Шифрование данных

Понятие Потока

- **Поток** — абстрактное представление последовательного устройства, облегчающее последовательное хранение данных и доступ к ним (по одному байту в каждый конкретный момент времени).

Потоки делятся на две категории:

- *потоки опорных хранилищ* - потоки, жестко привязанные к конкретным типам опорных хранилищ, такие
  как FileStream или NetworkStream
- *потоки-**декораторы*** - наполняют другие потоки, трансформируя данные тем или иным способом, такие
  как DeflateStream или CryptoStream

потоки-декораторы

- **Декораторы** освобождают потоки опорных хранилищ от необходимости самостоятельно реализовывать такие вещи, как сжатие и шифрование. **Декораторы** можно подключать во время выполнения, а также соединять их в цепочки (т.е. использовать несколько декораторов в одном потоке).

Шифрование

- Симметричное и асимметричное шифрования выполняются с использованием различных процессов. Симметричное шифрование выполняется в рамках потоков, поэтому его удобно применять для шифрования больших объемов данных.
- Асимметричное шифрование выполняется в рамках небольшого числа байтов, поэтому его удобно применять для шифрования только небольших объемов данных.

Класс CryptoStream

- Управляемые классы симметричного шифрования используются со специальным классом потока CryptoStream , который шифрует данные, считанные в поток. Класс **CryptoStream** инициализируется при помощи управляемого класса потока, класса, реализующего интерфейс ICryptoTransform (созданный из класса, который реализует алгоритм шифрования), и перечисления CryptoStreamMode , описывающего разрешенный тип доступа для **CryptoStream**.
- Класс **CryptoStream** может быть инициализирован при помощи любого класса, производного от класса Stream , включая FileStream, MemoryStream и NetworkStream. При помощи этих классов можно осуществлять симметричное шифрование для различных объектов потока.

Реализация алгоритмов симметричного шифрования в платформе .NET Framework

В качестве примера различных реализаций, доступных для алгоритма, рассмотрим симметричные алгоритмы. Основой для всех симметричных алгоритмов является SymmetricAlgorithm, абстрактный класс, который наследуется следующими алгоритмами:

- Aes
- DES
- RC2
- Rijndael
- TripleDES

Класс Rijndael

Представляет базовый класс, от которого наследуются все реализации алгоритма симметричного шифрования Rijndael.

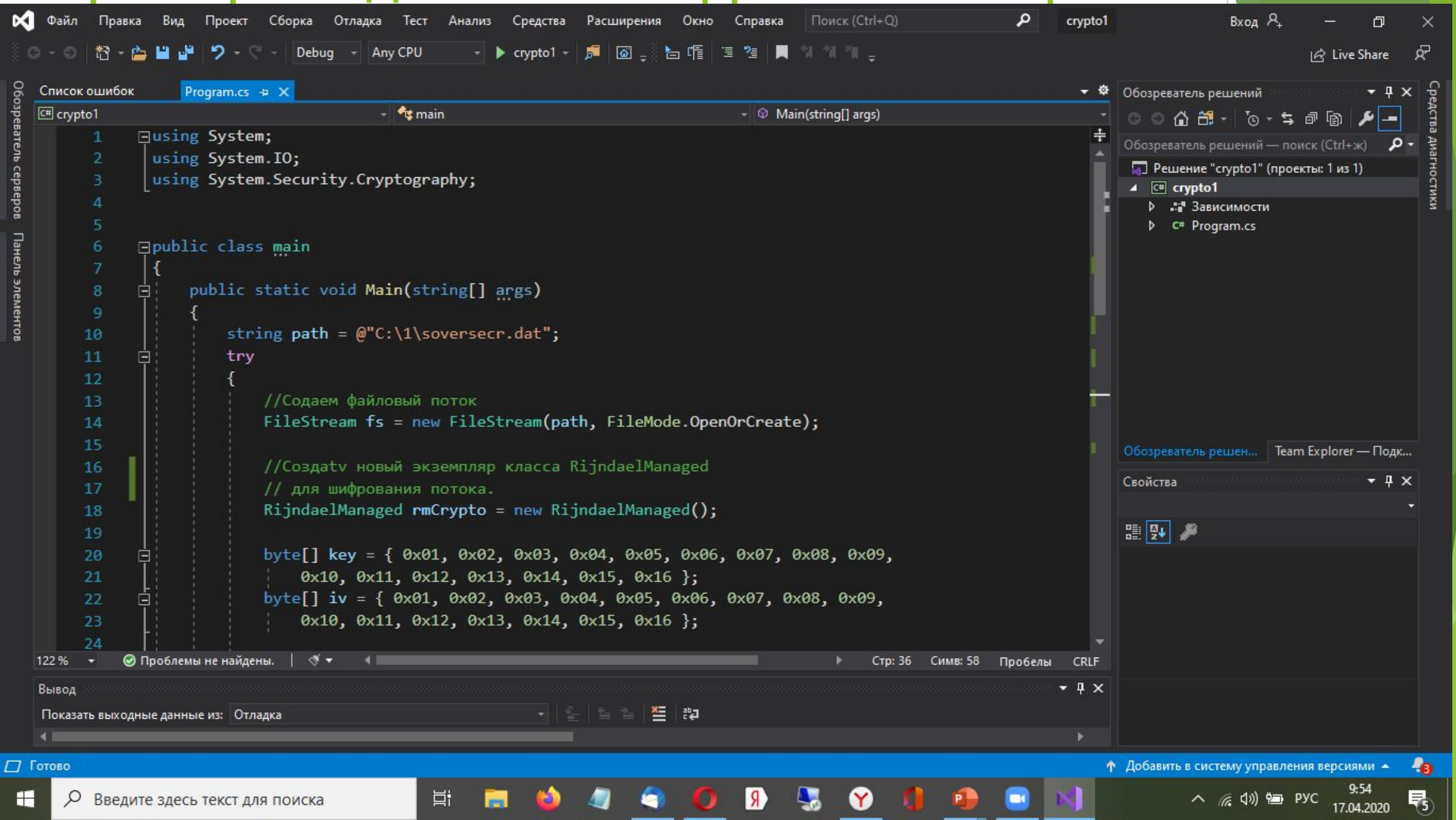
МЕТОДЫ:

