

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Казанский национальный исследовательский технологический университет»

(ФГБОУ ВО «КНИТУ»)

Институт Нефти, химии и нанотехнологий

Факультет Наноматериалов и нанотехнологий

**Лабораторная работа по дисциплине  
 «Методы оптимизации»**

на тему: «Метод поразрядного приближения»

**Выполнил**: студент группы 4381-22, Тимушев Даниил Анатольевич

**Проверил:** ДТН, профессор кафедры ИСУИР Герасимов Александр Викторович

Казань 2021

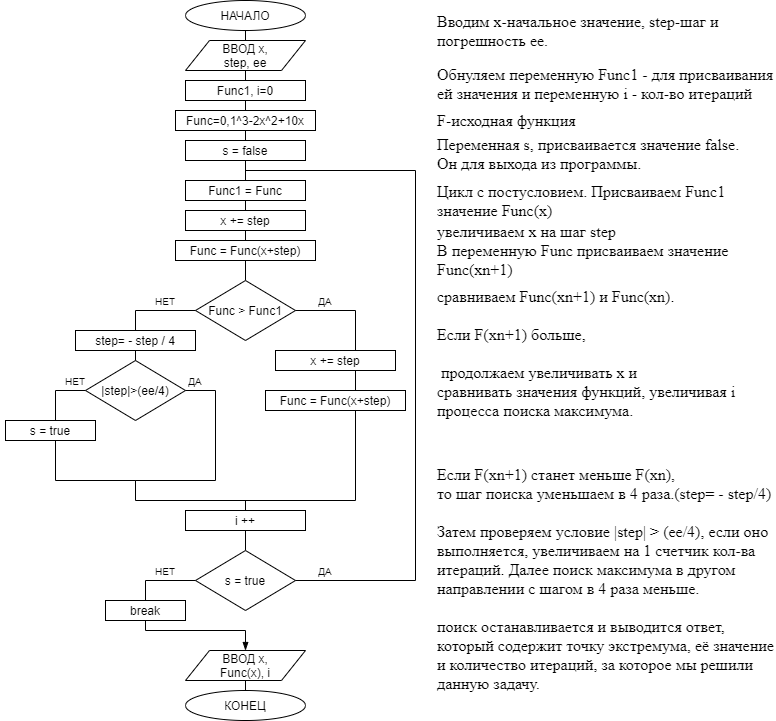
ПОСТАНОВКА ЗАДАЧИ

Метод поразрядного приближения является разновидностью метода равномерного поиска и реализуется следующим алгоритмом.

1. Задаем начальное приближение *x*=*x0*слева от максимума Func (*x*) и вычисляем Func (*x0*). Задаем D=step, где step= Δ*x –* начальный шаг поиска.
2. Полагаем Func1= Func (*x*n), где в начале Func (*x*n) = Func (*x0*), задаем *x*=*x+D* и вычисляем Func (*x*n+1) = Func (*x*).
3. Проверяем условие Func (*x*n+1)> Func1; если оно выполняется, идем к п.3, если нет – к п.4.
4. Полагаем D= -D/4. Проверяем условие |D| > E/4, где Е-заданная погрешность вычисления *x*m в точке максимума. Если оно выполняется, идем к п.2, т.е. обеспечиваем поиск максимума в другом направлении с шагом в 4 раза меньше прежнего. Если данное условие выполняется, заканчиваем счет.

Требуется найти максимум функции 0,1*x*^3-2*x*^2+10*x* .

