МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського  
 “Харківський авіаційний інститут”

Факультет економіки і менеджменту

Кафедра інженерії програмного забезпечення

**Лабораторна робота № 1**

З Інформаційних технологій розробки ПЗ

на тему: «Робота в команді, інтеграція проекту в систему контролю версій»

Виконав студент: 4-го курсу групи 641п

050112 (програмна інженерія)

Василевський А.А

Прийняв:

Національна Шкала: \_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_

Харків 2017

# **Цель работы**

Сформировать проектную команду. Научиться администрировать систему контроля версий, распределять и отслеживать задачи по проекту. Научиться интегрировать проект в систему контроля версий. Получить навыки практического использование клиента TortoiseHG.

# **Задание**

1. Разбиться на команды по 3-5 человек.
2. Выделить в команде роли (тимлид, программисты, тестировщик).
3. Разработать простое приложение, которое содержит в себе несколько классов, при этом каждый класс должен быть реализован отдельным программистом, а интерфейсы для каждого класса описываются тимлидом.
4. **Тимлиду**:
   1. Зарегистрироваться на ресурсе Bitbucket (bitbucket.org)
   2. Создать Репозиторий (который будет использован для выполнения 1й лб.).
   3. Создать Группу пользователей.
   4. Добавить в группу пользователей членов команды и назначить им полные права доступа.
   5. Создать солюшин с проектом, который содержит описание интерфейсов классов.
5. **Программистам**:
   1. Установить TortoiseHG (<http://tortoisesvn.net/>)
   2. Получить копию репозитория на локальный компьютер.
   3. Добавить в солюшин проекты, реализующие интерфейс и вкомитить его в репозиторий.
   4. Описать классы, реализующие описанные интерфейсы.
   5. Закоммитить и зафиксировать все изменения на сервере.
6. **Тестировщику**:
   1. Подключиться к репозиторию проекта.
   2. Извлечь из репозитория на компьютер рабочую копию проекта.
   3. Проверить функциональность и взаимодействие интерфейсов. Устранить баги.
   4. Залить обратно проект в svn.
7. Составить матрицу ответственных за выполнение проекта.
8. Составить отчёт.

# Ход роботы

Для выполнения данной лабораторной работы был создан аккаунт (Nekodelick) на ресурсе GitHub (рис.1).

