Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА**

Отчет по лабораторной работе №3

по дисциплине «Новые технологии в программировании»

Вариант 1

Студент гр. 588-2

\_\_\_\_\_\_\_\_\_\_\_\_ Робканов.К.Д

(подпись)

« 08 » мая 2021г.

Руководитель старший научный сотрудник, доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_ Горяинов А.Е.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_2021 г.

Томск 2021

**Содержание**

1 Введение…………………………………………………………………………3

2 Основная часть…………………………………………………………………..4

Заключение……………………………………………………………………….10

**1 Введение**

**Цель работы:** изучить разработку адаптивного пользовательского интерфейса десктоп-приложения.

**Задачи:**

1. Изучить использование фреймворка пользовательского интерфейса Windows Forms и его компоненты.
2. Освоить компоненты и свойства для создания адаптивного дизайна.
3. Освоить обработку событий компонентов Windows Forms.
4. Реализовать передачу данных между разрабатываемыми формами и пользовательскими элементами управления.
5. Реализовать защиту от некорректного ввода.

**2 Основная часть**

Вид главной формы приведен на рисунке 2.1.

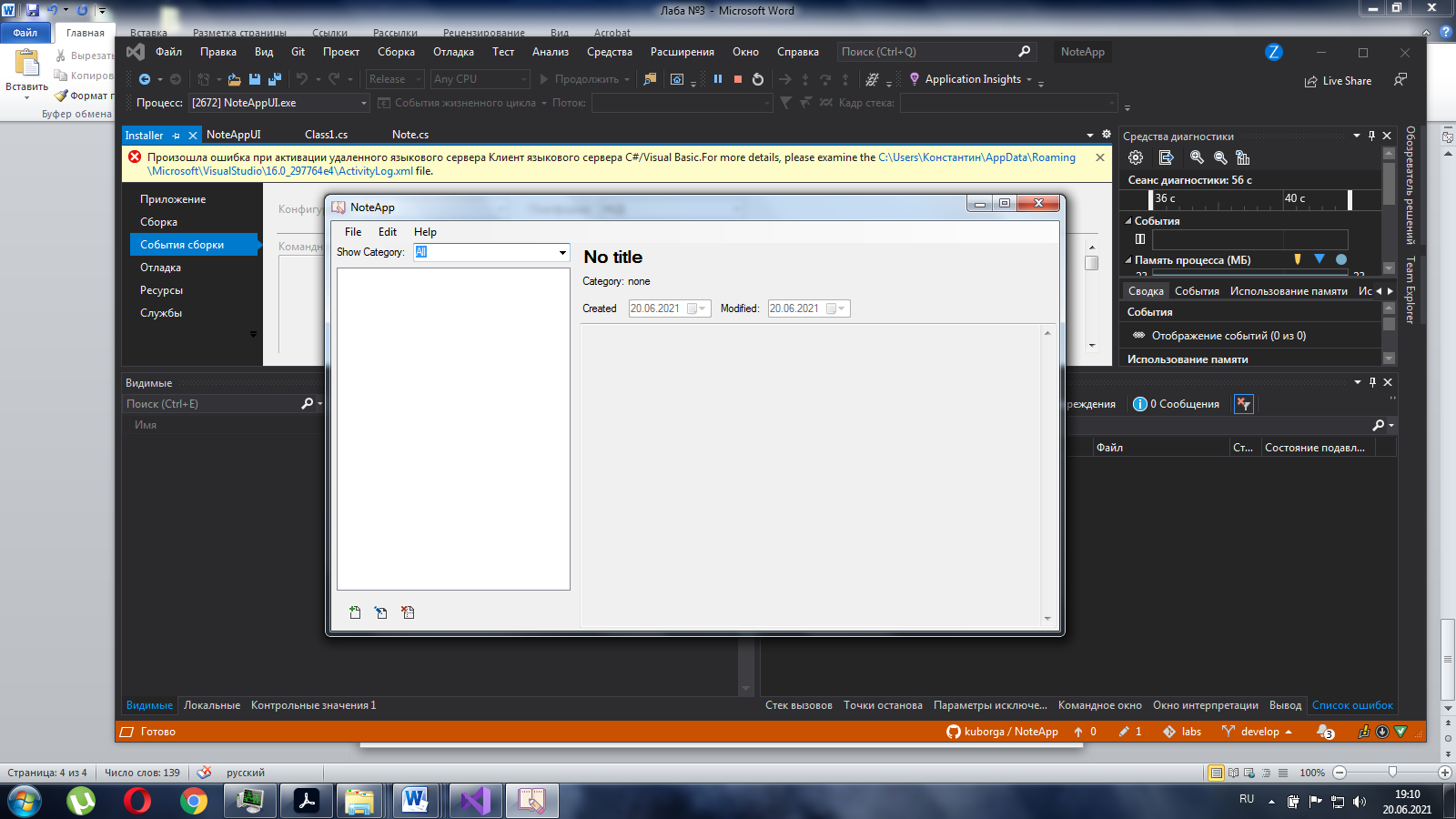


Рисунок 2.1 – Скриншот главной формы MainForm

В данной форме в верхней строке расположены три кнопки с выпадающим меню:

* *File* содержит вложенное поле *Exit*, при нажатии на которое программа закрывается
* *Edit* содержит три поля: *Add Note, Edit Note, Remove Note* — создание, редактирование и удаление заметки соответственно
* *Help* содержит поле *About*, при нажатии на которое открывается форма с информацией о разработчике

Show Category представляет собой выпадающий список со всеми категориями заметок. Ниже располагается NotesListBox, которые содержит перечень всех существующих заметок в виде их названий. Справа блок, который показывает содержимое выбранной заметки: ее название, дату создания и изменения, категорию и текст.

Под NotesListBox располагаются три кнопки, которые также добавляют, редактируют или удаляют выбранную заметку соответственно.

Форма редактирования и добавления заметок представлена на рисунке 2.2.