АЛГОРИТМ

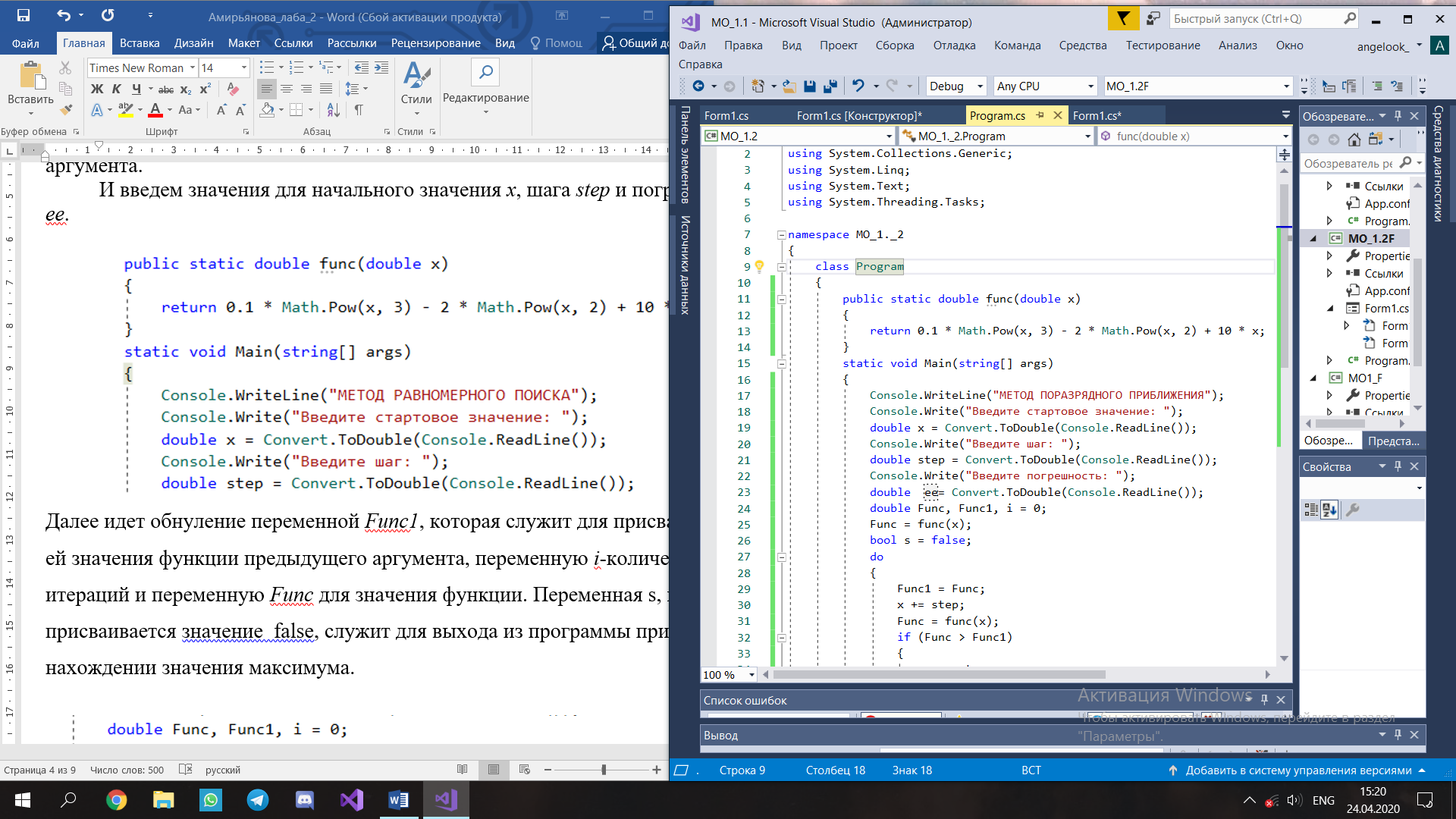


КОД ПРОГРАММЫ

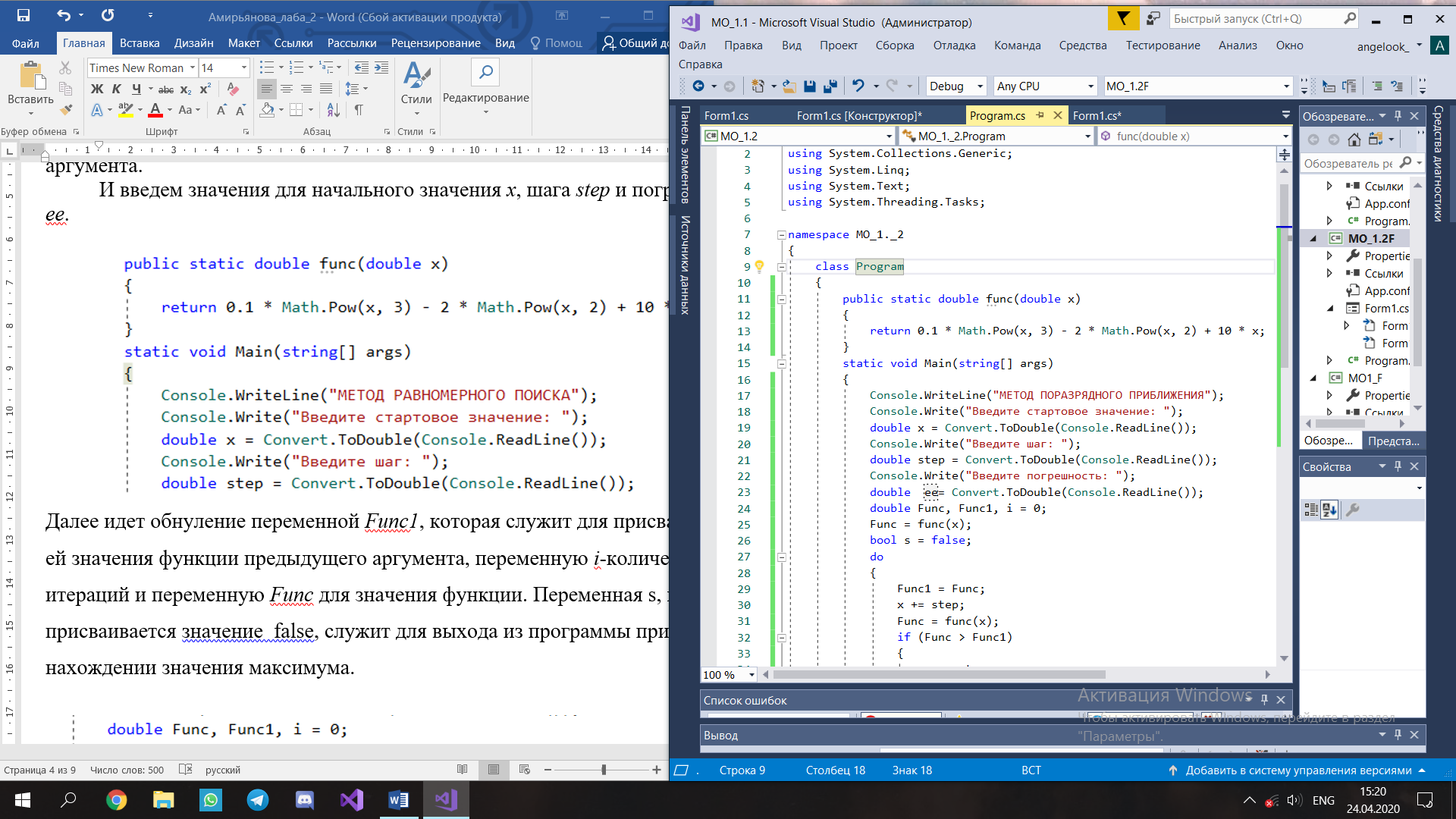
Программа на языке C# для реализации поиска максимума методом равномерного поиска.

Для удобной работы с функцией, для которой нужно найти максимум, создадим метод *func*, возвращающий значение данной нам функции для аргумента.