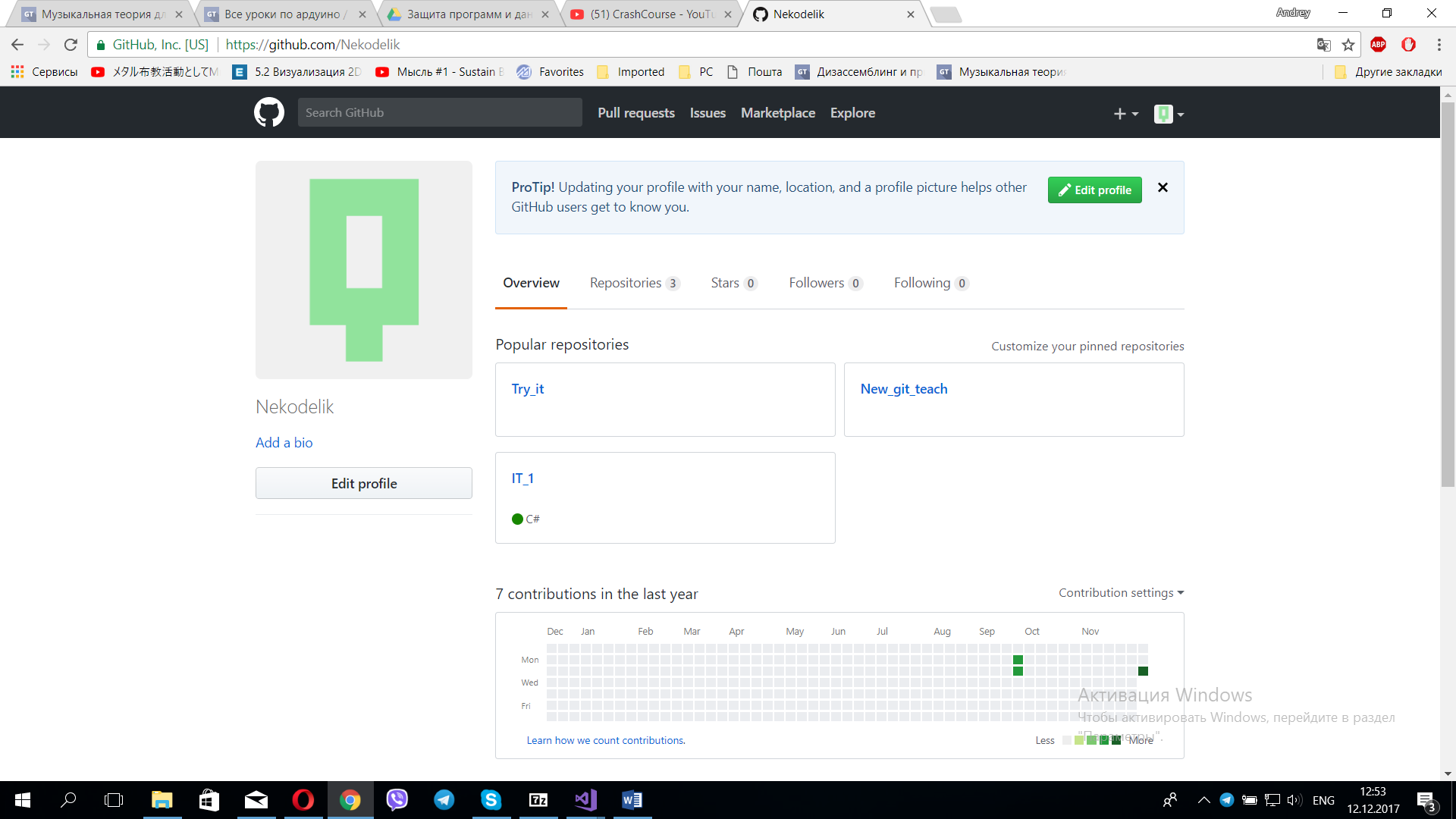


Рисунок 1 – Информация об аккаунте

Для загрузки файлов на сервер использовалось Расширение GitHub Extension for Visual Studio, представленное на рисунке 2.

