ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

###### Факультет информатики, математики и компьютерных наук

Курсовая бакалаврская работа

**РАЗРАБОТКА ИНФОРМАЦИОННОЙ ПОДСИСТЕМЫ ОТРАСЛЕВОЙ НАПРАВЛЕННОСТИ НА ПРИМЕРЕ РЕЦЕПЦИИ ГОСТИНИЦЫ**

по направлению подготовки 09.03.04 Программная инженерия

образовательная программа «Программная инженерия (о/з)»

|  |  |
| --- | --- |
| **Научный руководитель:**  преподаватель  Саратовцев Артем Романович | **Автор работы:**  студент 2 курса, группы ВПИ-22  очно-заочной формы обучения  Витюгов Даниил Алексеевич  Нижний Новгород, 2024 |

**Оглавление**

[Введение 3](#_Toc56769031)

[Глава 1. Предпроектное обследование 5](#_Toc56769032)

[1.1. Анализ предметной области 5](#_Toc56769033)

[1.2. Проектирование пользовательского интерфейса 13](#_Toc56769034)

[Глава 2. Разработка приложения 16](#_Toc56769035)

[2.1. Проектирование интерфейса 16](#_Toc56769036)

[2.2. Реализация интерфейса приложения 18](#_Toc56769037)

[2.3. Тестирование приложения 22](#_Toc56769038)

[Заключение 25](#_Toc56769039)

[Список используемой литературы 26](#_Toc56769040)

# **Введение**

Современные компьютеры, обладающие большим объемом функциональных возможностей, находят применение практически во всех сферах деятельности человека. Благодаря им процессы, связанные с представлением и обработкой данных, человек может выполнить без особых усилий.

В век информационных технологий возрастает популярность процессов автоматизации работ, своевременной обработки данных. Это существенно повышает уровень организации производства, облегчает прогнозирование и анализ хозяйственной деятельности, что способствует совершенствованию оперативного и долгосрочного планирования. Сейчас задачей всех организаций является минимизация затрат рабочего времени, материальных и трудовых ресурсов. Использование автоматизированных информационных систем позволяет снизить вышеуказанные показатели и упростить процесс обработки информации.

В настоящее время у преуспевающих организаций одной из основных составляющих деятельности является создание и использование баз данных и информационных систем. Все основные операции по накоплению, хранению и дальнейшей обработке делегируются вычислительной технике. Специалистом выполняется лишь малая часть ручных работ. Из всего этого следует то, что освоение принципов построения и применения соответствующих технологий и программных продуктов приобретает все большую актуальность.

Задачей данной работы является разработка предложения по автоматизации для рецепции гостиницы.

Гостиница выступает объектом исследования.

Цель курсового проекта: разработка информационной системы “Гостиница”, обеспечивающей автоматизацию процессов.

Задачи, которые необходимо решить для достижения поставленной цели:

1. исследование системы учета подачи заявок,

2. разработка функциональной модели предметной области (модель AS-IS),

3. разработка поведенческой модели (модель IDEF0, IDEF3).

В курсовой работе исследуемой организацией выступает гостиница, которая предоставляет клиентам номера с целью получения прибыли.

В работе гостиницы необходимо накапливать и хранить большое количество информации о постояльцах. До появления информационных технологий все данные хранились на бумажных носителях, что сильно снижало эффективность работы по обработке информации.

Целью разработки информационной системы является максимально возможное устранение ошибок в работе персонала, а также автоматизирование процессов создания документации и отчетов, простой и быстрый поиск, уменьшение времени выполнения каждой функции.

## **Глава 1. Предпроектное обследование**

# **1.1. Анализ предметной области**

Гостиница - имущественный комплекс с меблированными комнатами («номерами») для временного проживания.

В функционирование гостиницы входит:

1. Регистрация клиентов;
2. Учет состояния номеров;
3. Прием заявок на бронирование номеров;
4. Расчет стоимости проживания;
5. Справочная информация о дополнительных услугах.

Описание взаимодействия системы с окружающей средой происходит с помощью входов («Заявка клиента» и «Информация о клиенте»), выходов («Услуги», «Договор», «Номер» и «Чек»), управления («Законы РФ» и «Устав гостиницы») и механизмов («Материально-техническое обеспечение» и «Персонал»).

Заявка клиента – обращение, с целью получения услуг.

Информация о клиенте – сведения, необходимые для бронирования номера.

Услуги – экономические блага, направленные на удовлетворение потребностей клиента.

Договор – соглашение, где исполнитель (в лице представителя гостиницы) обязуется осуществить комплекс действий по обеспечению комфортного временного размещения и связанного с ним обслуживания заказчика (постояльца или временного гостя) в гостинице, а заказчик обязуется оплатить эти услуги.

Номер - помещение, оборудованное согласно требованиям, которые предъявляются к гостинице данной категории.

Чек – особый вид ценных бумаг, предъявляемых клиенту после оплаты услуг.

Законы РФ - законы по защите прав потребителя, и те, которые тем или иным образом контролируют качество, оказываемых гостиницей услуг.

Устав гостиницы - свод правил, которому подчиняются все работники гостиницы.

Материально-техническое обеспечение - дизайн, обстановка комнат и холлов, бытовые средства и инвентарь, предназначенный для уборки помещений, предметы гигиены, постельное белье и др.

Персонал гостиницы – работающие в гостинице люди, которые осуществляют приём клиентов, администрирование номеров, уборку комнат и др.

**Функциональная модель (IDEF0)**

Задачей функциональной модели является описание существующих бизнес-процессов предприятия (модель AS-IS) и выяснение, к чему нужно стремиться (модель TO-BE).

Благодаря методологии функционального моделирования IDEF0 существует возможность построения системы диаграмм для описания фрагментов системы в виде иерархии. Первым этапом является общее описание системы и ее взаимосвязи с окружающим миром (контекстная диаграмма). Далее проводится функциональная декомпозиция – описание подсистем в результате разбиения системы (диаграммы декомпозиции). Затем происходит процесс разбиения подсистем на более мелкие, пока не будет достигнута определенная степень подробности.

Все IDEF0-диаграммы содержат блоки и дуги.Изображение функций моделируемой системы происходит с помощью блоков, которые имеют вид прямоугольников. Они могут означать определенные процессы или задачи, происходящие в течение определенного времени и имеющие доступные для распознавания результаты. Имя работы – отглагольное существительное, обозначающее действие. Дуги используются для связи блоков и отображения взаимодействия между ними.

Все стороны блока имеют определенное назначение. Для входов предназначена левая сторона, для выходов – правая, для управления – верхняя, а для механизмов – нижняя. Такое расположение связано с определенными принципами: входы перестраиваются в выходы, управление предписывает условия преобразования, механизмы отображают, что и как выполняет функция.

Описание функционирования системы в виде контекстной диаграммы (см. Рис. 1) для построения информационной системы «Гостиница»:

В терминах входа (“Заявка клиента” и “Информация о клиенте”), выхода (результат процесса - “Номер”, “Услуги” “Договор” и “Чек”), управления (“Законы РФ” и “Устав гостиницы”) и механизмов (“Материально-техническое обеспечение”, “Персонал”) описывается взаимодействие с окружающей средой.

В качестве клиента выступает лицо, использующее услуги гостиницы. Оказанные услуги оплачиваются и приносят прибыль гостинице, что является основной целью деятельности любого предприятия.

“Законы РФ” и “Устав гостиницы” – это упорядоченные правила, обеспечивающие управление процессом функционирования гостиницы. Все гостиницы имеют внутренние правила и обязанности соблюдения законодательства Российской Федерации.

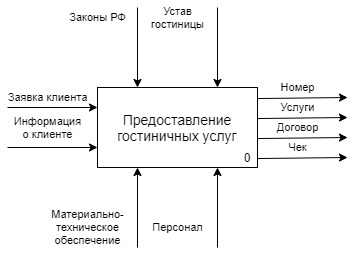
**

Рис. 1 Контекстная диаграмма IDEF0. Функционирование гостиницы

“Персонал” гостиницы принимает участие в оказании услуг клиенту. Для предоставления номеров и получения прибыли необходимо “Материально-техническое обеспечение”.

Данная модель описывает следующие предоставляемые гостиницей услуги:

1. предоставление гостиничных номеров,
2. оказание дополнительных услуг,
3. оформление выезда из номера,
4. прием оплаты.

После описания контекстной диаграммы происходит разбиение системы на подсистемы, которые впоследствии отдельно описываются, то есть проводится функциональная декомпозиция (см. Рис. 2).

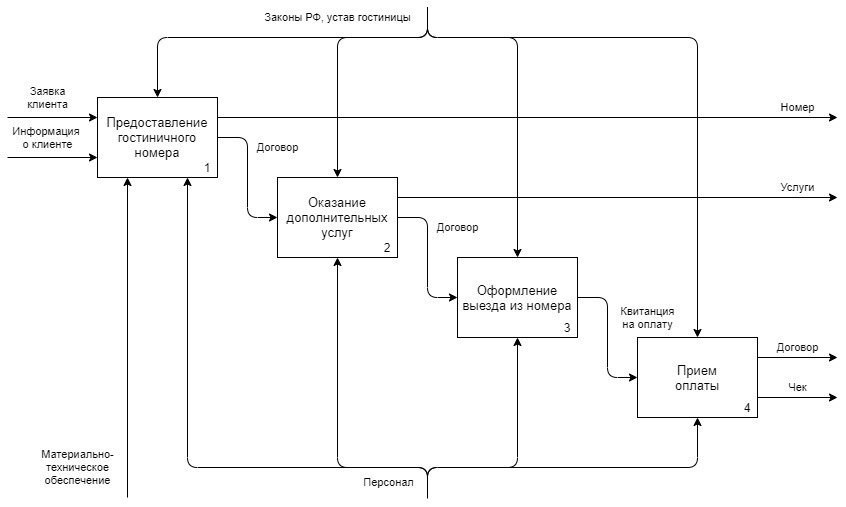


Рис. 2 Диаграмма декомпозиции IDEF0. Функционирование гостиницы

“Предоставление гостиничного номера” - сдача номера клиенту, который прошел регистрацию;

“Оказание дополнительных услуг” – предоставление платных услуг, повышающих рейтинг и конкурентоспособность гостиницы.

“Оформление выезда из номера” – формирование гостиницей итогового счёта, который необходимо оплатить клиенту за вычетом предоплаты.