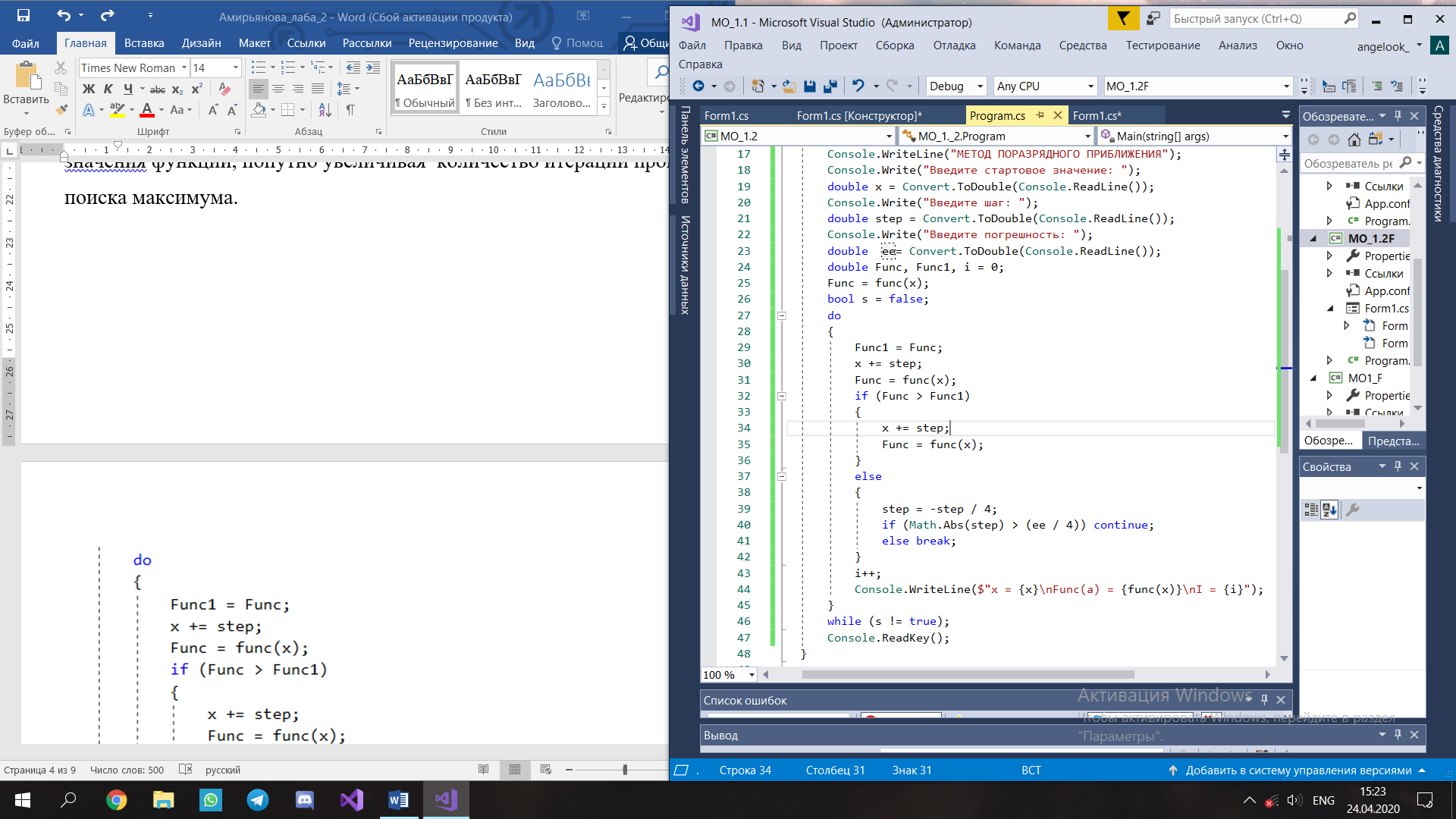
И введем значения для начального значения *x*, шага *step* и погрешности *ee*.



Далее идет обнуление переменной *Func1*, которая служит для присваивания ей значения функции предыдущего аргумента, переменную *i*-количество итераций и переменную *Func* для значения функции. Переменная s, которой присваивается значение false, служит для выхода из программы при нахождении значения максимума.



Далее начинается цикл с постусловием, в котором присваиваем *Func1* значение функции от *x*, увеличиваем *x* на шаг и сравниваем *Func* (*x*n+1) и *Func(xn)*. Если *Func(xn+1)* больше, то продолжаем увеличивать *x* и сравнивать значения функций, попутно увеличивая количество итераций процесса поиска максимума.



Если F(xn+1) станет меньше F(xn), то шаг поиска уменьшаем в 4 раза.

Затем проверяем условие |step| > (ee/4), если оно выполняется, увеличиваем на 1 счетчик количества итераций. Далее поиск максимума в другом направлении с шагом в 4 раза меньше.

Когда s не true, поиск останавливается и выводится ответ, который содержит точку экстремума, её значение и количество итераций.

РЕЗУЛЬТАТЫ ПРОГРАММЫ

Было создано приложение Windows Forms для визуализации программы. На рисунке 1 представлена форма для реализации метода равномерного поиска.

Здесь имеются элементы TextBox для записи в них начального значения и шага. После ввода данных при нажатии кнопки «ВЫЧИСЛИТЬ» в окошке результатов появятся значения:

« х = » - здесь записывается значения экстремума (точки, в которой функция принимает максимальное значение);

« F(x) = » - здесь записывается максимальное значение функции;

« i = » - здесь отображается количество итераций, за которое программа нашла решение.