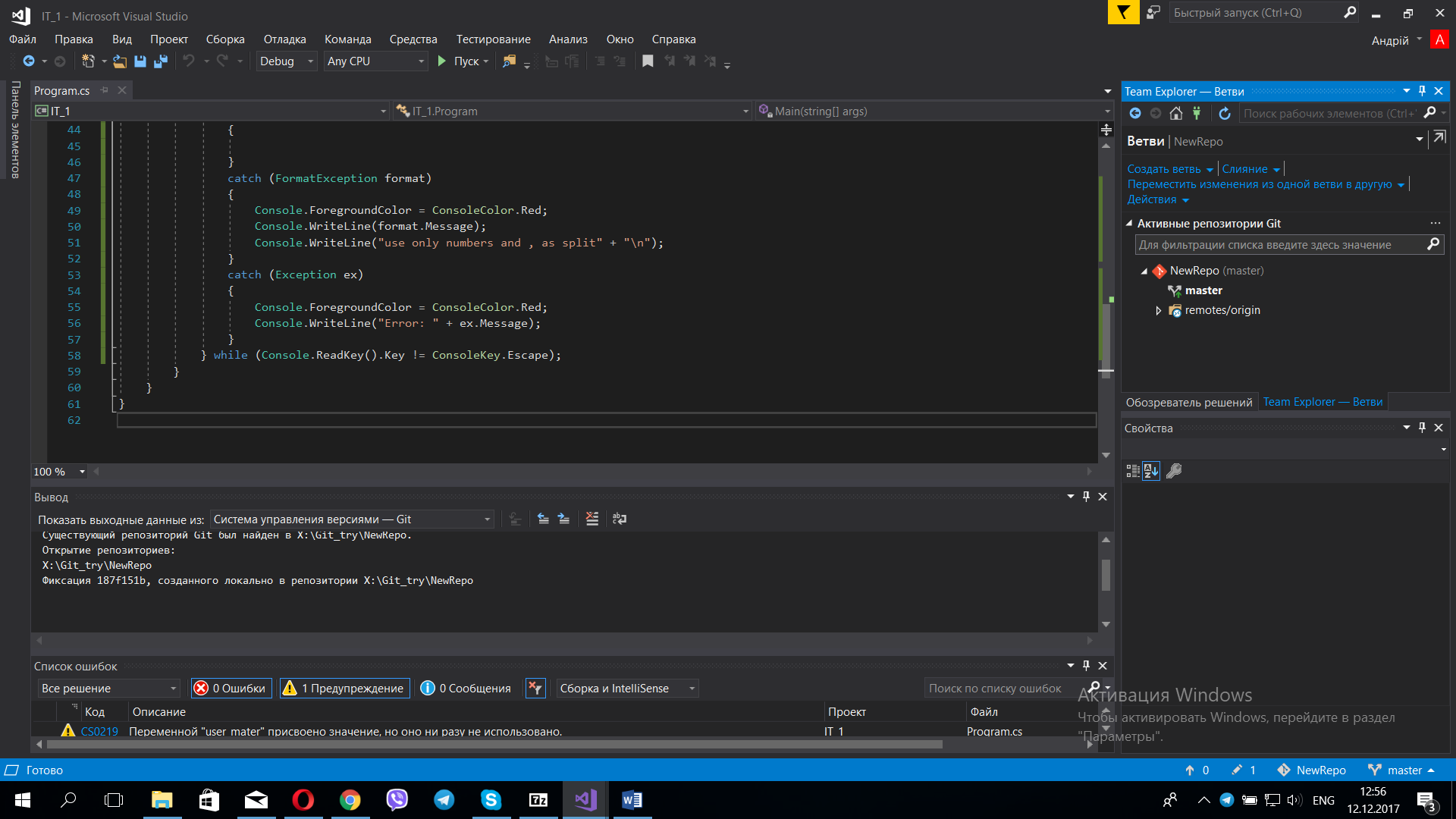


Рисунок 2

Для выгрузки из сервера использовалось приложение Github Decktop, в котором создан новый репозиторий, что клонирует прошлый, рис. 3.