**Модель потоков данных (DFD)**

Основным средством моделирования функциональных требований к создаваемой системе являются диаграммы потоков данных (Data Flow Diagramming, DFD). Они показывают каким образом происходит преобразование процессом своих входных данных в выходные, а также выявляют отношения между этими процессами. Как и IDEF0, DFD моделируемая система представляется как сеть связанных работ. Диаграммы потоков данных применяются для описания документооборота организации и обработки информации. Основными компонентами DFD являются процессы (работы), внешние сущности, потоки данных и хранилища данных. В связи с тем, что в нотации DFD существуют понятия “внешняя ссылка” и “хранилище данных”, процесс моделирования документооборота более удобен, чем в IDEF0.

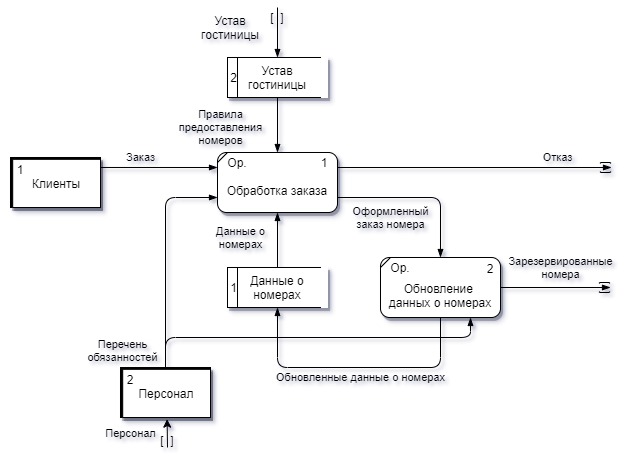


Рис. 3 Диаграмма декомпозиции в нотации DFD. Предоставление гостиничного номера

На рисунке 3 представлена “Диаграмма декомпозиции в нотации DFD. Предоставление гостиничного номера”, описывающая деятельность по предоставлению клиентам номеров.

“Персонал” и “Клиенты” – это внешние ссылки.

“Данные о номерах отеля” и “Устав гостиницы” – хранилища данных.

Создание клиентского приложения позволит собрать все данные в одном месте и упростит процесс обновления данных о постояльцах.

В отличие от стрелок IDEF0, представляющие собой фиксированные взаимосвязи, стрелки DFD отображают движение объектов (в том числе данных) от одной работы к другой. Например, “Заказ”, приходящий от клиента в определенной форме (телефонный звонок, обращение к персоналу), влечет за собой процедуру “Обработка заказа”. “Персонал” выполняет этот процесс, производит запрос информации из хранилища “Данные о номерах” и, согласованно с “Правилами предоставления номеров” резервирует номер, или же отказывает клиенту в резервировании номера. Если произошло “оформления заказа номера”, данные о номерах обновляются и заносятся в хранилище “Данные о номерах”.

**Физическая UML-диаграмма**

Формирование физической диаграммы происходит на основе информации о деятельности гостиницы и ее бизнес-процессов.

Физическая диаграмма гостиницы представлена на рисунке 4. Основными объектами, которые осуществляют взаимодействие с гостиничным предприятием, являются клиенты, поставщики и службы такси.

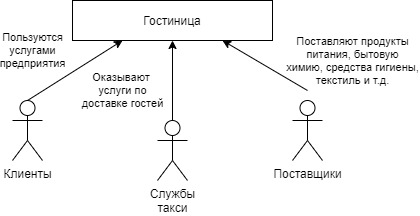


Рис. 4 Физическая UML – диаграмма

**Диаграмма прецедентов**

Данная диаграмма предназначена для моделирования взаимодействия проектируемой информационной системы с внешним миром. Основные элементы диаграммы – актеры и прецеденты (или варианты использования) Актеры выступают внешними сущностями по отношению к системе, а прецеденты моделируют функции проектируемой ИС. Рамкой на диаграмме изображаются границы информационной системы. Диаграмма прецедентов для информационной системы гостиницы представлена на рисунке 5.

Основное назначение диаграммы – функциональное и поведенческое описание, которое позволяет заказчику, разработчику и конечному пользователю совместно обсуждать проектируемую или имеющуюся систему.

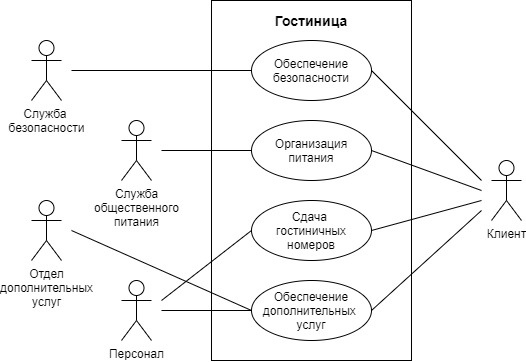
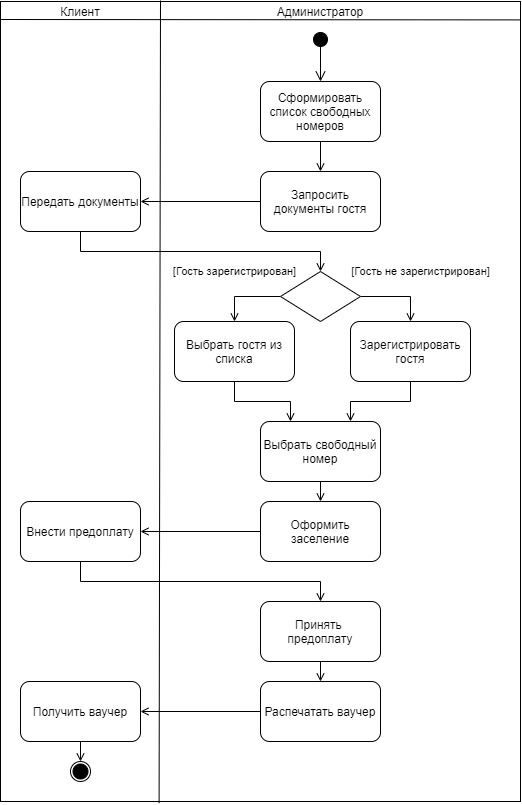


Рис. 5 Диаграмма прецедентов

**UML-диаграмма деятельности**

Диаграмма деятельности отображает разложение некоторой деятельности на составные части. Диаграмма для прецедента “Предоставление гостиничного номера” представлена на рисунке 6. При заселении в гостиницу клиент большую часть времени взаимодействует с администратором.



*Рис. 6 UML-диаграмма деятельности. “Предоставление гостиничного номера”*

На диаграмме присутствует узел ветвления. Если гость останавливается в гостинице в первый раз, то необходима регистрация. Если же гость является “постоянным”, то достаточно выбрать его из списка.

# **1.2. Проектирование пользовательского интерфейса**

Проектирование интерфейса состоит из:

1. описания структуры приложения,
2. описания полей,
3. представления внешнего вида приложения.

В данной пункте будет представлен пользовательский интерфейс гостиницы.

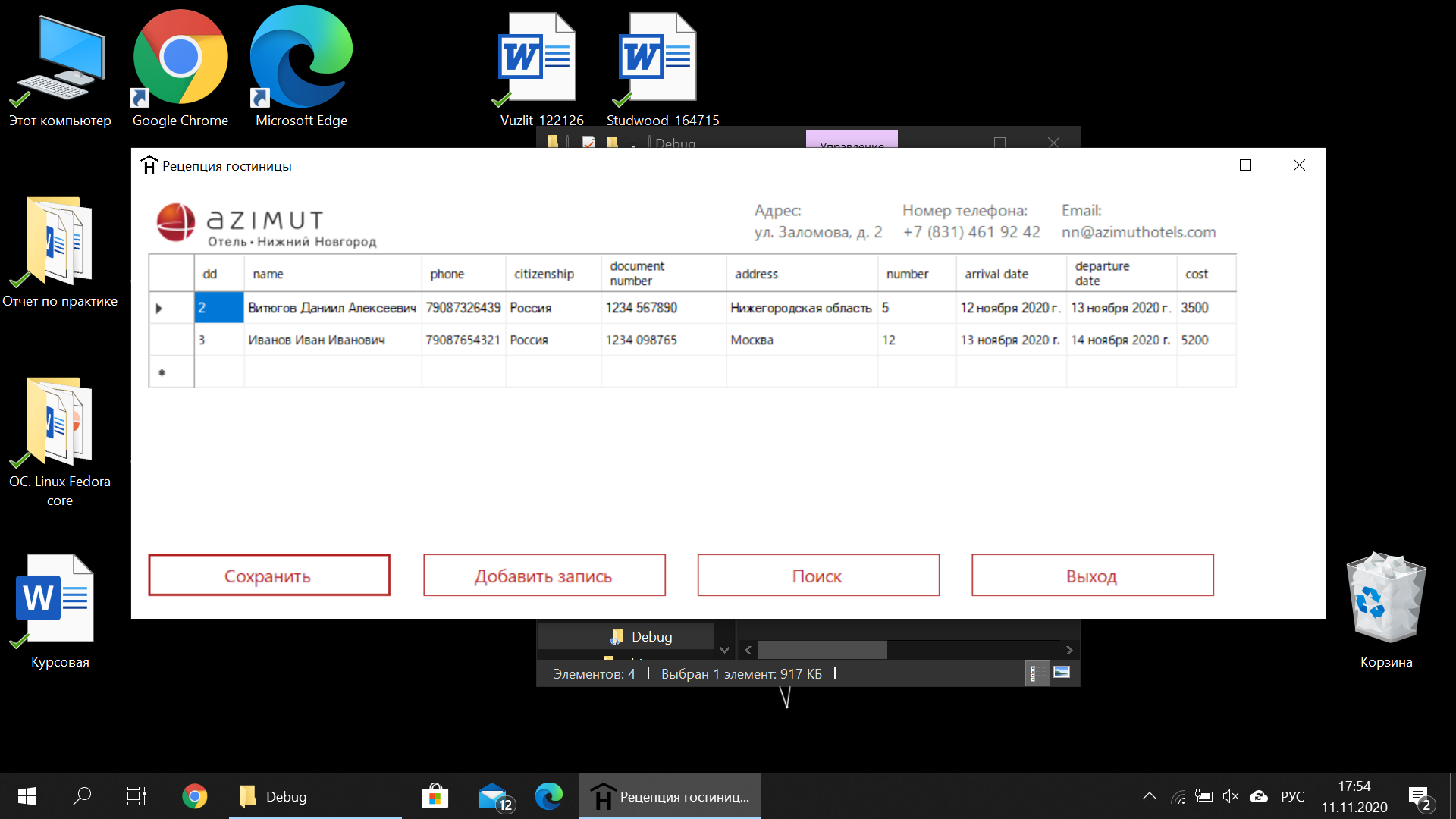
Важным функциональным элементом приложения гостиницы является база данных, которая должна выполнять следующие основные задачи:

1. хранить сведения обо всех номерах гостиницы,
2. хранить сведения о забронированных номерах клиентов,
3. обеспечивать поиск нужного номера,
4. обеспечивать оформление номера на нужного покупателя.