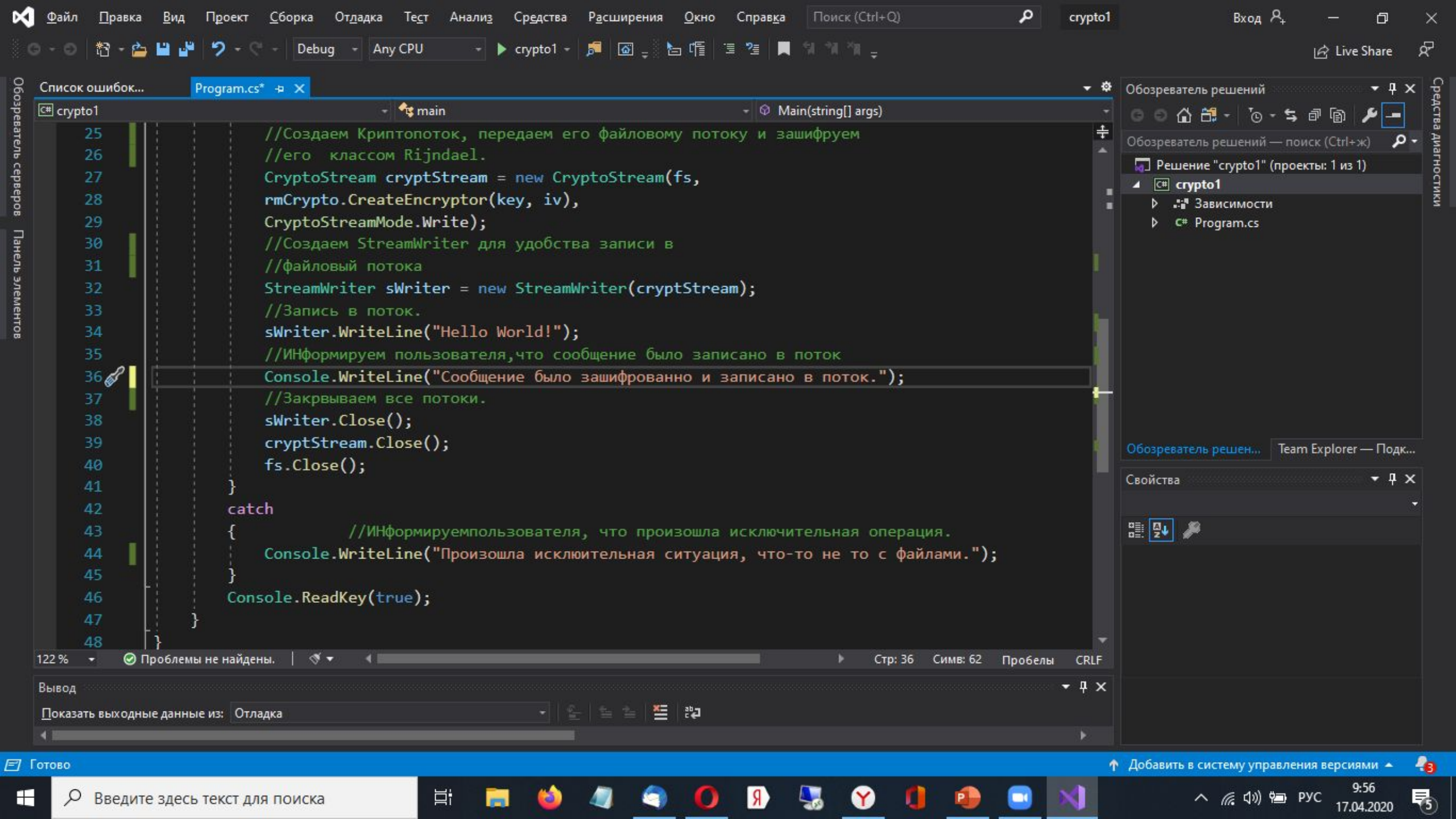
Clear()	Освобождает все ресурсы, используемые классом SymmetricAlgorithm. (Унаследовано от SymmetricAlgorithm)
Create()	Создает криптографический объект для выполнения алгоритма Rijndael.
Create(String)	Создает криптографический объект для выполнения заданной реализации алгоритма Rijndael.
CreateDecryptor()	Создает симметричный объект-дешифратор с текущим свойством Key и вектором инициализации (IV).
CreateDecryptor(Byte[], Byte[])	При переопределении в производном классе создает симметричный объект-дешифратор с указанным свойством Key и вектором инициализации (IV).
CreateEncryptor()	Создает симметричный объект-шифратор с текущим свойством Key и вектором инициализации (IV).
CreateEncryptor(Byte[], Byte[])	Если переопределено в производном классе, создает симметричный объект-шифратор с заданным свойством Key и вектором инициализации (IV).
Dispose()	Освобождает ресурсы, используемые текущим экземпляром класса SymmetricAlgorithm.

