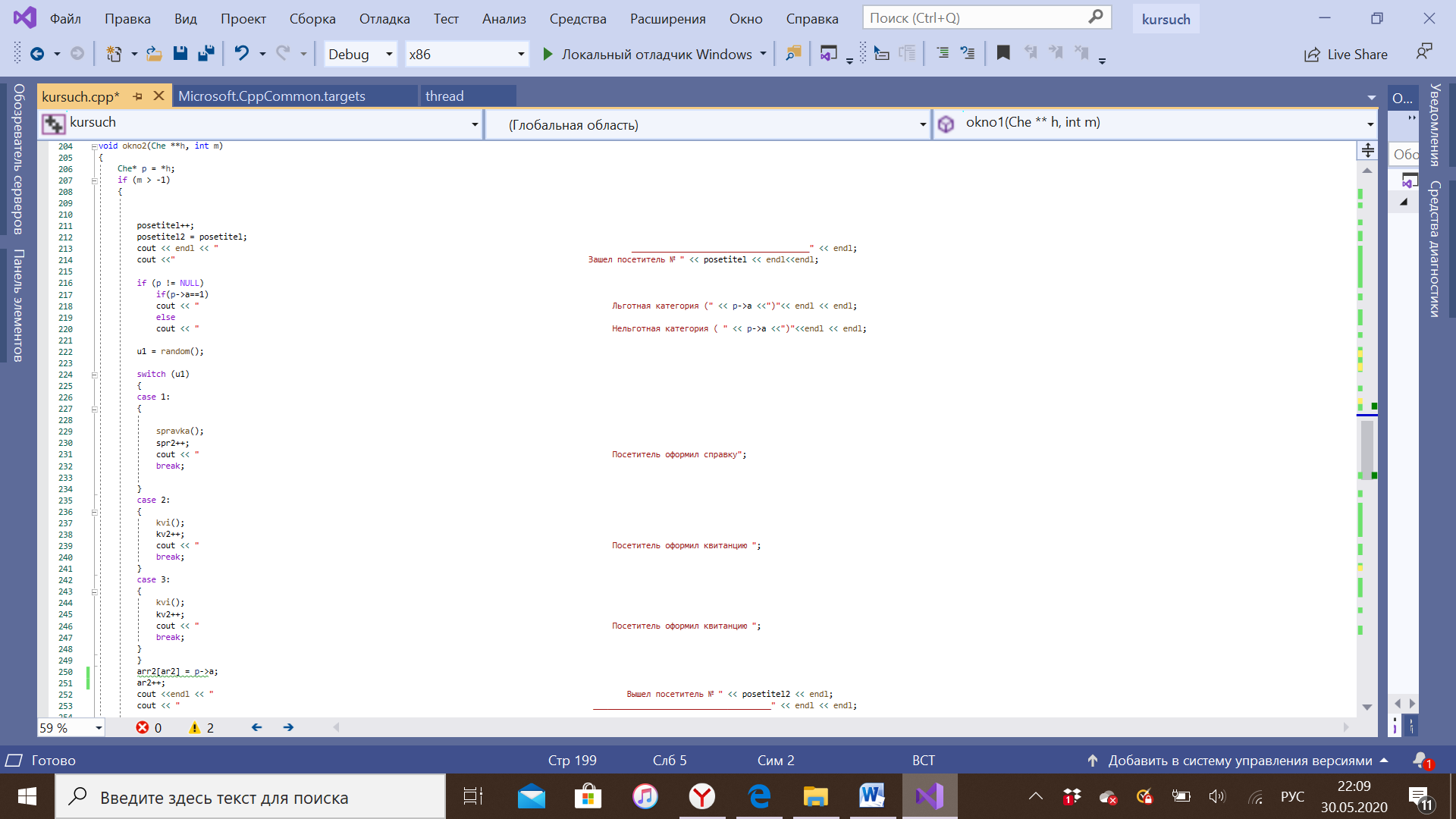


* записываем, какие посетители заходили в первое окно