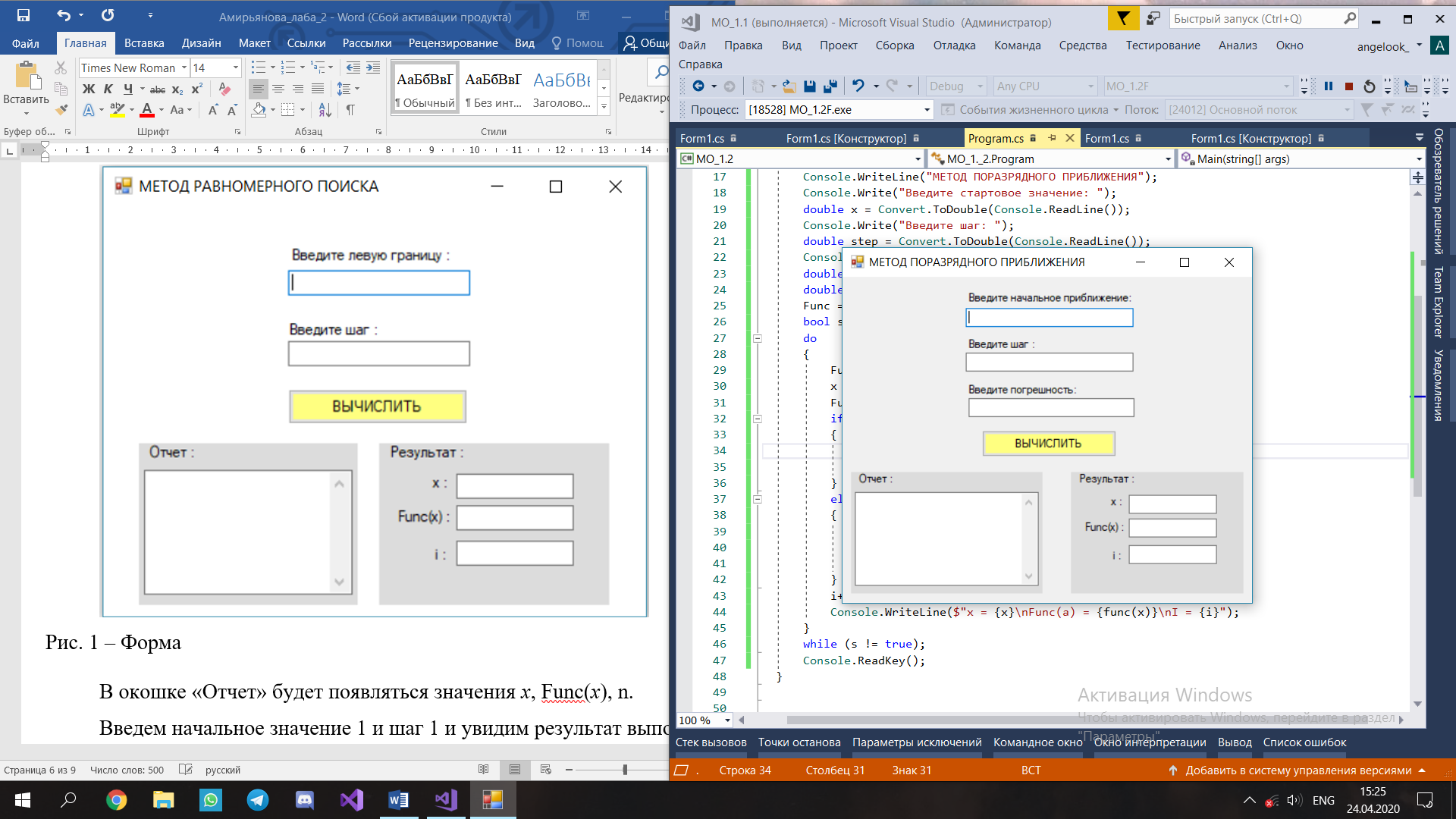


Рис. 1 – Форма

В окошке «Отчет» будет появляться значения *x*, Func(*x*), n.

Введем начальное значение 1, шаг 1, погрешность 0.0001 и увидим результат выполнения программы на рисунке 2.   
За 30 итерации был найден максимум равный 14,814814808192 в *х* =3,333251953125.