База данных должна содержать следующую информацию о клиентах: номер заказа, полное имя, номер телефона, гражданство, номер и серию паспорта, адрес проживания, зарезервированный номер, сроки проживания и цена. Возможность просматривать эти данные есть только у администрации.

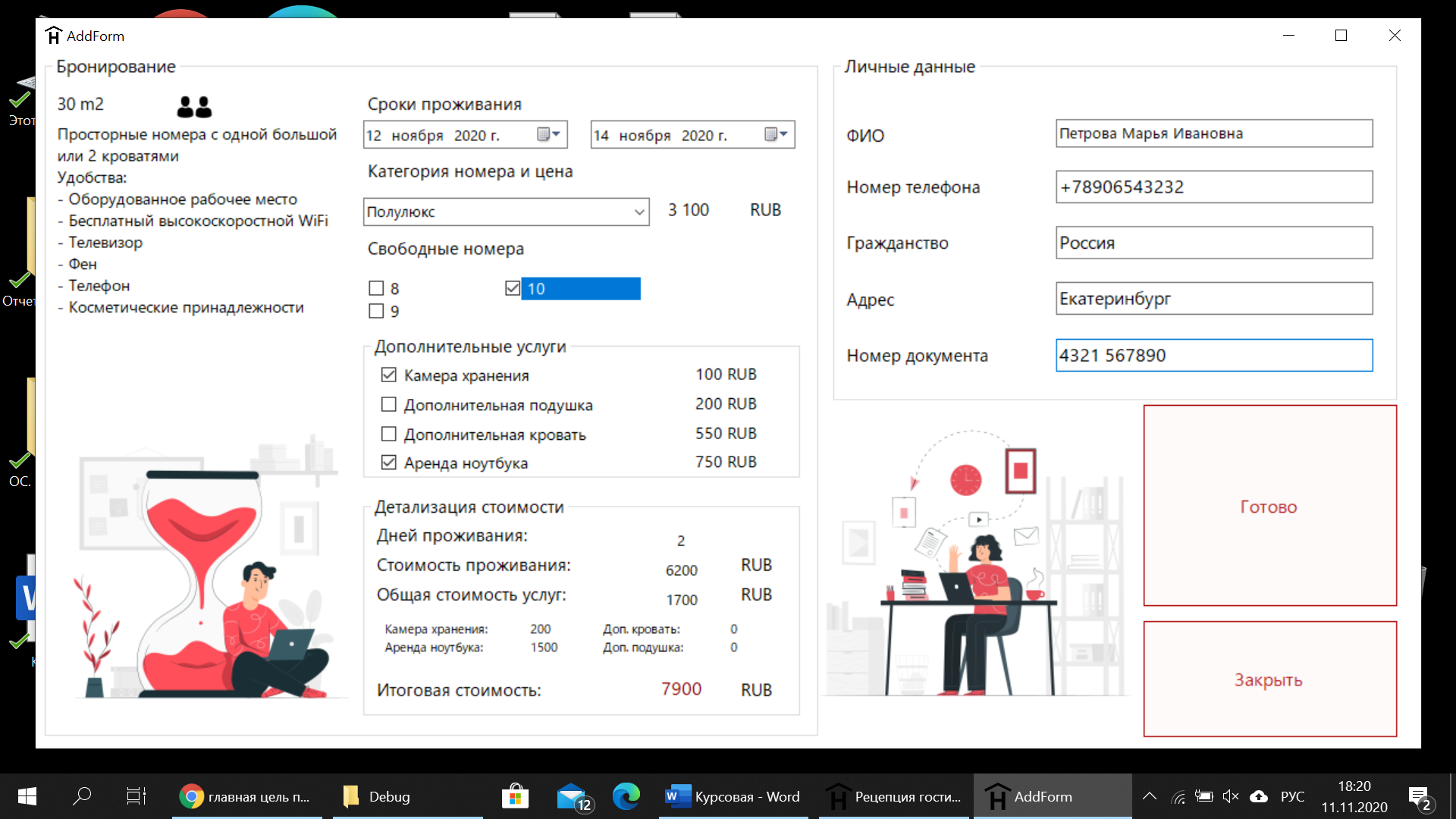
Приложение, в связи со своим функционалом, состоит из 3 форм.

Главная форма (см. Рис. 7) появляется при открытии приложения. Основным элементом здесь является база данных, где отображаются сведения о постояльцах. Над ней находятся данные о гостинице (логотип, адрес, номер телефона и электронная почта). Внизу расположены функциональные кнопки: кнопка сохранить обновляет базу данных, кнопка выход закрывает приложение, кнопки поиск и добавление записи открывают новые окна, где можно произвести соответствующие названиям действия. Малое количество элементов и их логическое расположение обеспечивает короткий и удобный путь к получению интерфейса, что является главной целью всех пользовательских интерфейсов.



*Рис. 7 Главный экран приложения “Гостиница”*

Форма Добавления записи изображена на Рис. 8. Она является самой сложной, потому что здесь реализовано большое количество функций. Для легкости восприятия окно было разделено на две условные части: бронирование и личные данные.



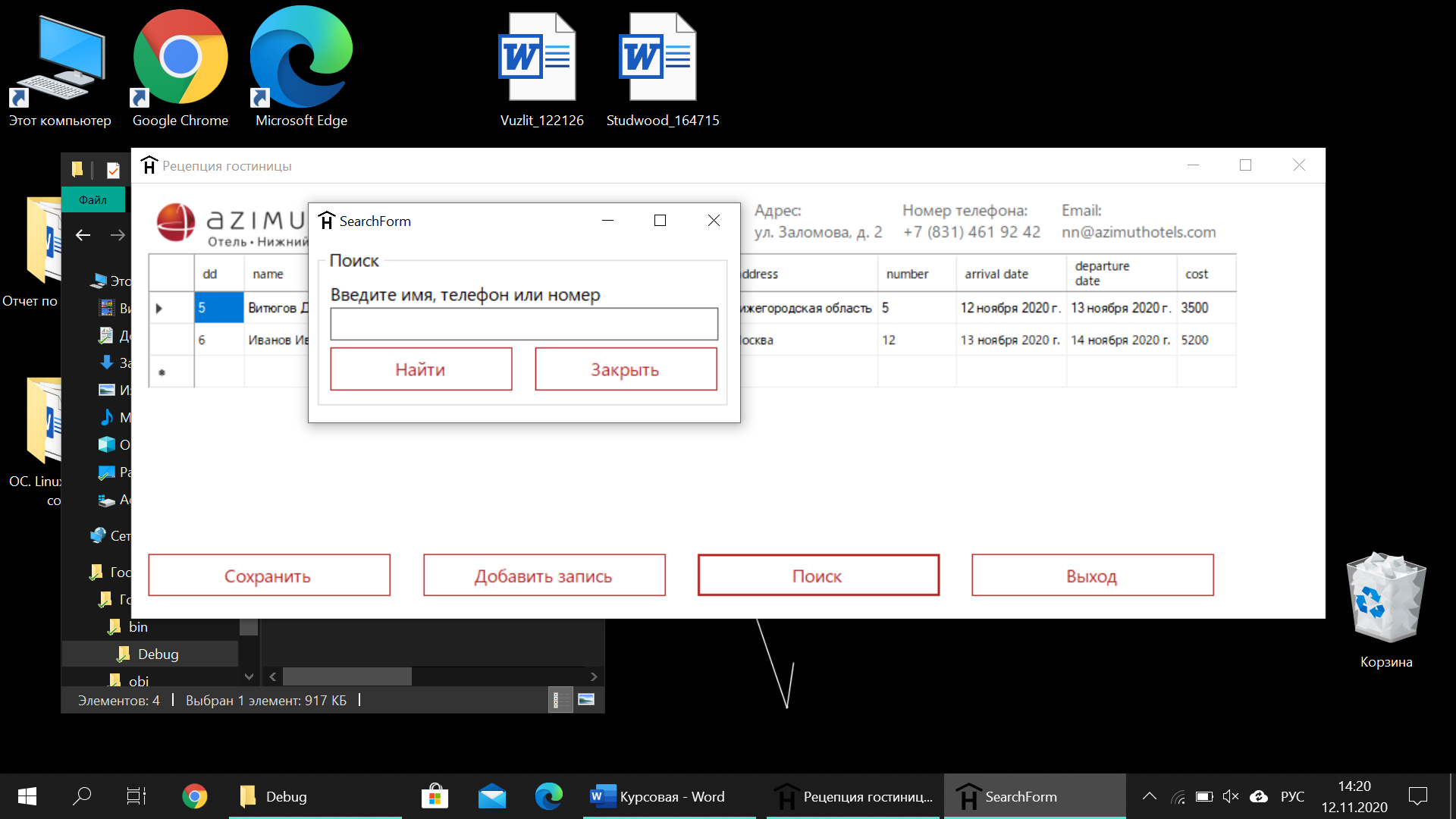
*Рис. 8 Форма “Добавление записи”*

В группе элементов “Бронирование” происходит внесение данных о сроках проживания, типе номера и свободных номерах, дополнительных услугах и стоимости, в зависимости от вышеуказанных характеристик. Для легкого ориентирования персонала в информации, при выборе категории на экран выводится вся данные о номере (площадь, вместимость, описание и удобства).

После того, как на экране отображена детализация стоимости, осуществляется переход к группе элементов “Личные данные”, где необходимо ввести полное имя, номер телефона, гражданство, адрес, серию и номер паспорта. Гостиница должна обеспечить сохранность всей информации о клиенте и номере.

Кнопка Готово предназначена для внесения заполненной информации в базу данных. Кнопка Выход осуществляет закрытие текущей формы.