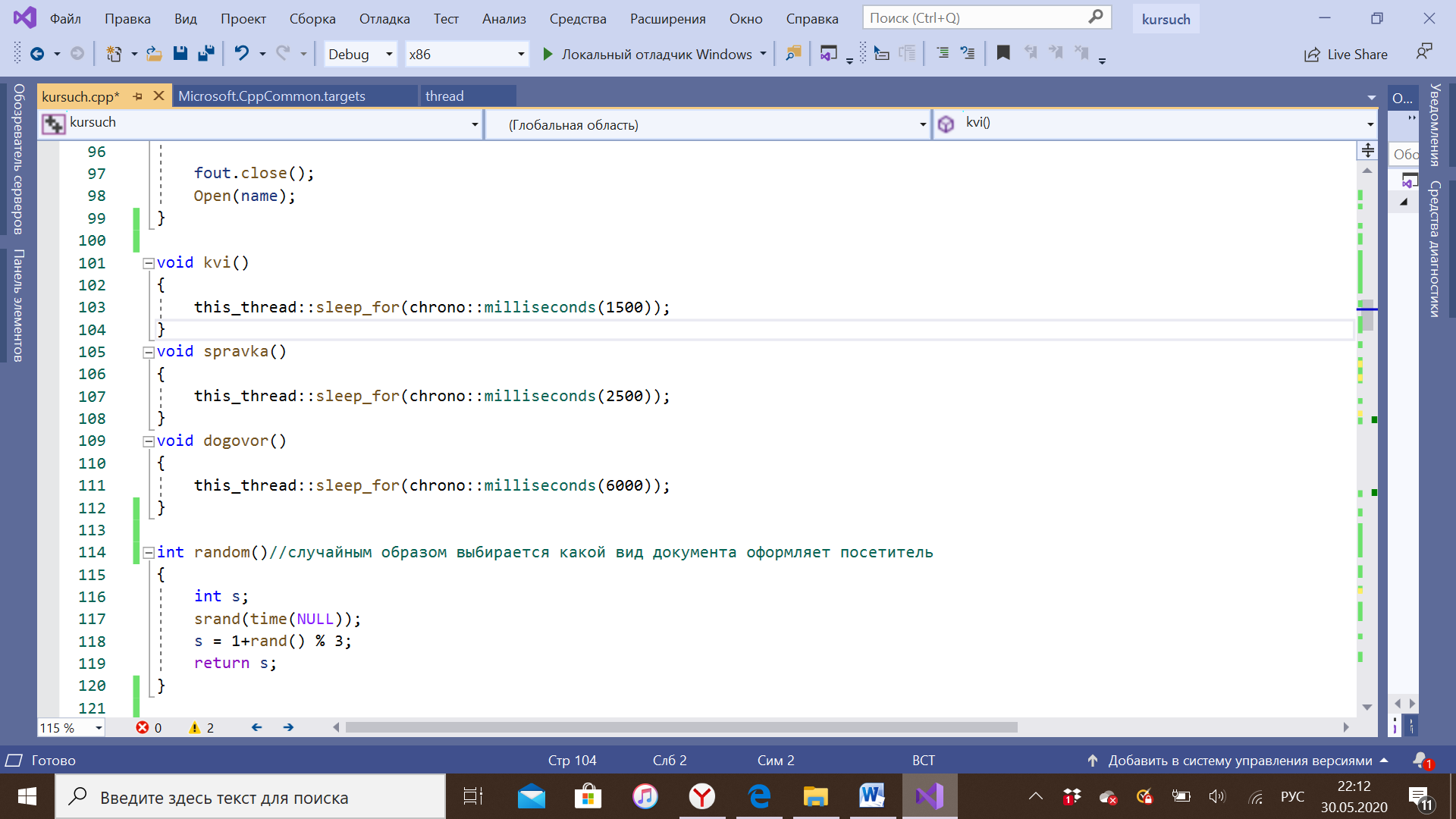
**Функция okno2()**

Работает аналогично функции okno1(), но без блока с функцией dogovor()