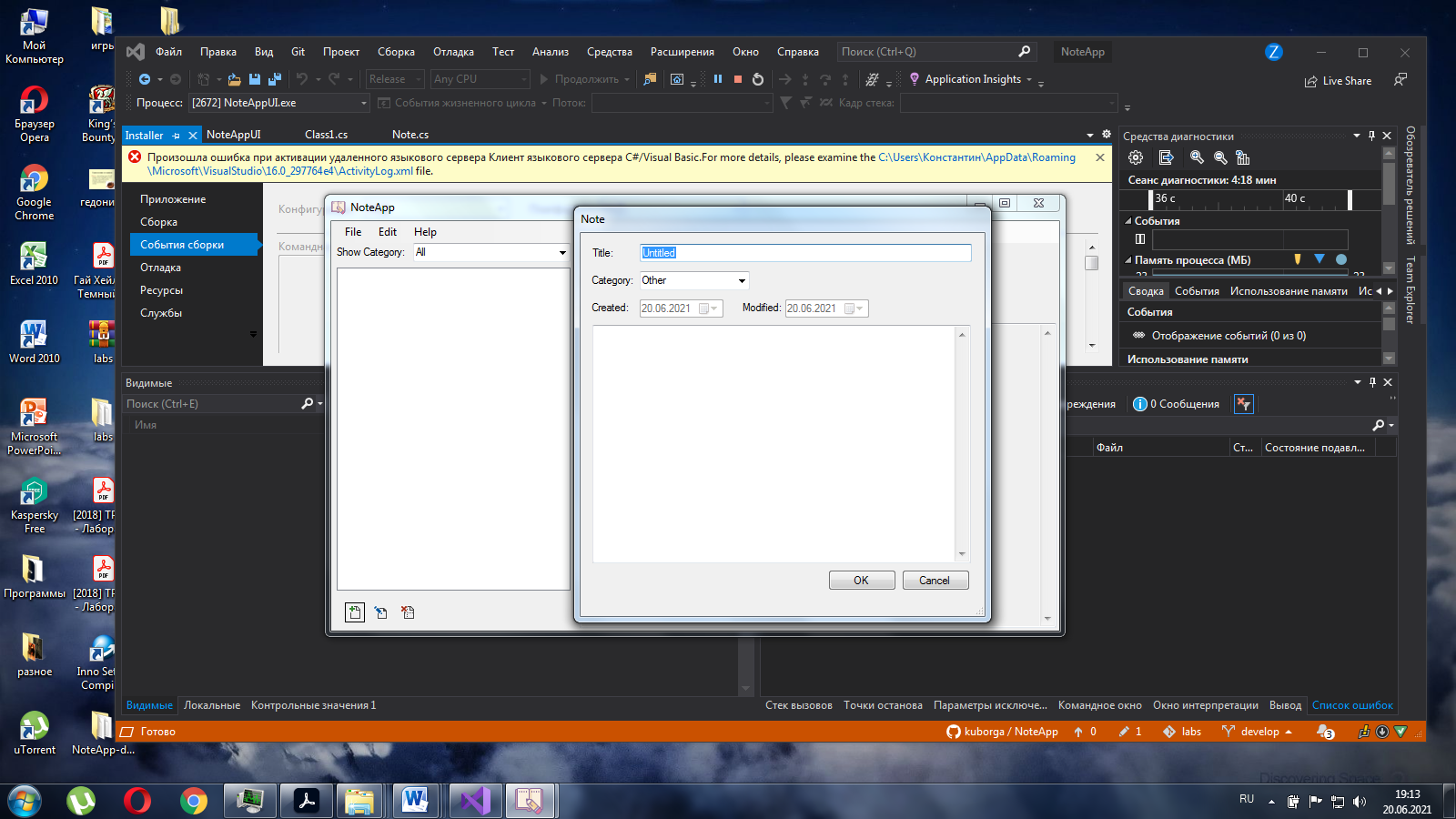


Рисунок 2.2 – Скриншот формы для добавления и редактирования заметок

В поле Title вводится название заметки. Оно защищено проверкой ввода. Сначала первая проверка выделяет поле красным цветом при длине названия больше 50 символов (рисунок 2.3).