С помощью формы Поиск (см. Рис. 9) у пользователя имеется возможность найти запись для уточнения других сведений о заказе.



*Рис. 9 Форма “Поиск”*

Грамотное проектирование пользовательского интерфейса приложения в дальнейшем экономит время, облегчает работу и снижает вероятность дополнительных вложений.

Наряду с экономической и технической стороной вопроса, дизайн влияет на востребованность у аудитории. Если программный продукт не востребован, то можно сказать, что труды разработчиков напрасны.

**Глава 2. Разработка приложения**

**2.1. Проектирование интерфейса**

Под проектированием интерфейса понимается написание кода программы. Кодом, в данном случае, являются все закодированные элементы и слова на объектно-ориентированном языке программирования C#.

Написание программы производилось с помощью Microsoft Visual Studio (линейка продуктов компании Microsoft, которая включает в себя интегрированную среду разработки программного обеспечения и ряд других инструментальных средств).

Для создания проекта выбран шаблон “Приложение Windows Forms (.NET Framework)”, являющийся частью Microsoft и отвечающий за графический интерфейс пользователя. После выбора шаблона происходит указание места хранения будущей программы и финальным этапом является запуск окна Конструктора.

Первым этапом работы стало проектирование пользовательского интерфейса программы. Были определены пользовательские сценарии и начался процесс создания.

При проектирование главной формы (см. Рис. 7) использованы следующие элементы: DataGridView, отвечающий за подключение базы данных (в качестве системы управления базами данных была выбрана программа Microsoft Access); 4 функциональных блока Button, при щелчке на которые возникает определенное событие; PictureBox, с помощью которого добавлен логотип гостиницы; 3 элемента Label, служащих для вывода на экран информации о гостинице.

Форма добавления записи (см. Рис. 8), открытие которой происходит при нажатии одноименной кнопки, состоит из двух условных групп элементов, которые обозначены с помощью элемента GroupBox. Этот элемент также использован для группирования компонентов подгрупп “Дополнительные услуги” и “Детализация стоимости”.

В группе “Бронирование” происходит внесение данных о сроках проживания, для чего использовано 2 элемента dateTimePicker, где указываются дата прибытия и отъезда. СomboBox служит для выбора категории номера, при этом в соседних полях Label выводится общие сведения о номерах данной категории и стоимость за день проживания. Информация о свободных номерах выбранной категории появляется в СheckedListBox, где пользователь может выбрать один из вариантов. В объединении элементов “Дополнительные услуги” пользователь выбирает необходимые дополнительные услуги (рядом в Label указана стоимость каждой) и делает пометку в нужном СheckBox. В подгруппе “Детализация стоимости” самостоятельно происходит детальный расчет стоимости, в зависимости от выбранных характеристик.

Элементами группы “Личные данные” являются 5 полей TextBox, предназначенных для ввода сведений о клиенте, и соответствующее количество Label. На протяжении всего приложения элементы Label используются для конкретизации рядом стоящих элементов. Это позволяет пользователю легко ориентироваться в программе и помогает избежать ошибок при внесении информации в базу данных. Для удобства элементы TextBox используются только для ввода информации, что повышает уровень восприятие и помогает в определении ответных действий пользователя, возникающих в результате использования программного продукта (“User eXperience”).

Форма добавления записи, помимо предыдущих элементов, включает в себя два функциональных блока Button, отвечающих за сохранение данных и закрытие формы, а также два PictureBox, в которых содержится изображения. Они разбавляют внешний вид формы, добавляя ему свежести и динамики.

Заключительная форма поиска отвечает за нахождение записей из базы данных по введенному в TextBox набору символов. Как и на предыдущих формах Label служит для пояснения. Два функциональных блока Button отвечают за поиск и закрытие формы. Все элементы заключены в GroupBox.

**2.2. Реализация интерфейса приложения**

Следующим этапом в процессе разработке программного обеспечения является непосредственное написание кода. При выполнении всех вышеперечисленных этапов данная ступень подразумевает чисто механическую работу. Программисту в этом случае не нужно что-то выдумывать и самостоятельно разрабатывать, ему нужно просто написать код, который реализует заданный алгоритм.

Первым шагом стало подключение базы данных в компонент “DataGridView”, с помощью встроенного в Visual Studio “Мастера настройки источника данных”.

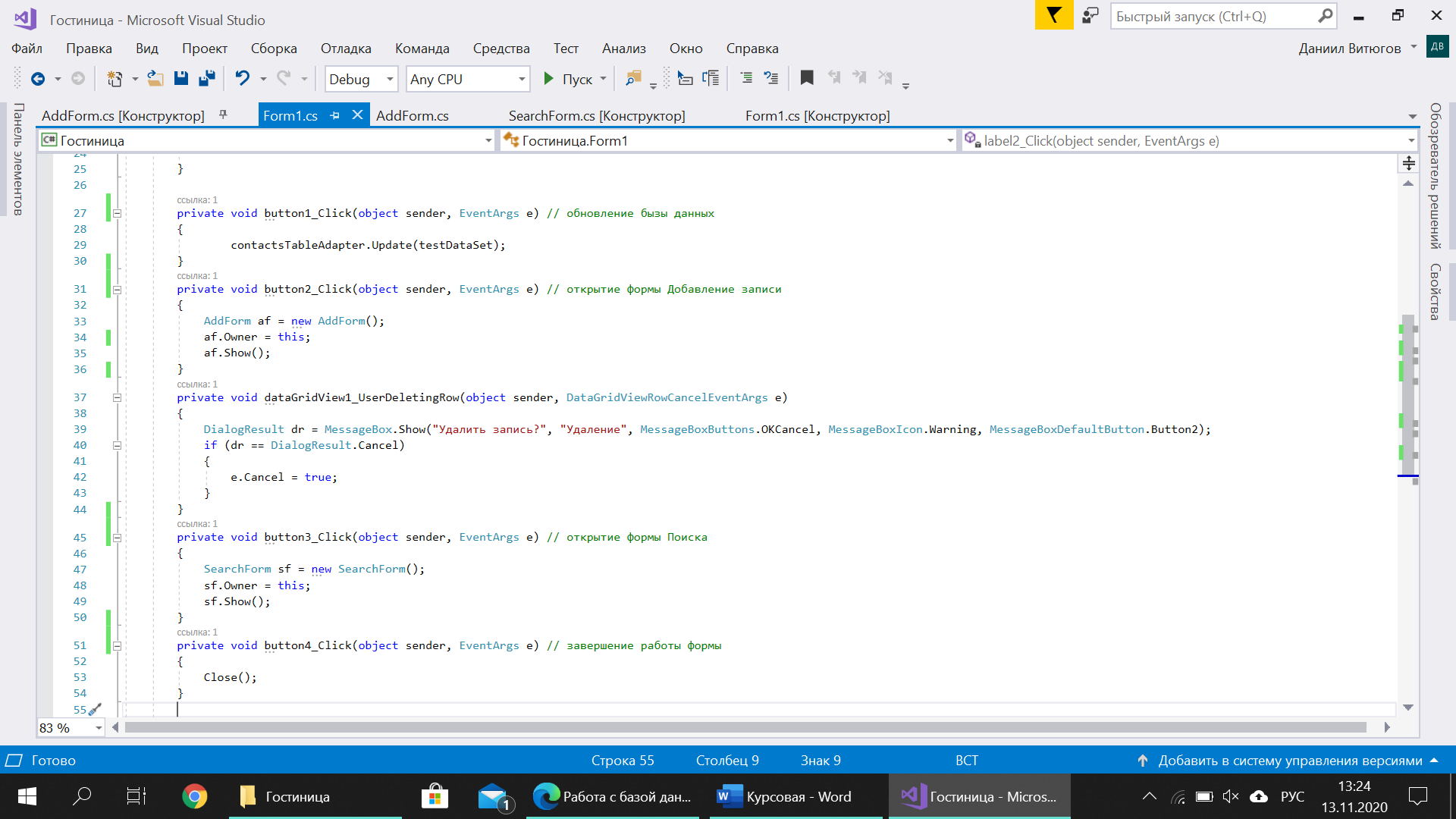
Следующей задачей стало создание событий при нажатии и на функциональные элементы “Button”. Каждый из них отвечает за обработку своего события. Фрагмент кода представлен на рисунке 10.

При нажатии на элементы button2 и button3 происходит открытие новых форм. Для этого необходимо сначала их создать, это можно выполнить щелчком правой кнопкой мыши в обозревателе решений на проект.

Чтобы пользователь случайно не удалил запись в базе данных, появляется окно, где пользователь подтверждает или отменяет процесс удаления. Для реализации этой функции был установлен обработчик для события UserDeletingRow у компонента DataGridView. Как только пользователь выделит строчку и нажмет кнопку “Delete” сработает событие “dataGridView1\_UserDeletingRow” и появится окно с вопросом об удалении.