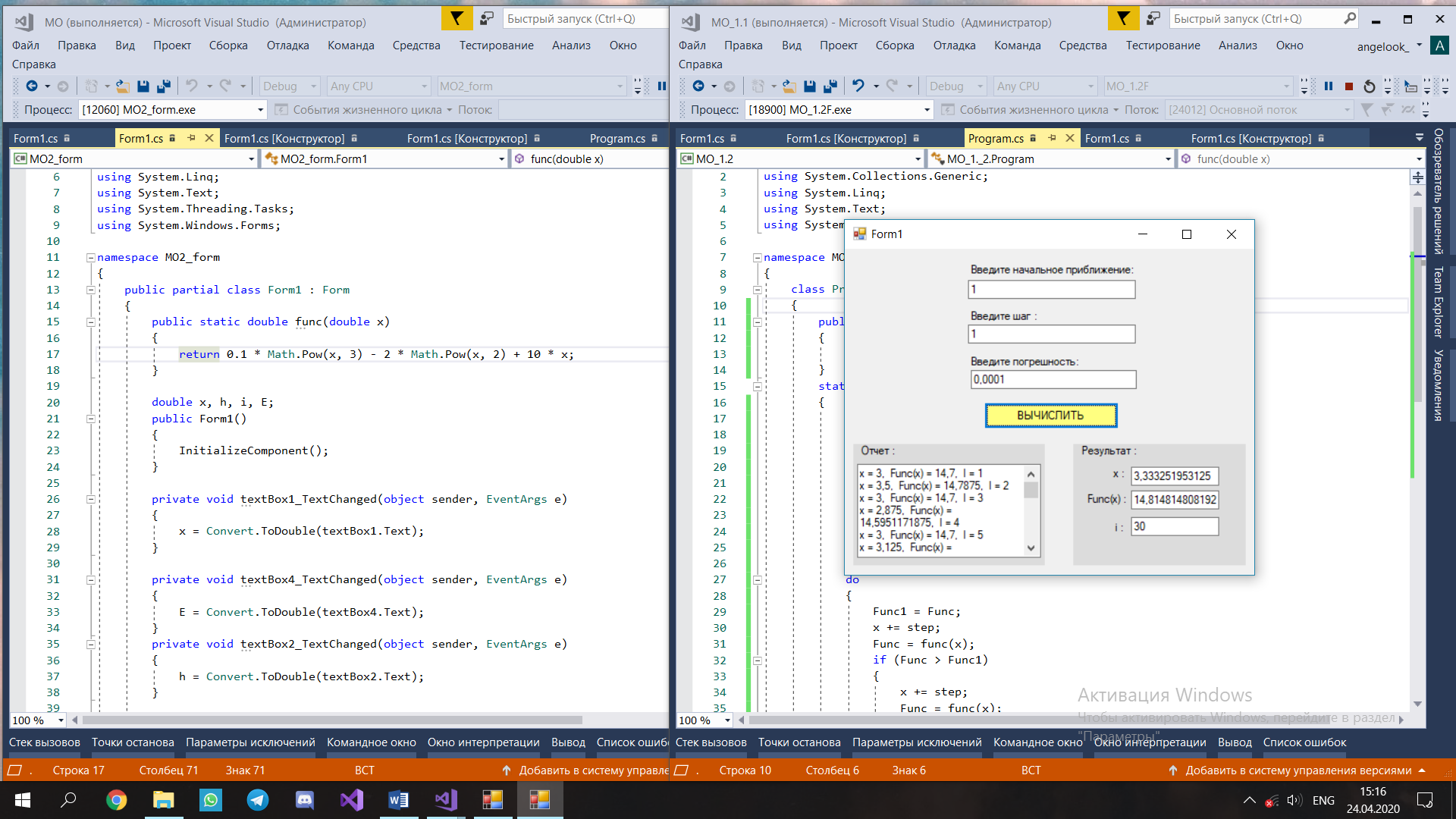


Рис. 2 – Результаты тестирования

А далее можно уменьшим шаг до 0,01. Начальное значение оставим равным 1.