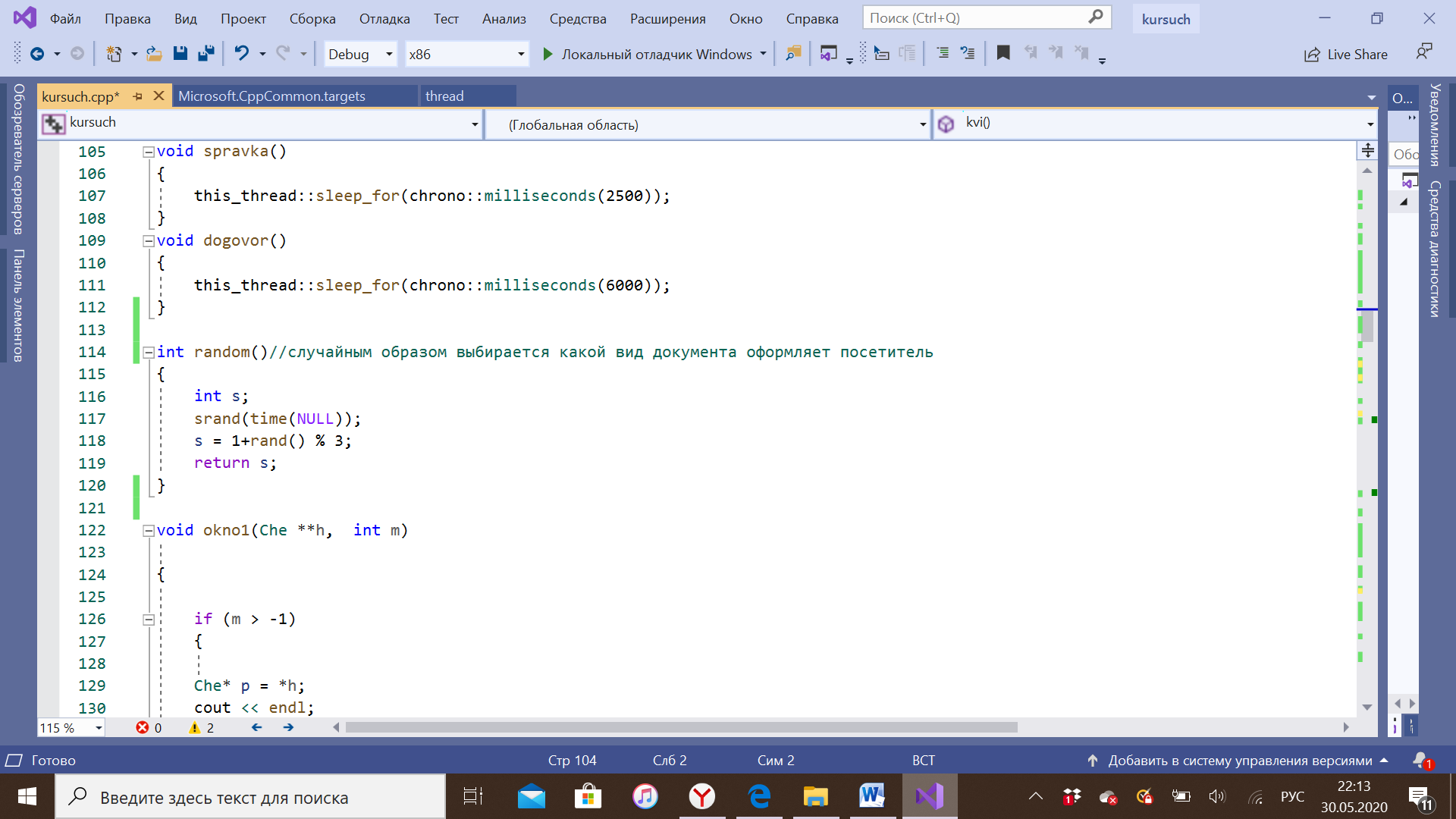
**Функции kvi(), spravka(), dogovor()**

Останавливают на определенное время работу потока



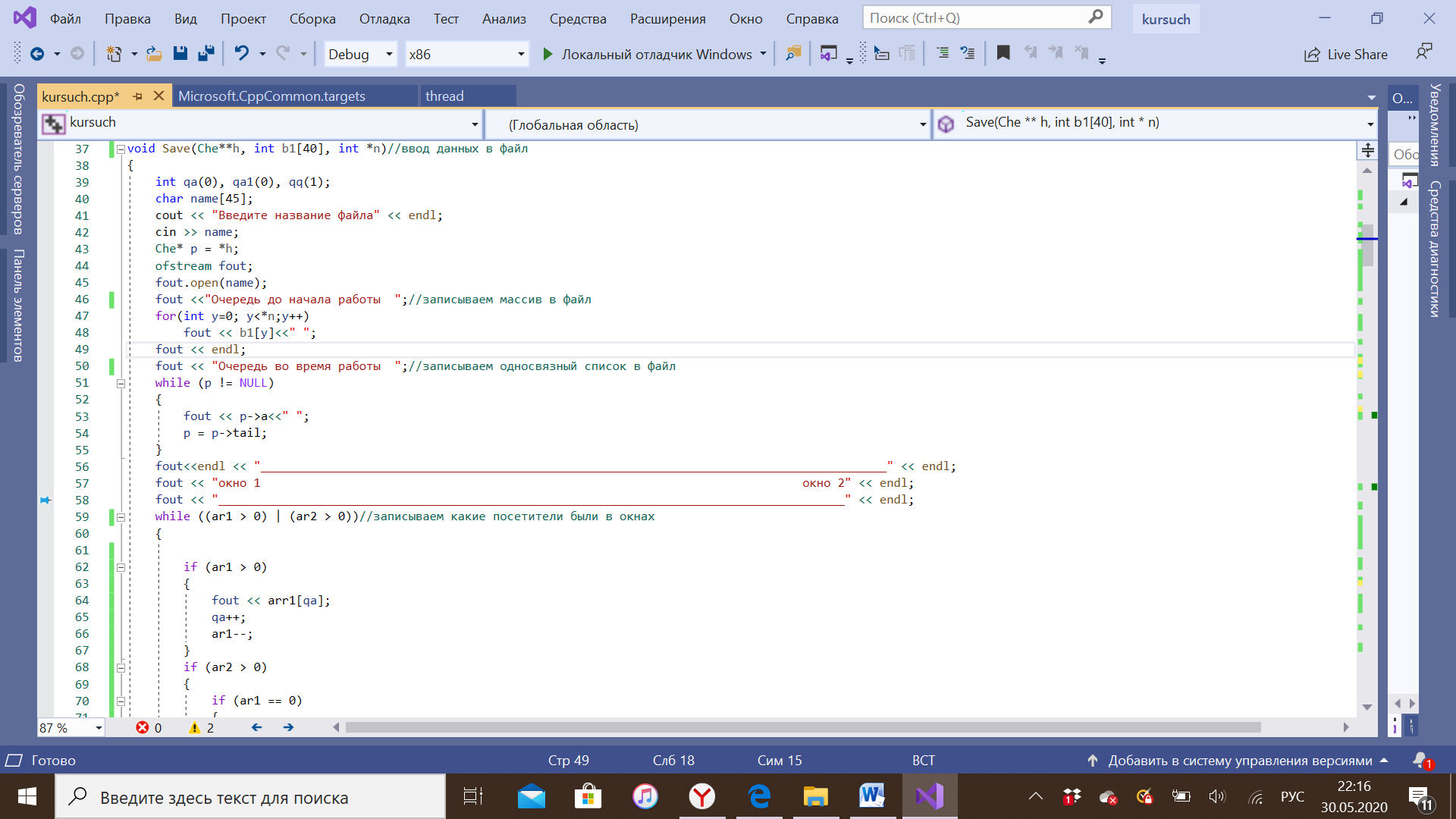