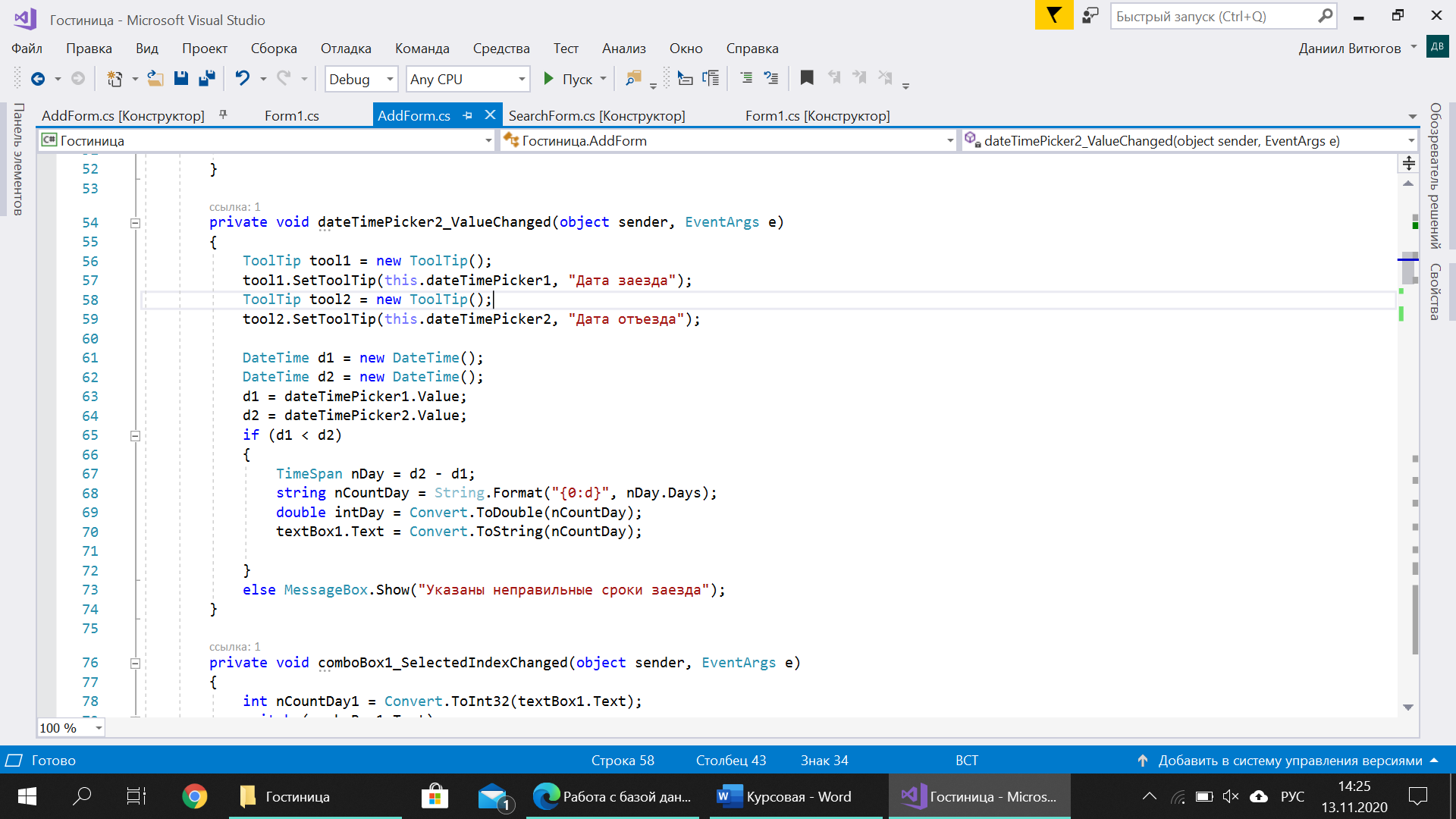
Чтобы база данных была доступна для разных с ней манипуляций во всех создаваемых нами формах необходимо изменить модификаторы доступа на «public» в «Form1.Designer.cs» у следующих файлов:

1. System.Windows.Forms.DataGridView dataGridView1,
2. testDataSet testDataSet,
3. System.Windows.Forms.BindingSource contactsBindingSource,
4. testDataSetTableAdapters.contactsTableAdapter contactsTableAdapter.



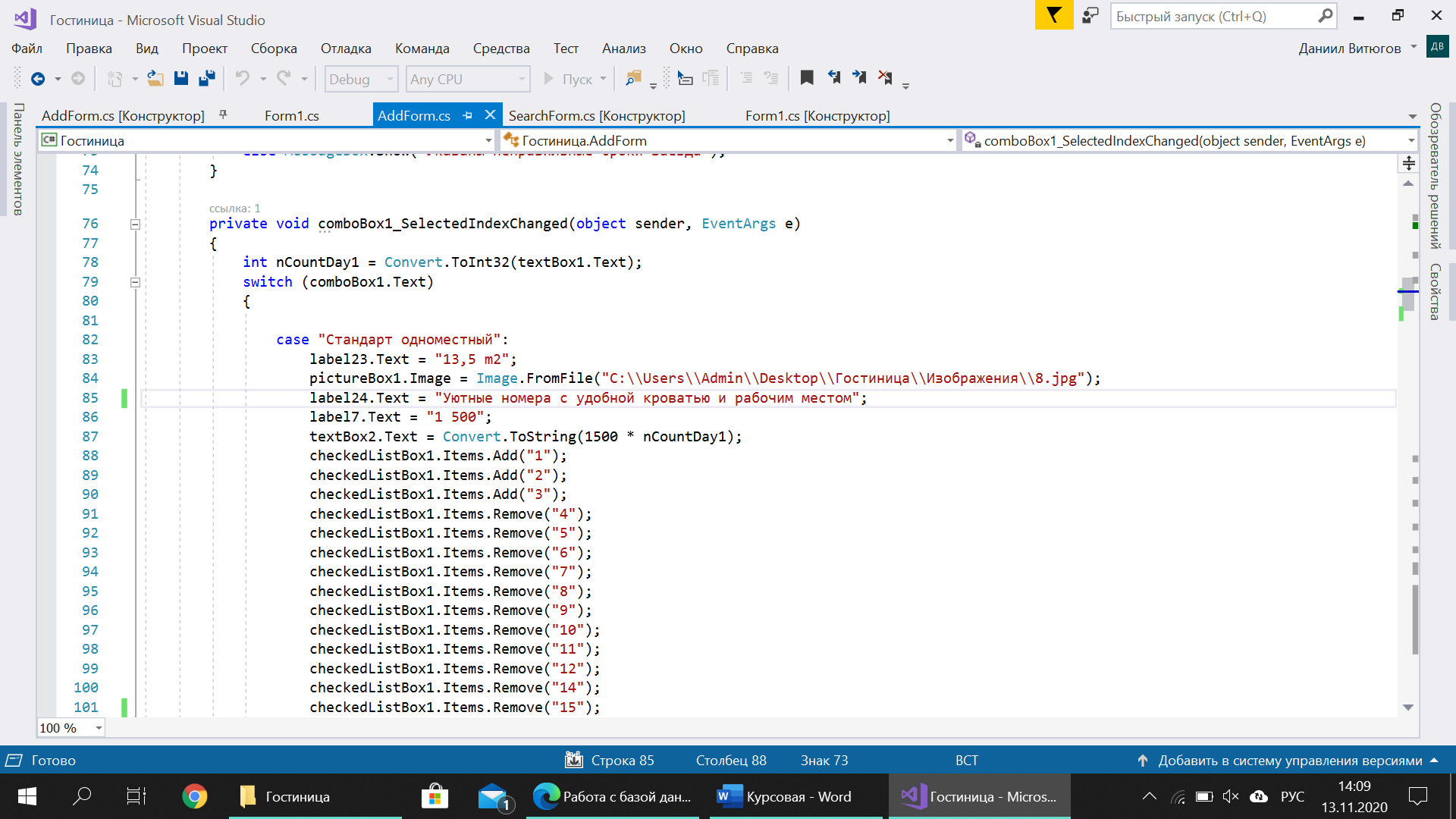
*Рис. 10 Создание событий*

Форма добавления записи предназначена для обработки и внесения сведений в базу данных. Здесь была реализована работа разнообразных элементов для разных типов задач. С помощью элементов dateTimePicker происходит выбор дат проживания, данные о которых потом отправятся в базу данных. Также они содержит в себе вычисления по количеству дней проживания в гостинице (Рис. 11).



*Рис. 11 Код в* *элементе* *dateTimePicker*

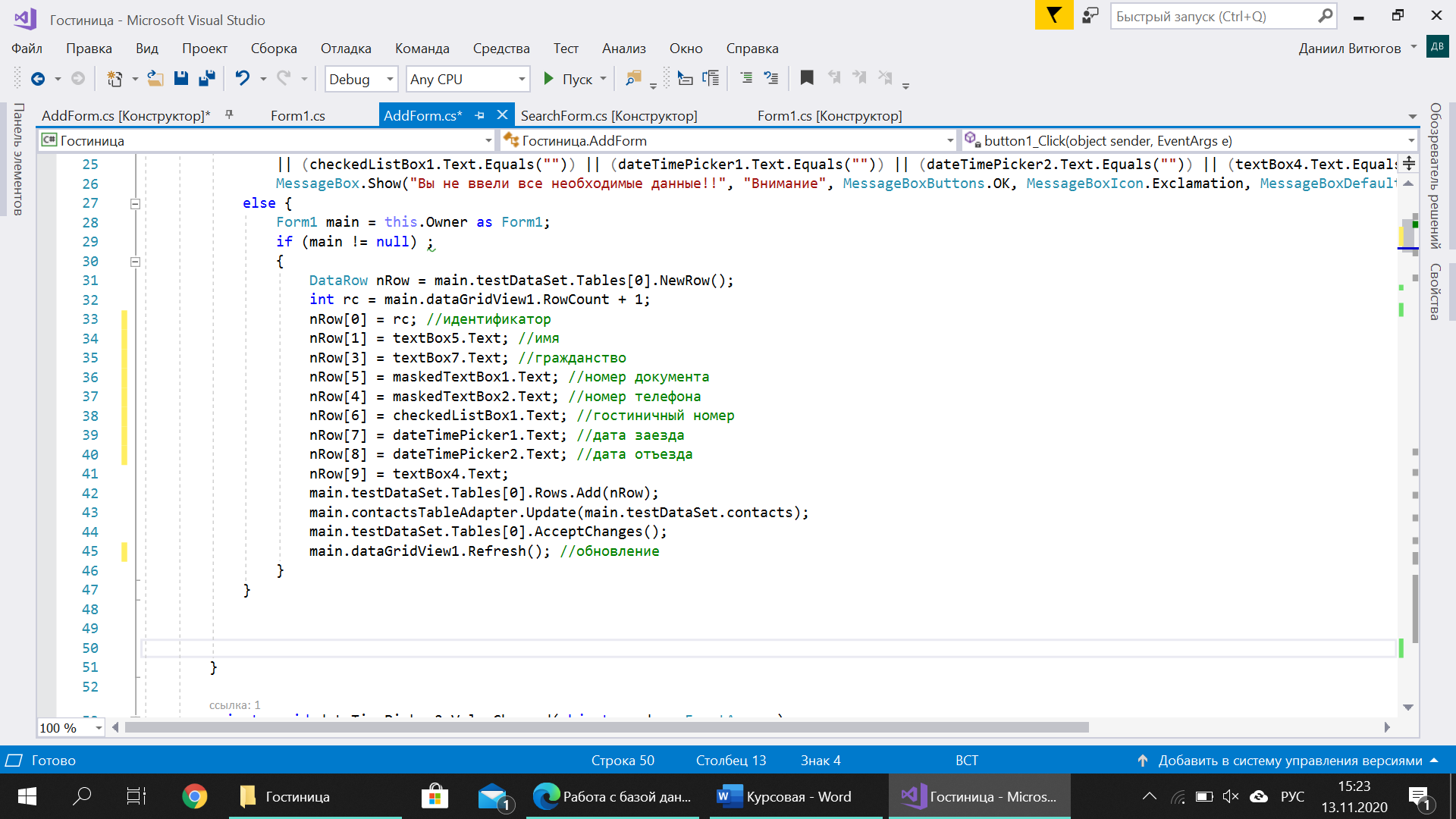
В элементе СomboBox прописан код для каждой категории номера. Здесь происходит вычисление стоимости проживания и вывод информации о номере. Также от выбранной категории зависит список номеров, вывод которых производится в СheckedListBox. Фрагмент кода представлен на рисунке 12.