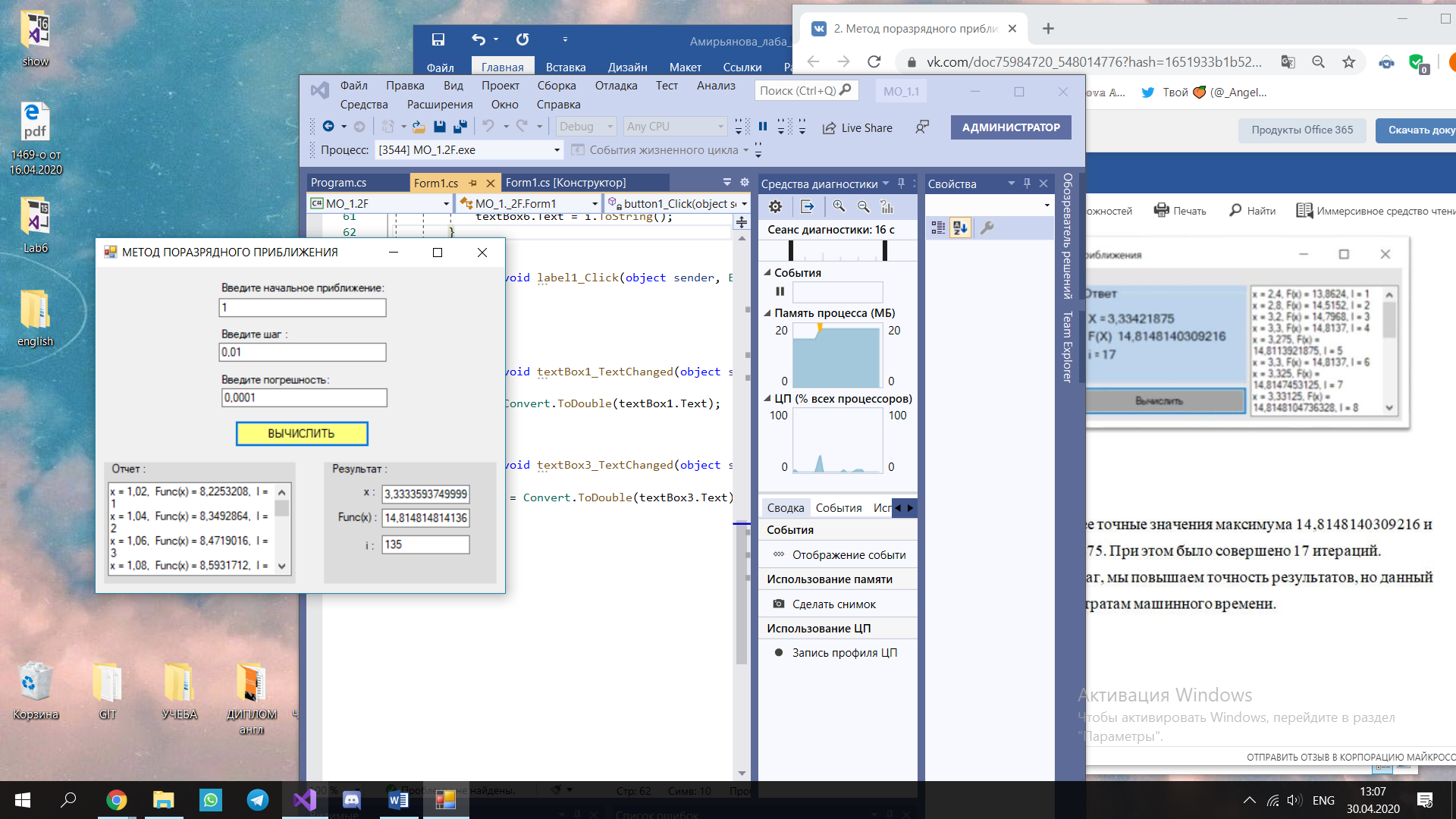


Рис. 3 - Результаты второго тестирования

Получили более точные значения максимума функции. Максимальным значением оказалось 14,814814814136 , а точкой максимума 3,3333593749999. Было совершено 135 итераций.

Можно сделать итог, что уменьшая шаг, мы увеличиваем точность результата. Но данный метод запрашивает много машинного времени.

СПИСОК ЛИТЕРАТУРЫ

1. Бродецкий, Г.Л. Экономико-математические методы и модели в логистике: процедуры оптимизации: Учеб. для студентов учреждений высшего профессионального образования / Г.Л. Бродецкий. - М.: ИЦ Академия, 2012.
2. Аттетков, А.В. Методы оптимизации: Учебное пособие / А.В. Аттетков, В.С. Зарубин, А.Н. Канатников. - М.: ИЦ РИОР, НИЦ Инфра-М, 2013.
3. Васильев, Ф.П. Методы оптимизации в 2-х книгах. Кн.2 / Ф.П. Васильев. - М.: МЦНМО, 2011.