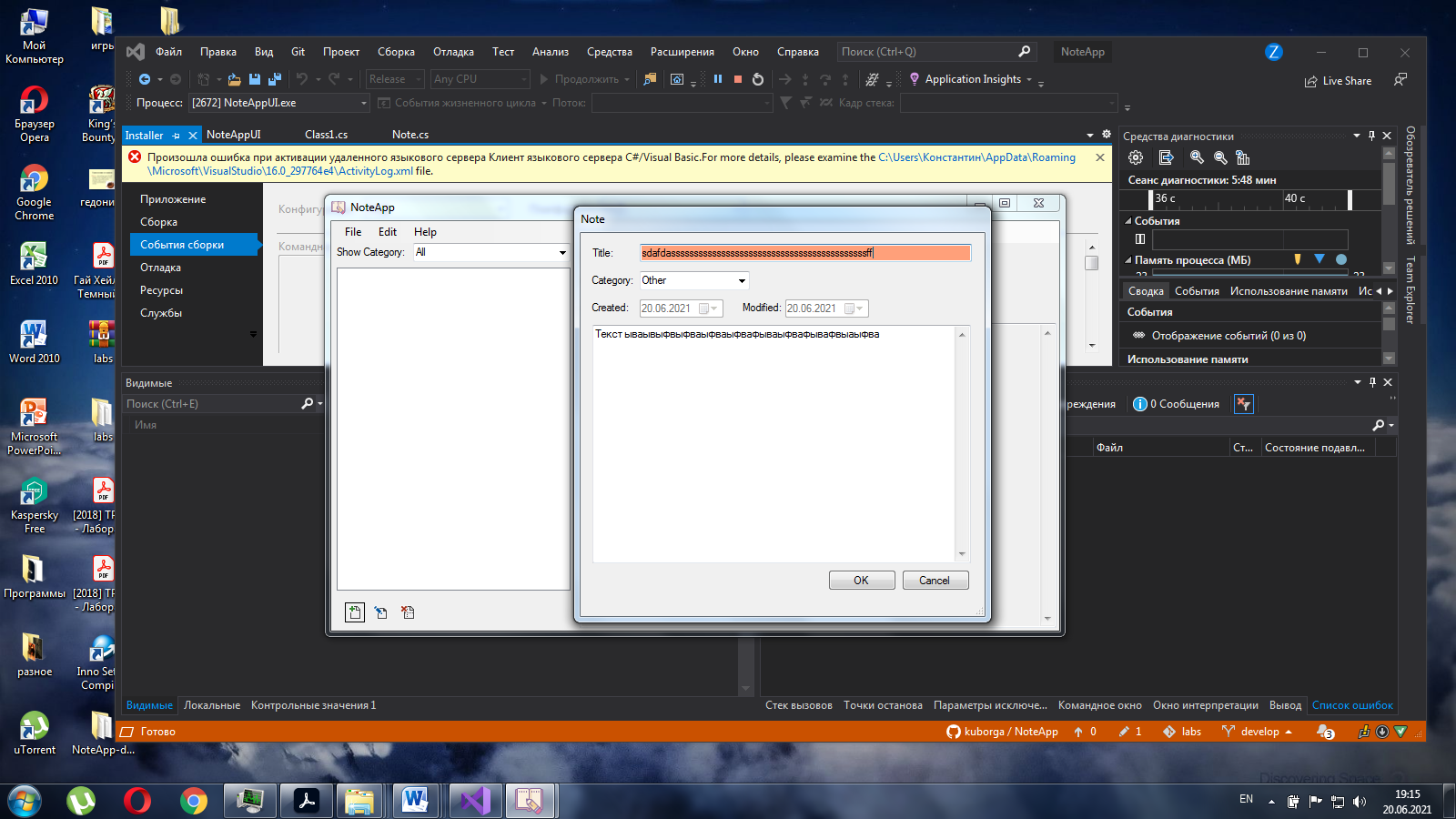


Рисунок 2.3 – Скриншот работы защиты от ввода больше 50 символов

Если пользователь не обратил на это внимание, то при нажатии кнопки «ОК» программа высвечивает окошко с неправильным значением выделенного поля и предлагает пользователю его изменить, иначе не позволяет сохранить созданную заметку (рисунок 2.4)