*Рис. 12 Код в* *элементе* *СomboBox*

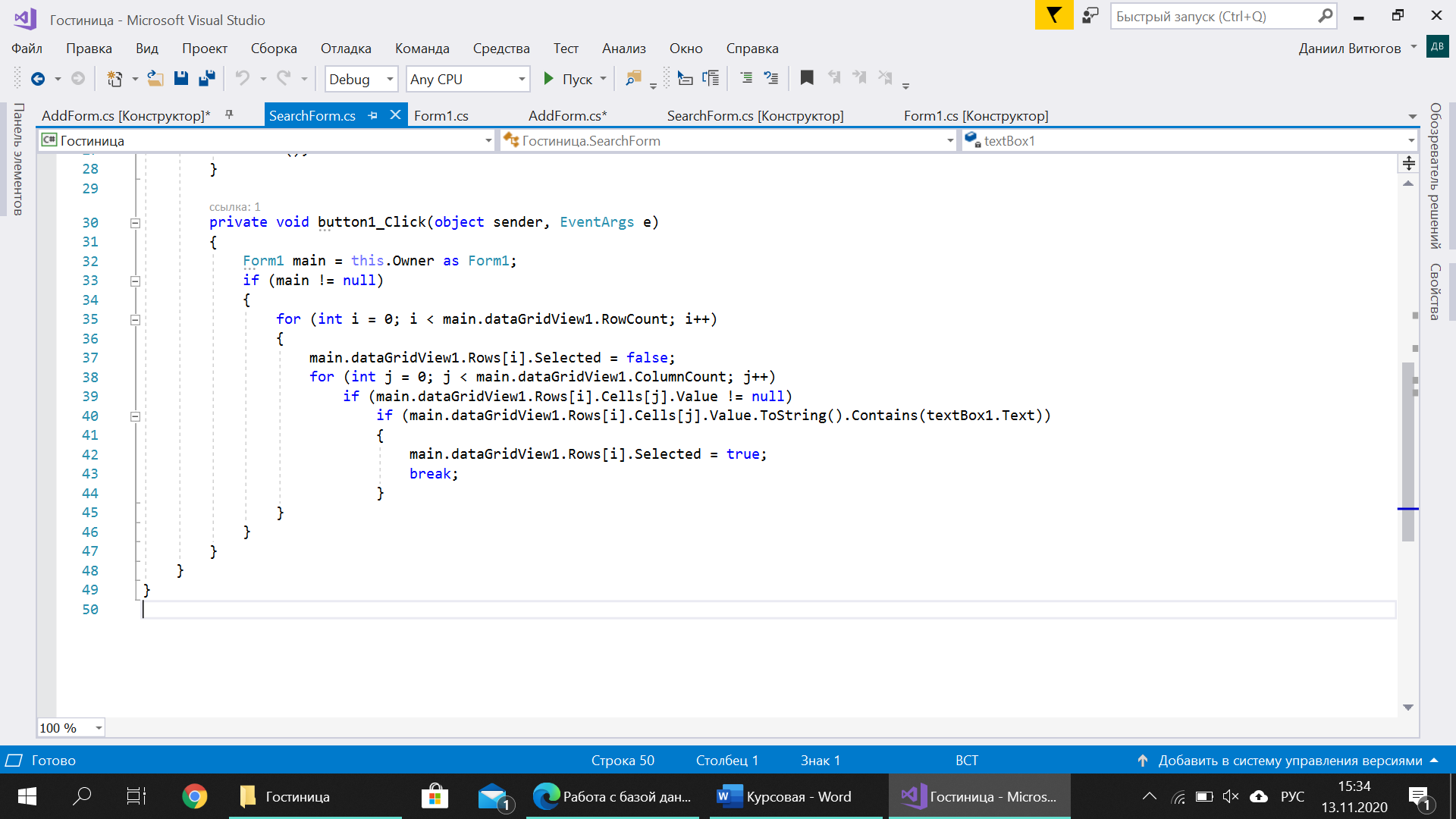
Итоговая стоимость зависит количества дней проживания, категории номера и дополнительных услуг. Каждая из этих характеристик вычисляется по отдельности и результат помещается в label. Затем происходит суммирование данных и вывод в собственный label.

Кнопка “Готово” производит внесение информации из заполненных полей в базу данных. Каждая запись имеет свой уникальный номер, который образуется при добавлении 1 к номеру предыдущей записи. Для внесения информации указаны номера столбцов базы данных. Фрагмент кода представлен на рисунке 13.



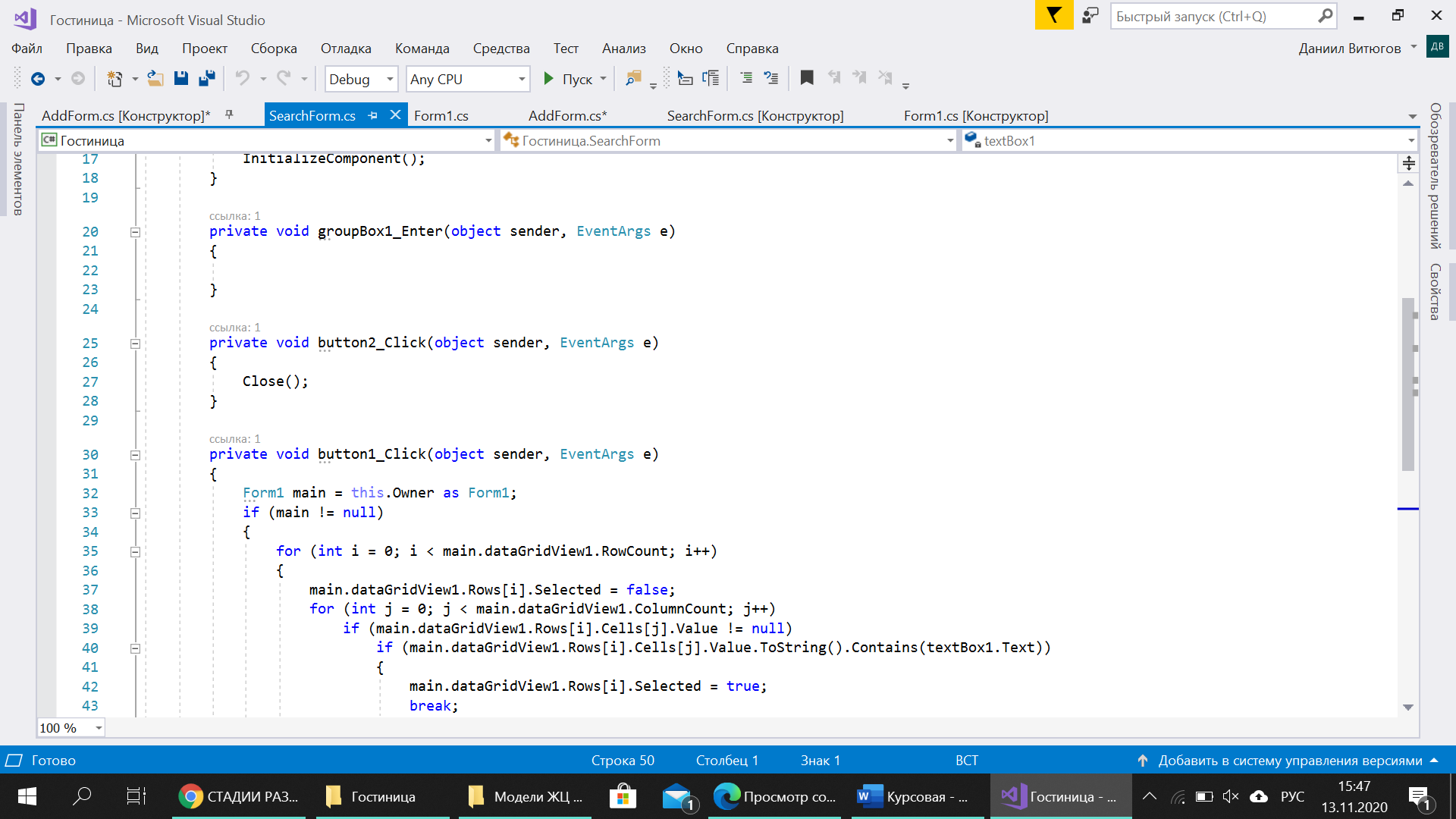
*Рис. 13 Код кнопки “Готово”*

Весь функциональный код формы Поиска находится в кнопке Найти, которая отвечает за поиск в ячейках определенного набора символов. Фрагмент кода представлен на рисунке 14.



*Рис. 14 Код кнопки “Найти”*

Во всех формах имеется кнопка Закрыть (Рис. 15), которая осуществляет возврат к главной форме или полный выход из приложения.



*Рис. 15 Код кнопки “Закрыть”*

Этап реализации при наличии анализа предметной области и пользовательского интерфейса заключается в применении накопленных знаний и наработок в области программирования. В современном мире, чтобы оставаться конкурентоспособным, нужно постоянно расширять свои навыки и самообучаться, изучая и применяя те многочисленные возможности, которые доступны в среде разработки.

**2.3. Тестирование приложения**

Программный продукт перед внедрением необходимо тщательно проверить на предмет ошибок разного характера. Не всегда программа работает так, как это первоначально задумывалось. Это означает то, что программа в своей структуре имеет ошибки. Процесс обнаружения и устранения ошибок называется отладкой.

Различают:

1. синтаксические ошибки - ошибки, фиксируемые интерпретатором при выполнении синтаксического и частично семантического анализа программы,
2. ошибки компоновки - обнаруженные компоновщиком ошибки при объединении модулей программы,
3. ошибки выполнения - ошибки, обнаруженные операционной системой, аппаратными средствами или пользователем при исполнении программы.