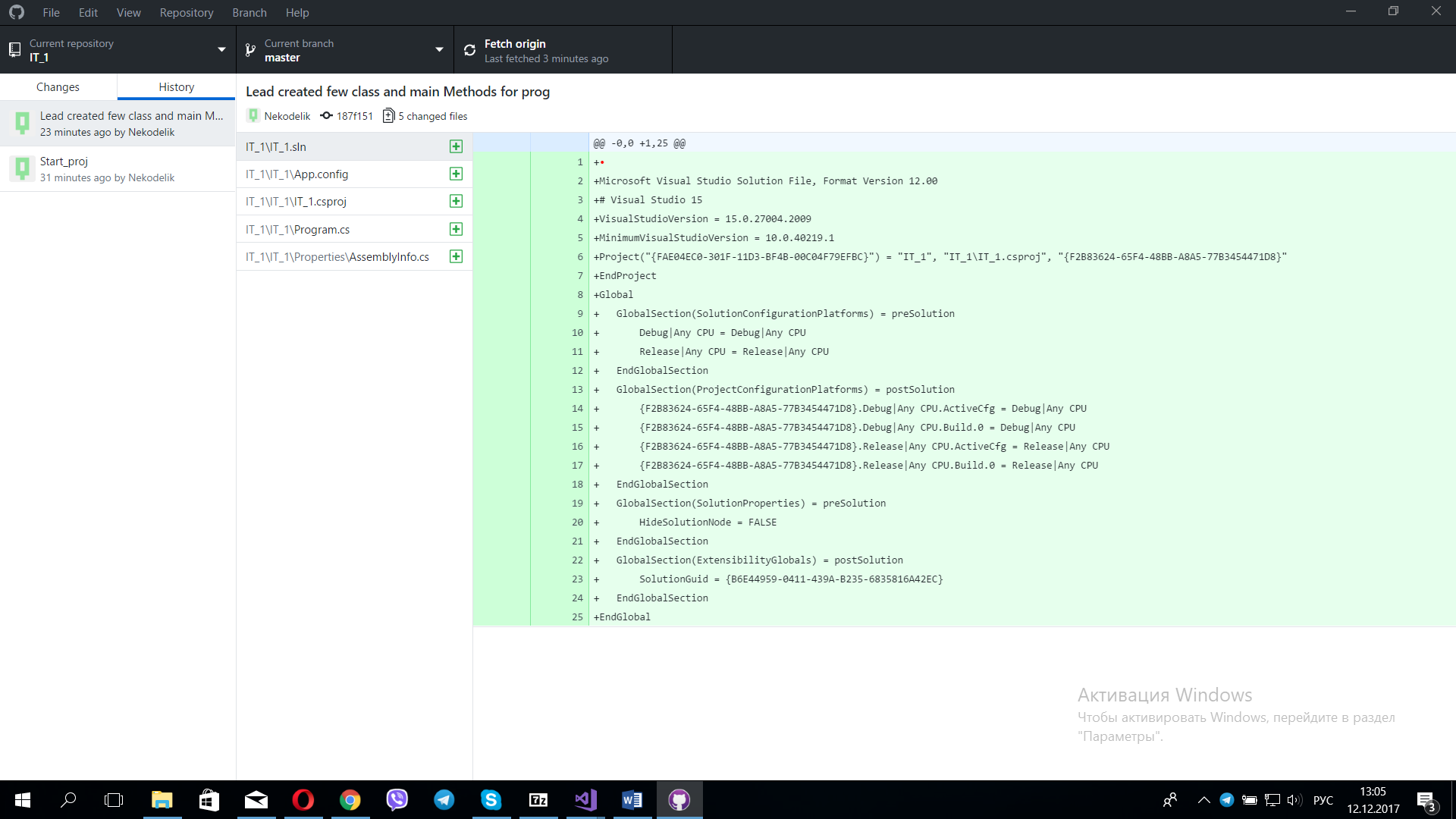


Рисунок 3

Завершаем разработку программы, реализуя методы, заданы в прошлой итерации, коммитим и выгружаем обратно, рис. 4.