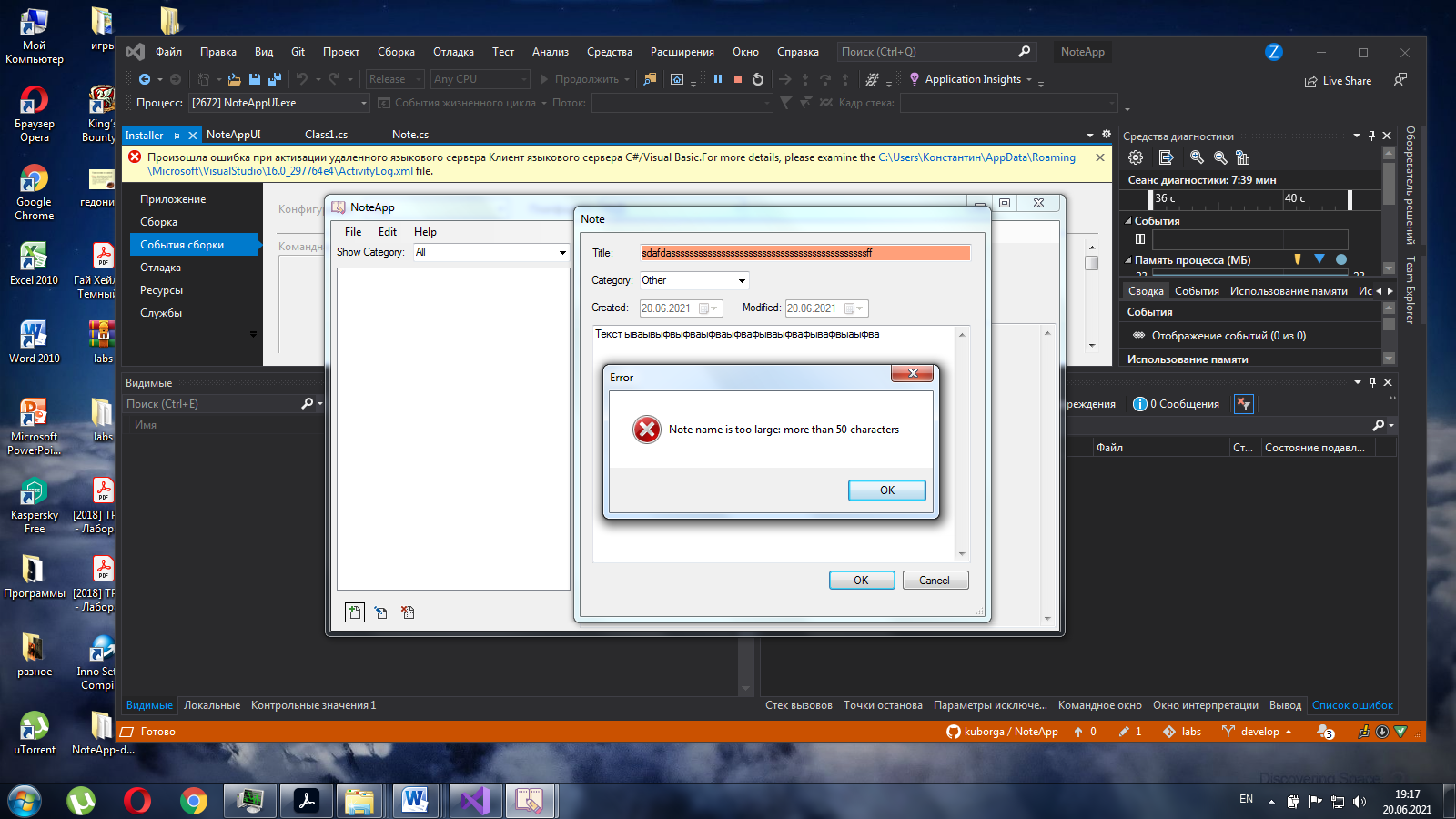


Рисунок 2.4 – Окно с сообщением о неправильном вводе значения

Поле Category дает пользователю выбрать категорию своей будущей заметки или изменить ее для старой. Поля Created и Modified открыты только для чтения. В них заносится время создания и изменения заметки соответственно. Поле Text предназначено для ввода содержимого заметки.

Содержимое формы c информацией о разработчике представлено на рисунке 2.5.