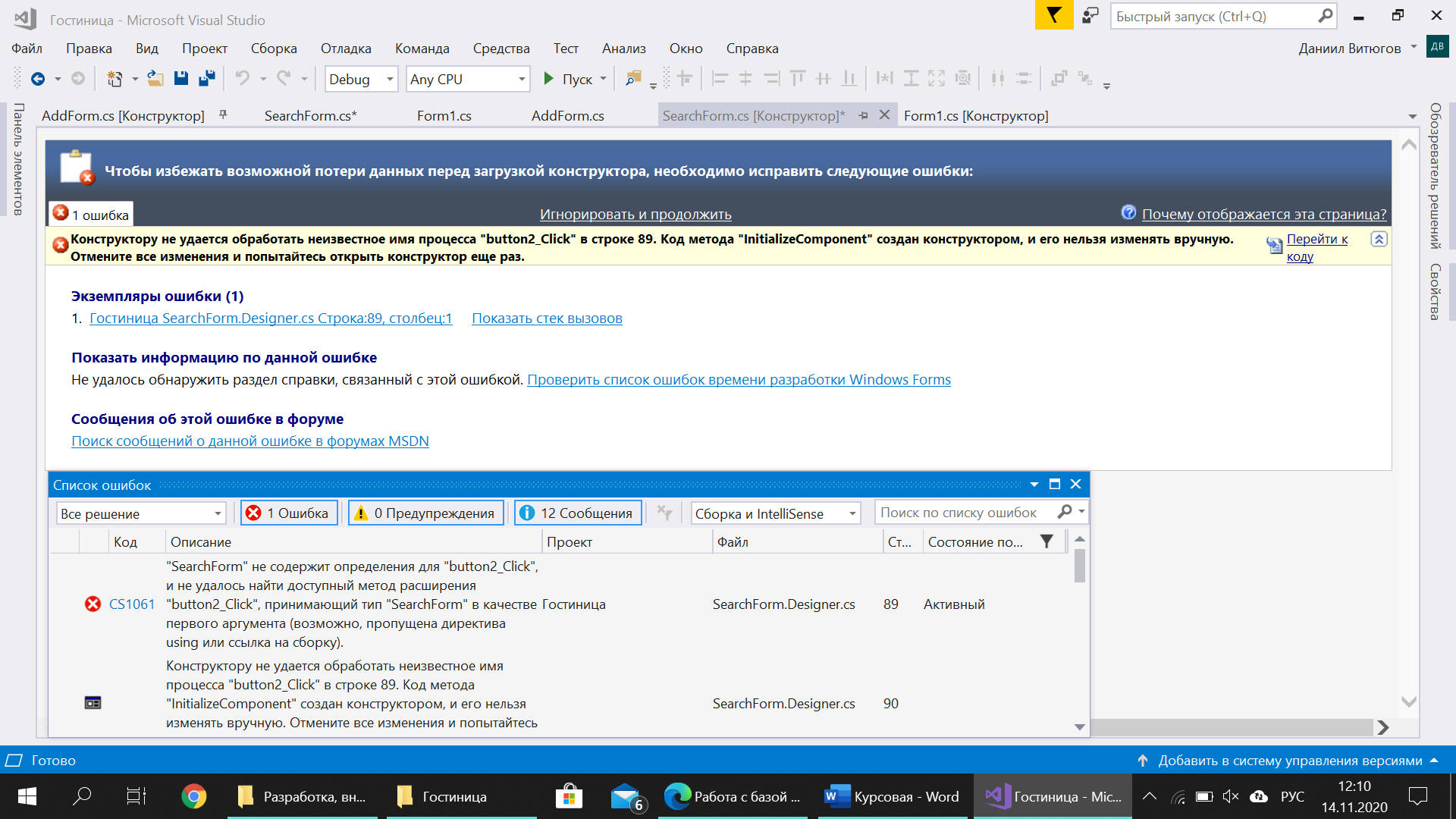
Синтаксические ошибки являются самыми простыми в решении, потому что синтаксис языка четко конкретизирован, и при наличии ошибки возникает комментарий с указанием ее местоположения.

Visual Studio, где создавался проект, включает эффективный интегрированный набор средств сборки и отладки проектов. Ошибку можно определить по подчеркиванию красной волнистой линией. Простым способом быстрого и эффективного устранения ошибок в коде C# является использование быстрых действий, доступных с помощью значка лампочки или отвертки. Во вкладке Список ошибок окна Вывод сборки можно увидеть четкое описание проблемы.

Ошибки синтаксиса, возникшие при создании проекта:

1. Смешивание операций проверки на равенство == с операцией присваивания;
2. Отсутствие оператора break в структуре оператора switch.

Ошибки компоновки связаны с проблемами, обнаруженными при работе со ссылками на другие модули. Во время работы с формой Поиска случайно был удален фрагмент кода, отвечающий за кнопку закрытия. При переходе в конструктор на месте формы появилось сообщение об ошибке с конкретным указанием причины и места. Решением этой проблемы стал возврат к предыдущему действию с помощью сочетанием клавиш Ctrl + Z (отмена).



*Рис. 16 Отсутствие определения для “button2\_Click”*

В процессе тестирования наибольшее количество ошибок было связано с выполнением. Причины таких ошибок различны, что затрудняет процесс отладки. В процессе тестирования в элементы форм подставлялись разнообразные значения, что позволило найти слабые стороны кода.

Первой ошибкой стало введение прошедшей даты в поле дата заезда. Решением данной проблемы стала постановка условия, где на экран появляется сообщение об ошибке введенных данных, если дата заезда уже прошла. Также при тестировании программы была обнаружена ошибка при работе с датами. В случае если дата отъезда раньше даты заезда, то при вычислении количества дней проживания образуется отрицательное число. Условие в dateTimePicker1 и dateTimePicker2 осуществляет проверку введенных дат (см. Рис. 17). Если имеется ошибка при введении дат, то на экране появляется сообщение о неправильности введения данных.

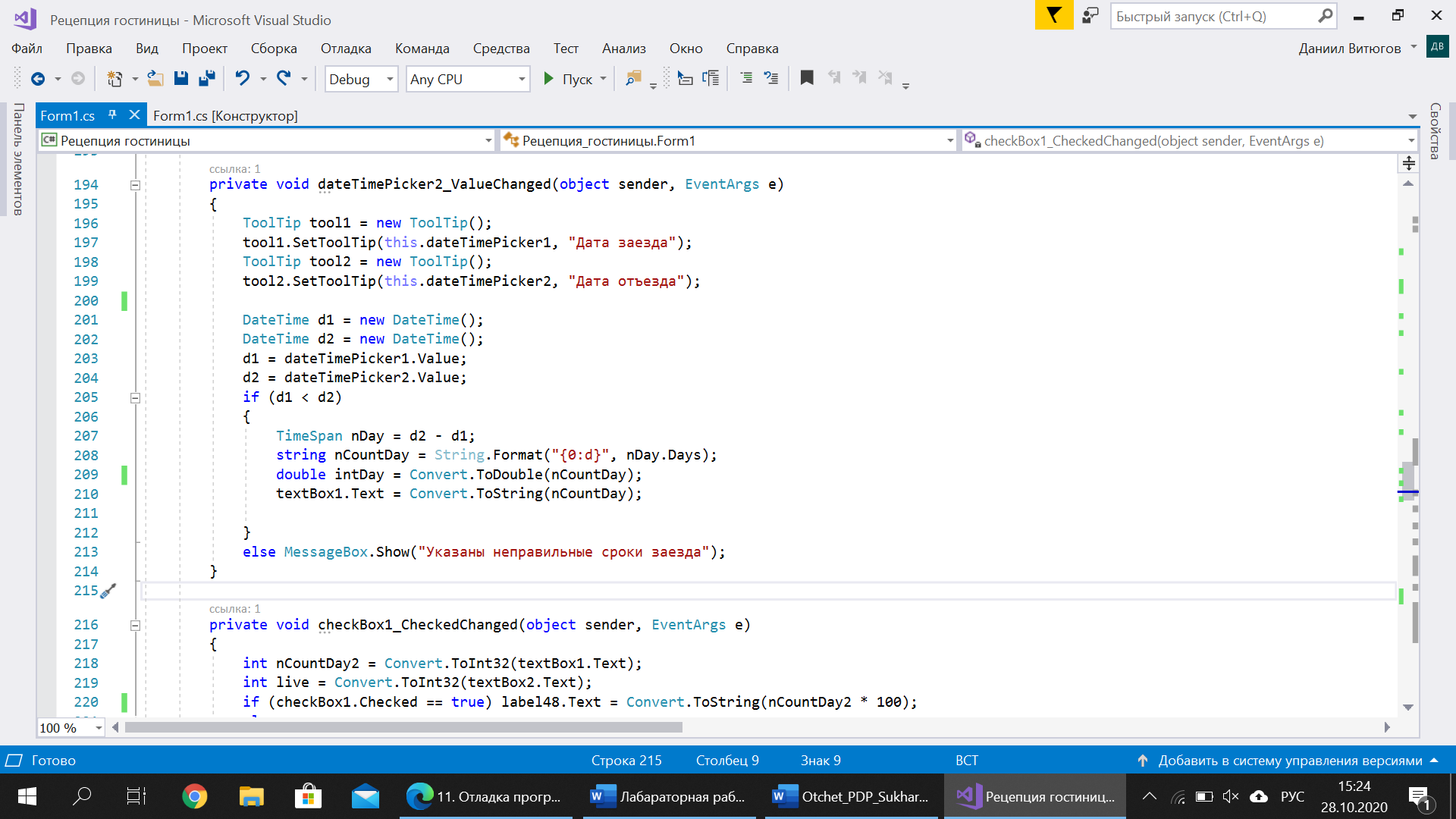


Рис. 17 Фрагмент кода “Проверка введенных дат”

При нажатии кнопки Готово происходит отправка сведений в базу данных, и если пользователь пропустил одно из полей, то соответствующая ячейка в таблице оставалось пустой. В условии функционирования гостиницы это могло привести к сбою в работе с данным клиентом. Чтобы этого избежать, добавлен фрагмент кода (см. Рис. 18), отвечающий за проверку заполненности полей.