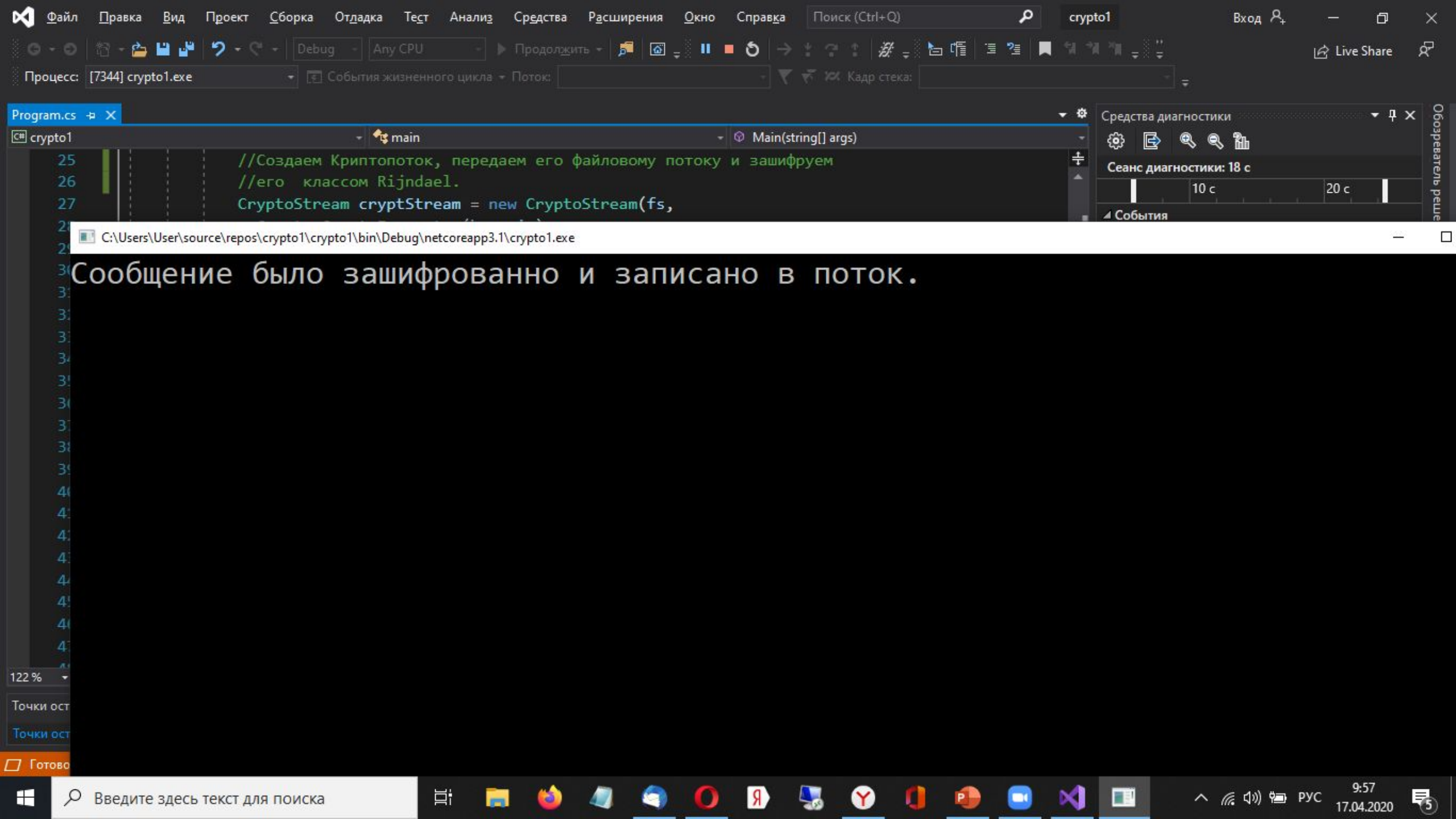
Класс Rijndael

Dispose(Boolean)	Освобождает неуправляемые ресурсы, используемые SymmetricAlgorithm, и дополнительно освобождает управляемые ресурсы.
Equals(Object)	Определяет, равен ли указанный объект текущему объекту.
GenerateIV()	Если переопределено в производном классе, создает произвольный вектор инициализации (IV), используемый для алгоритма.
GenerateKey()	Если переопределено в производном классе, генерирует произвольный ключ (Key), используемый для алгоритма.
GetHashCode()	Служит в качестве хэш-функции по умолчанию.
GetType()	Возвращает объект Type для текущего экземпляра.
MemberwiseClone()	Создает неполную копию текущего объекта Object.
ToString()	Возвращает строку, представляющую текущий объект.
ValidKeySize(Int32)	Определяет допустимость указанного размера ключа для текущего алгоритма.

Пример создания зашифрованного файла




Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
crypto1
Вход
Debug
Any CPU
crypto1
Live Share
Список ошибок...
Program.cs*
crypto1
main
Main(string[] args)
//Создаем Криптопоток, передаем его файловому потоку и зашифруем
//его классом Rijndael.
CryptoStream cryptStream = new CryptoStream(fs,
rmCrypto.CreateEncryptor(key, iv),
CryptoStreamMode.Write);
//Создаем StreamWriter для удобства записи в
//файловый потока
StreamWriter sWriter = new StreamWriter(cryptStream);