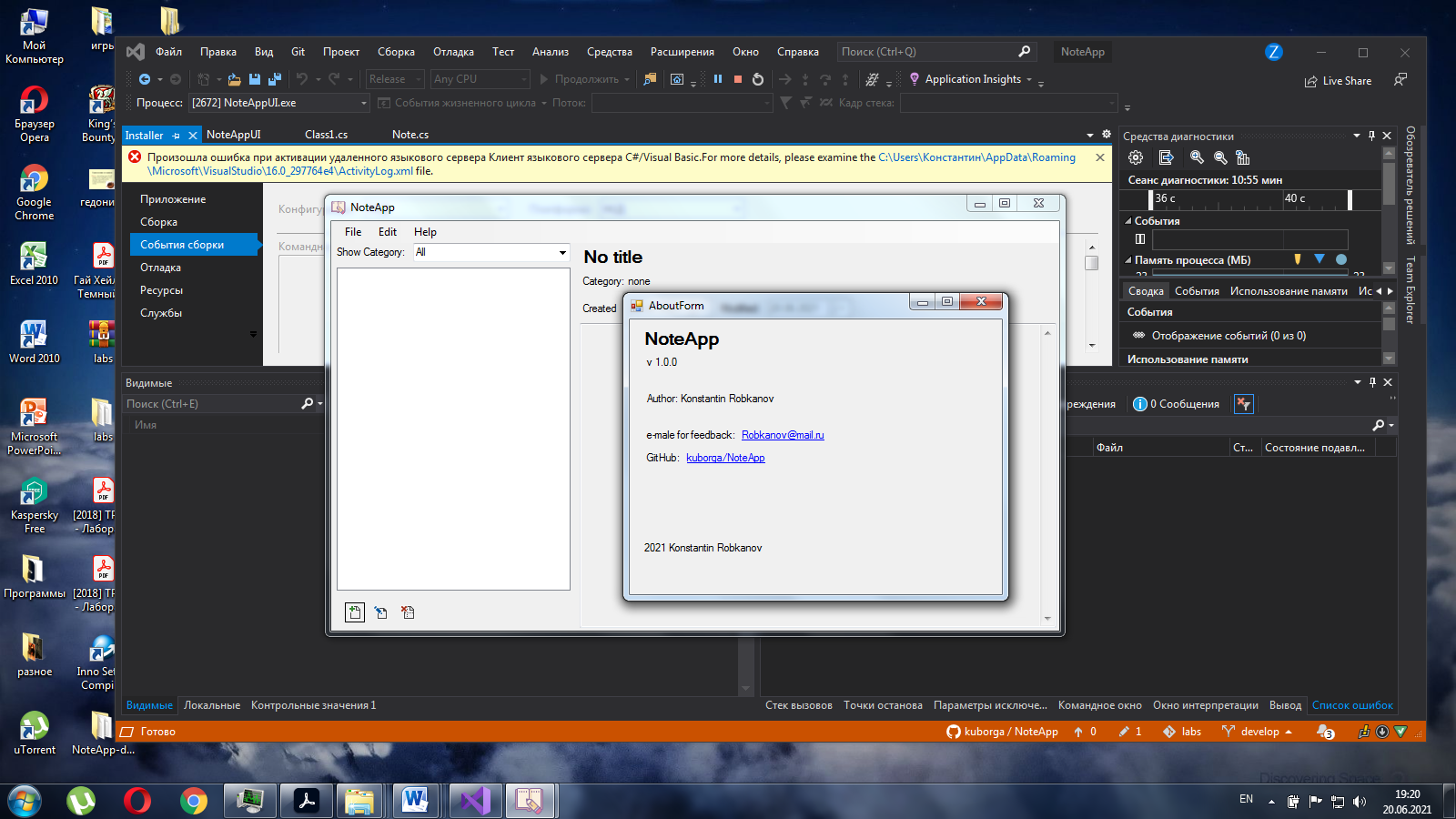


Рисунок 2.5 – Скриншот формы About

Данная форма только для чтения и для перехода на почту или профиль GitHub для связи с разработчиком приложения. Окно зафиксировано в размерах и не имеет кнопок сворачивания или разворачивания в полноэкранный режим.

Пример создания окна About

var formAbout = new AboutForm();

Функция AddNote() создаёт новый экземпляр класса Note, затем передаёт данные новосозданного экземпляра в окно редактирования заметки. Затем, если пользователь нажмёт кнопку «ОК» формы редактирования заметки, рассматриваемая функция заберёт данные из формы редактирования назад и сохранит их в список всех заметок. А также добавит новую заметку в список непосредственно в самой форме, для возможности её выбора пользователем. В завершении список заметок будет сохранён в файл.

private void AddNote()

{

var note = new Note();

var noteForm = new NoteForm();

noteForm.Note = note;

noteForm.ShowDialog();

if (noteForm.DialogResult == DialogResult.OK)

{

note = noteForm.Note;

//Добавляет заметку в реальный список

\_project.Notes.Insert(0, note);

//Добавляет заметку в ListBox

NoteListBox.Items.Insert(0, note.Title);

//Добавляет заметку в отображаемый список

\_viewedNotes.Insert(0, note);

NoteListBox.SelectedIndex = 0;