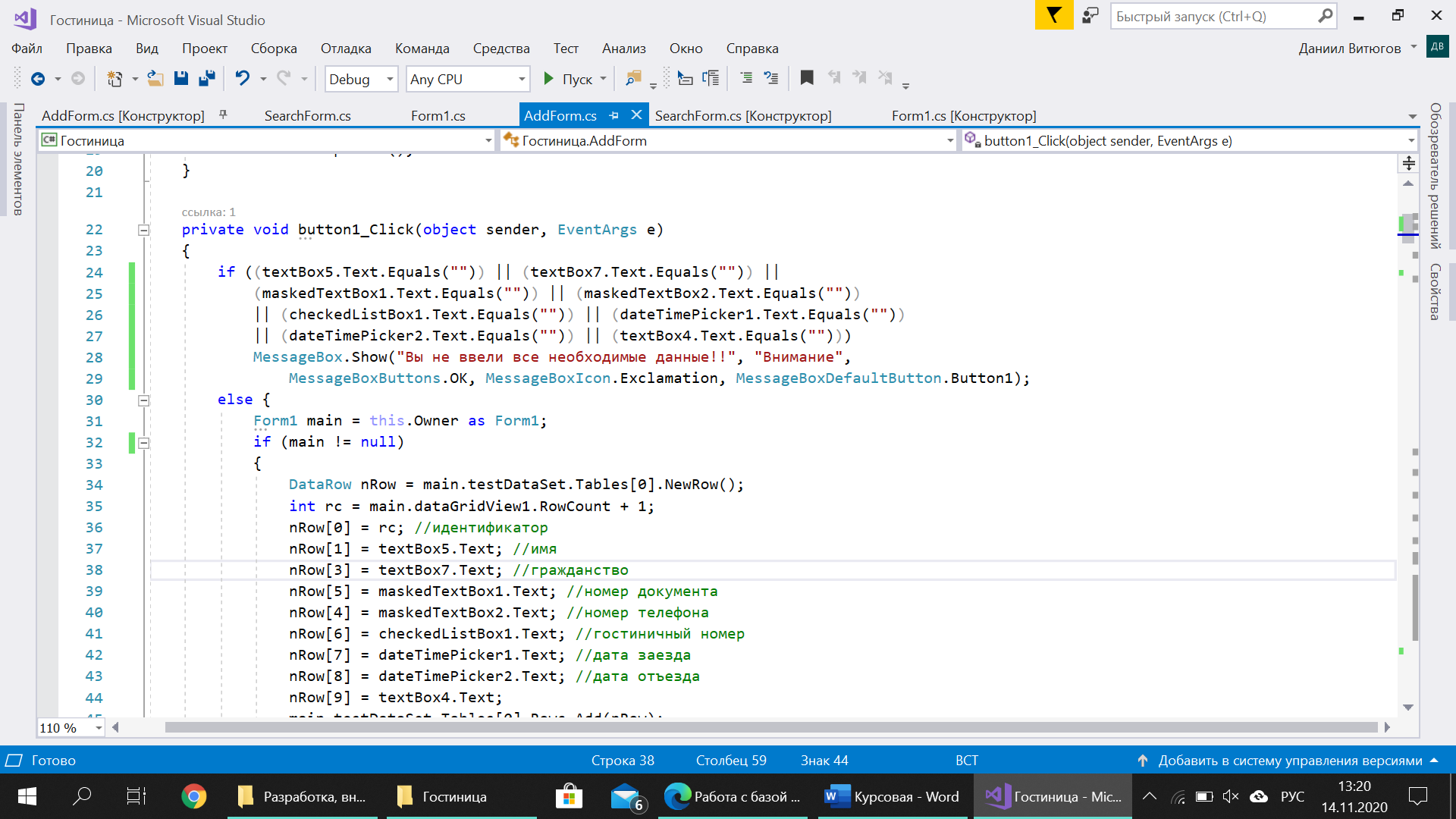


Рис. 18 Фрагмент кода “Проверка заполненности полей”

С целью минимизации ошибок при введении контактного номера, номера и серии паспорта гражданина России, textbox был заменен на maskedtextbox. Данный элемент позволяет создать маску при введении стандартизированных по форме представления значений. Выбирая свойство Mask, перед разработчиком возникает окно (см. Рис. 19), где можно выбрать маску из списка, или же задать собственную.

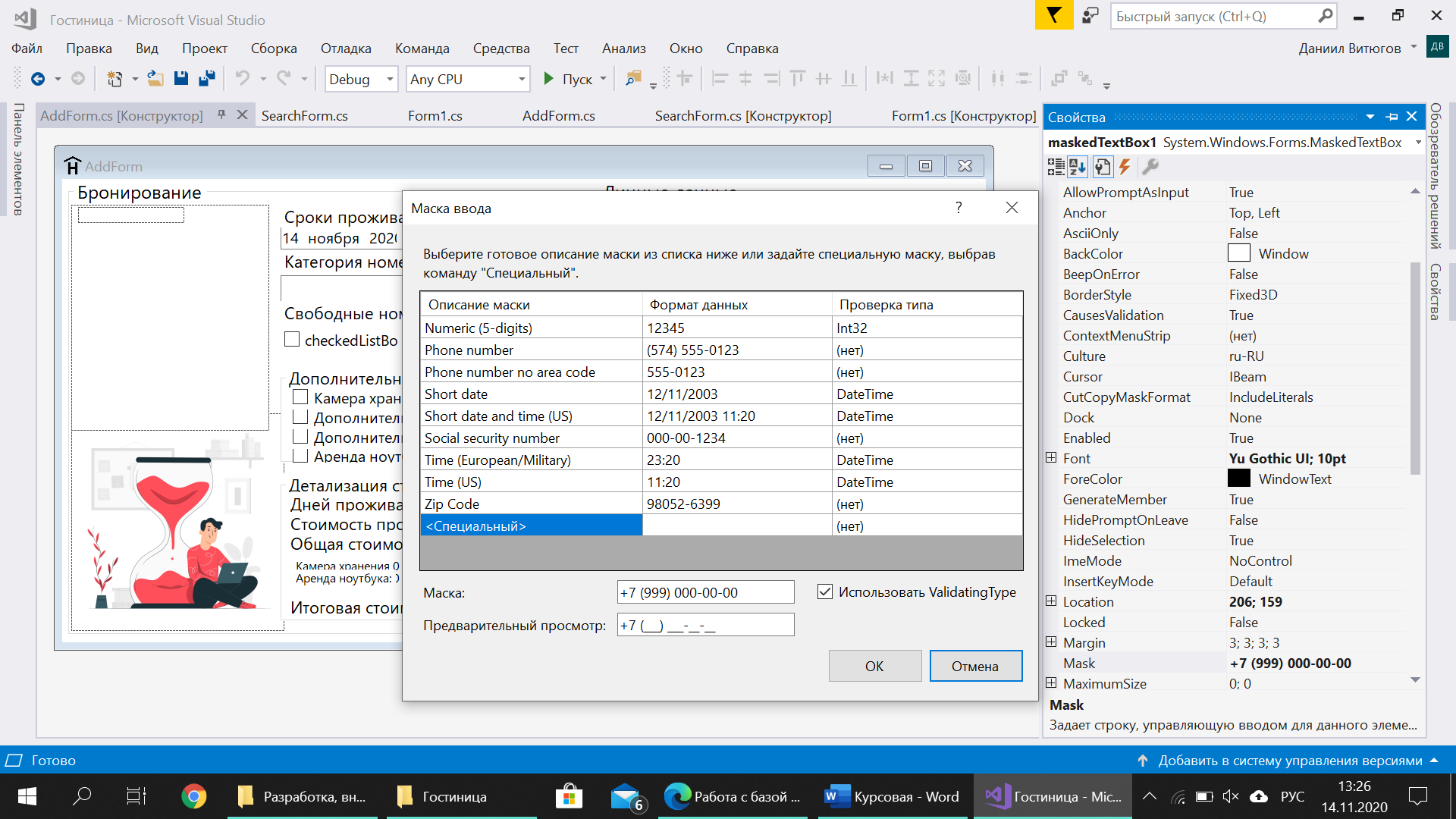


Рис. 18 Фрагмент кода “Проверка заполненности полей”

## **Заключение**

В результате курсового проекта была спроектирована и реализована объектно-ориентированная модель информационной подсистемы отраслевой направленности на примере гостиницы. Для достижения данного результата было выполнено изучение предметной области.

Получившаяся система соответствует всем требованиям, которые были представлены в задании и выполняет большую часть функций, необходимых сотрудникам гостиницы.

В настоящее время внедрение информационных систем является неотъемлемой частью любого предприятия. Повышается уровень управления, так как внедрение информационных технологий способствует получению эффективных вариантов решения задач. За счет автоматизации повторяющейся работы, облегчается деятельность сотрудников. Объем информации на бумажных носителях стремительно снижается, что позволяет организовать быстрый обмен данными между персоналом. Все это уменьшает общие затраты на производство.

В процессе разработки программное обеспечение проходит ряд этапов, которые тесно связаны между собой. Создание автоматизированной информационной системы не ограничивается лишь написанием программного кода. Это является большой работой первоначально по анализу области, по которой предстоит создание системы, а затем по применению знаний в программировании. Симбиоз этих компонентов способен принести организации нужный результат.

Завершение разработки программного обеспечения состоит в доведении характеристик до заданных требованиями технического задания. Это выполняется на этапе тестирования и отладки. Внесение искаженных исходных данных не должно являться причиной отказа в работе программного обеспечения.

## **Список используемой литературы**

**Используемая литература:**

1. Боуэн Док., Котлер Ф., Мейкенз Дж. Маркетинг. Гостеприимство. Туризм. 2005. - 1063 с.
2. Блинков Ю. А. Проектирование информационных систем, - Саратов: Саратовский государственный университет, [б/г]. - 377 с.
3. Волков Д.И., Волков П.Д. Концепция проекта корпоративного научно-образовательного портала. М., 2006. - 39 с.
4. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем. Ростов н/Д.: Феникс, 2009. - 512 с.

**Сайты в интернете:**

1. Студенческая онлайн библиотека Studbooks. Разработка программного модуля для формирования плана изготовления продукции на предприятии [Электронный ресурс] URL: <https://studbooks.net/2175003/informatika/opisanie_protsessa_otladki_programmy> (дата обращения 23.10.2023).
2. Отладка программного обеспечения. [Электронный ресурс] URL: <https://studopedia.ru/3_175688_otladka-programmnogo-obespecheniya.html> (дата обращения 29.10.2023).