**Функция random()**

Случайным образом выбирает, какой вид документа оформляет посетитель

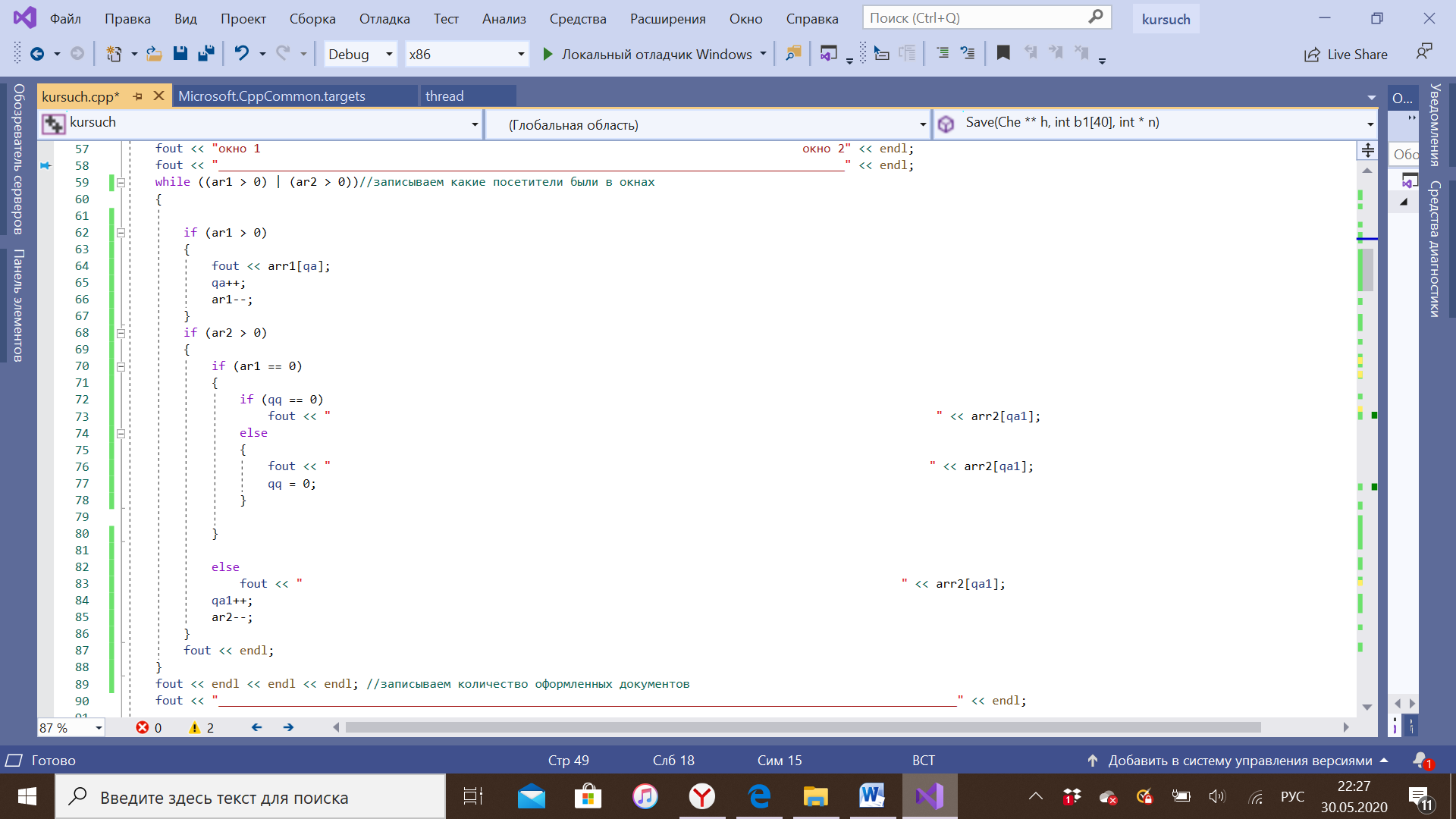


**Функция Save()**

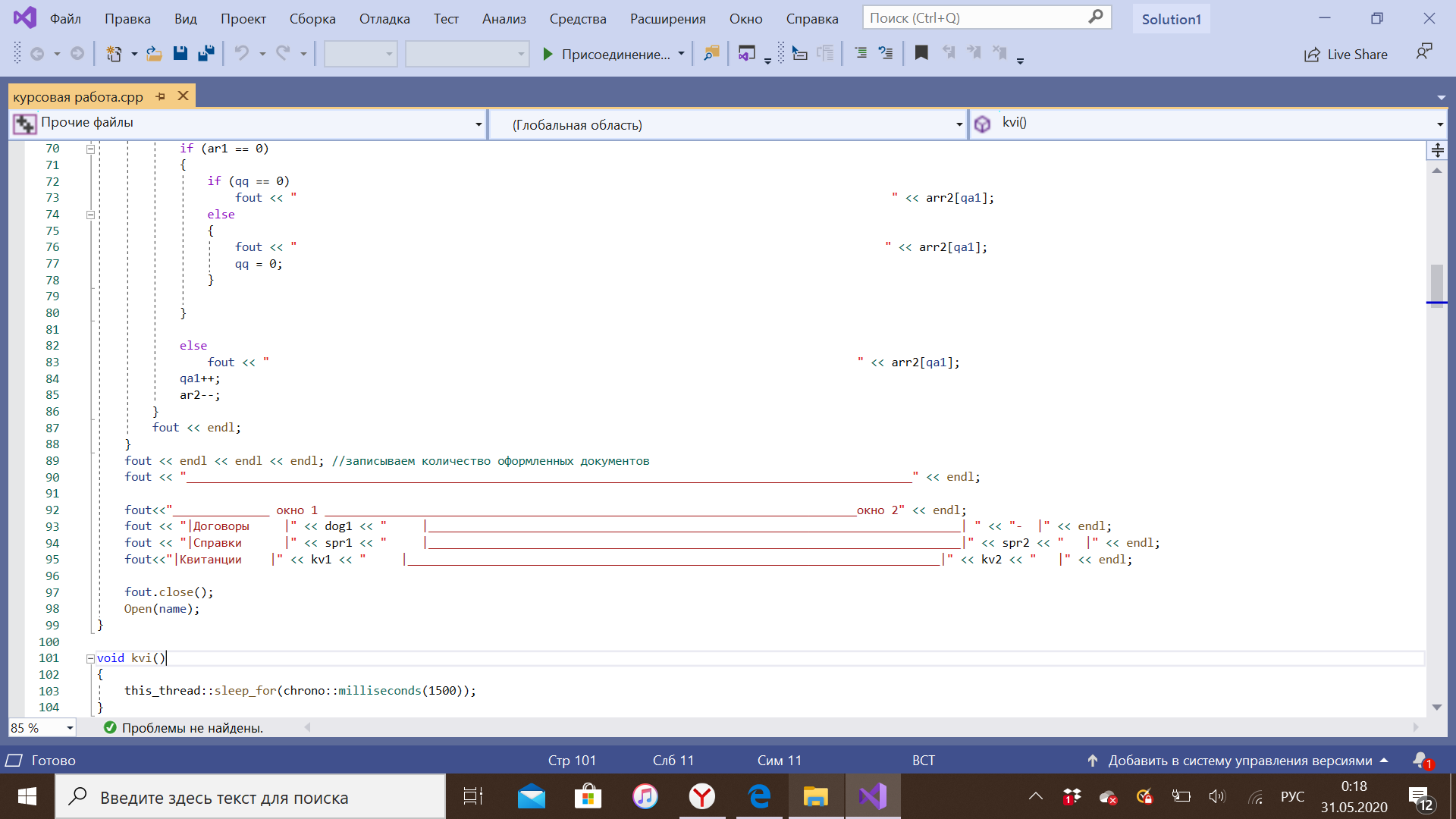
Вводим название файла. Открываем его. Записываем очередь, которая была введена пользователем. Записываем очередь, полученную в результате работы программы.



Записываем распределение посетителей по окнам:



Записываем количество документов, оформленных в окнах. Вызываем функцию, которая выводит данные из файла на экран.



**Функция Open()**

Выводит данные из файла.