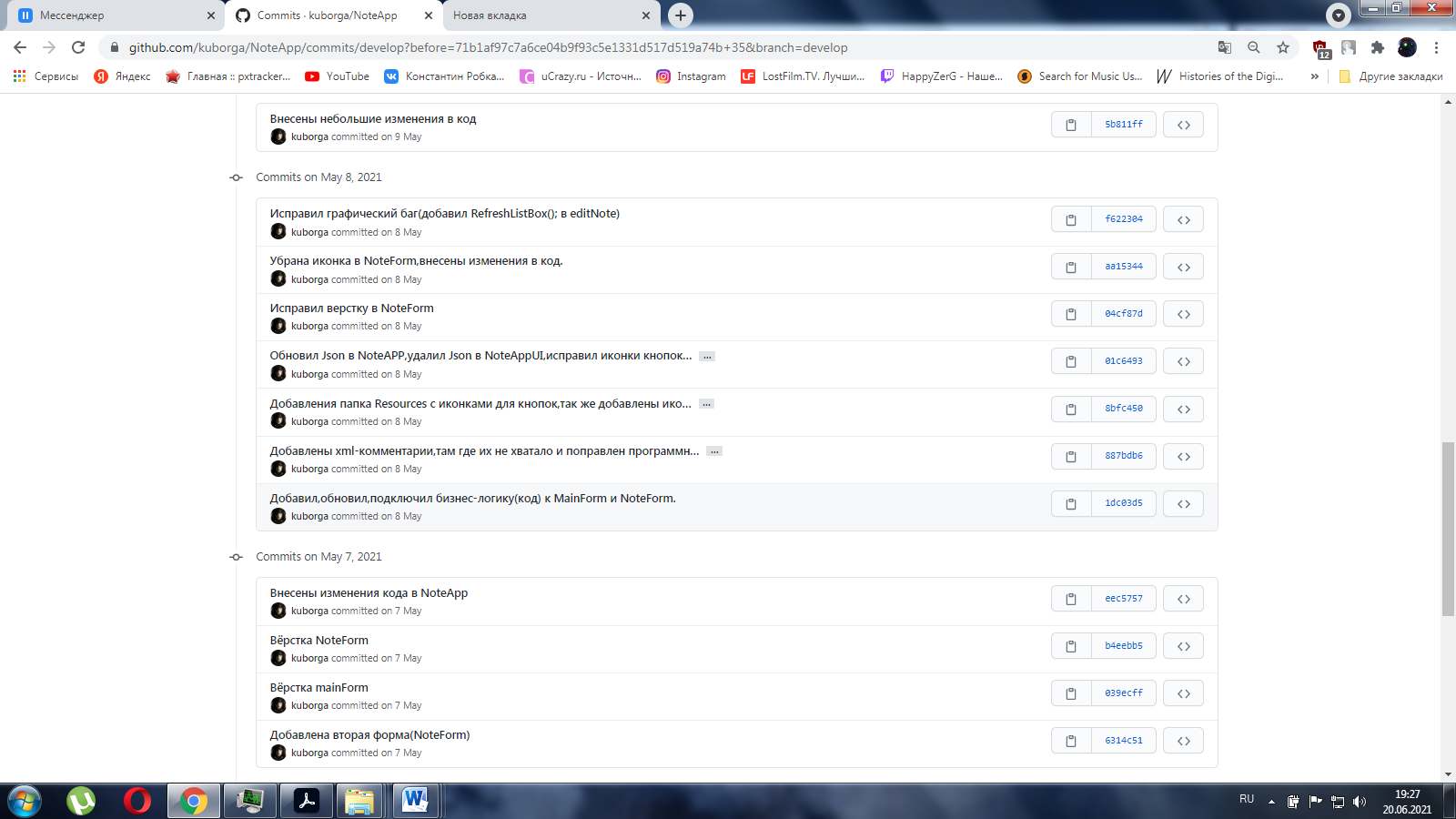
RefreshListBox();

ProjectManager.SaveToFile(\_project, ProjectManager.DefaultPath);

}

}

По итогу выполнения лабораторной работы история коммитов в репозитории на сервисе GitHub стала выглядеть следующим образом (рисунок 2.6).

Рисунок 2.6 – История коммитов в репозитории

**Заключение**

В ходе лабораторной работы изучена разработка адаптивного пользовательского интерфейса десктоп-приложения..