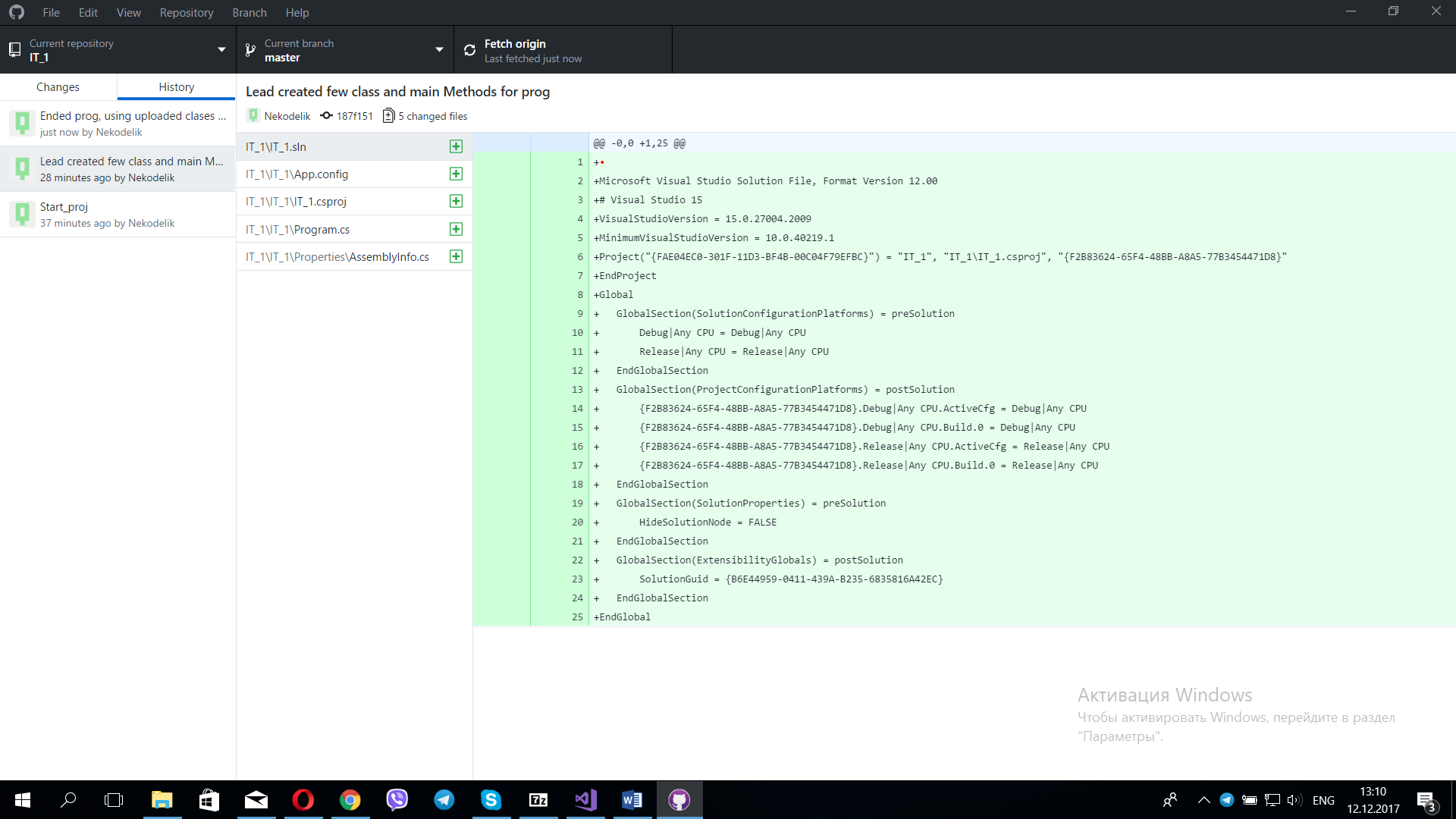


Рисунок 4 — новые коммиты после завершения редактирования кода.

**Финальный код:**

using System;

namespace IT\_1

{

class Price\_counter

{

double u\_electricity\_value;

const double electricity\_price = 0.75;

double Total\_price = 0;

public static double Electricity\_price => electricity\_price;

public double U\_electricity\_value { get => u\_electricity\_value; set => u\_electricity\_value = value; }

public void Math\_price()

{

Total\_price += U\_electricity\_value \* Electricity\_price;

}

public void Math\_price\_over()

{

Total\_price += 2 \* U\_electricity\_value \* Electricity\_price;

}

public void Write\_result()

{

Console.WriteLine("Your electricity price 0,75. Price over 100 is 2 times more");

Console.WriteLine("Your total prise =>>" + Total\_price + "\n");

}

}

class Program

{

static void Main(string[] args)

{

Console.ForegroundColor = ConsoleColor.Magenta;

Console.WriteLine("Press Esc to exit or EnyKey to continuation");

do

{

Console.ResetColor();

Console.WriteLine("Hi!/n plese enter u meter:");

Price\_counter User\_electricity\_price = new Price\_counter();

double user\_mater = 0;

try

{

user\_mater = Convert.ToDouble(Console.ReadLine());

if (user\_mater < 0)

{

Console.ForegroundColor = ConsoleColor.Red;

throw new Exception("Mater cannot be lower than 0" + "\n");

break;

}

Console.WriteLine("u\_mater ==>" + user\_mater.ToString());

if (user\_mater > 100)

{

//electricity before 100

User\_electricity\_price.U\_electricity\_value = 100;

User\_electricity\_price.Math\_price();

//electricity over 100

User\_electricity\_price.U\_electricity\_value = user\_mater - 100;

User\_electricity\_price.Math\_price\_over();

}

else

{

User\_electricity\_price.U\_electricity\_value = user\_mater;

User\_electricity\_price.Math\_price();

}

User\_electricity\_price.Write\_result();

}

catch (FormatException format)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine(format.Message);

Console.WriteLine("use only numbers and , as split" + "\n");

}

catch (Exception ex)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Error: " + ex.Message);

}

} while (Console.ReadKey().Key != ConsoleKey.Escape);

}

}

}

**Робота программы:**



**Матрица ответственности**

|  |  |  |
| --- | --- | --- |
|  | **Проектная команда** | |
| **Виды работ** | **Василевский** | **Василевский** |
| Распределение задач | R | S |
| Проектирования классов и методов | R | S |
| Обьявления классов и методов | S | R |
| Реализация классов и методов | S | R |
| Тестирование | S | R |
| \*R – ответственный  S – поддерживает /участвует |  |  |

**Выводы**

В ходе выполнения лабораторной роботы были изучены и получены навыки пользования серверной и клиентской части GitHub. А также способы добавления просмотра коммитов, создания и клонирования репозиториев. Загрузку и выгрузки информации на из сервера.