//Запись в поток.
sWriter.WriteLine("Hello World!");
//ИНформируем пользователя,что сообщение было записано в поток
Console.WriteLine("Сообщение было зашифрованно и записано в поток.");
//Закрвываем все потоки.
sWriter.Close();
cryptStream.Close();
fs.Close();
}
catch
{ //ИНформируемпользователя, что произошла исключительная операция.
Console.WriteLine("Произошла исключительная ситуация, что-то не то с файлами.");
}
Console.ReadKey(true);
}
}
122 %
Проблемы не найдены.
Стр: 36
Симв: 62
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "crypto1" (проекты: 1 из 1)
crypto1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Средства диагностики
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
9:56
17.04.2020


Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
crypto1
Вход
Debug
Any CPU
Продолжить
Live Share
Процесс: [7344] crypto1.exe
События жизненного цикла
Поток:
Кадр стека:
Program.cs
crypto1
main
Main(string[] args)
//Создаем Криптопоток, передаем его файловому потоку и зашифруем
//его классом Rijndael.
CryptoStream cryptStream = new CryptoStream(fs,
Средства диагностики
Сеанс диагностики: 18 с
10 с
20 с
События
Обозреватель реше
C:\Users\User\source\repos\crypto1\crypto1\bin\Debug\netcoreapp3.1\crypto1.exe
Сообщение было зашифрованно и записано в поток.
122 %
Точки ост
Точки ост
Готово
Введите здесь текст для поиска
РУС
9:57
17.04.2020

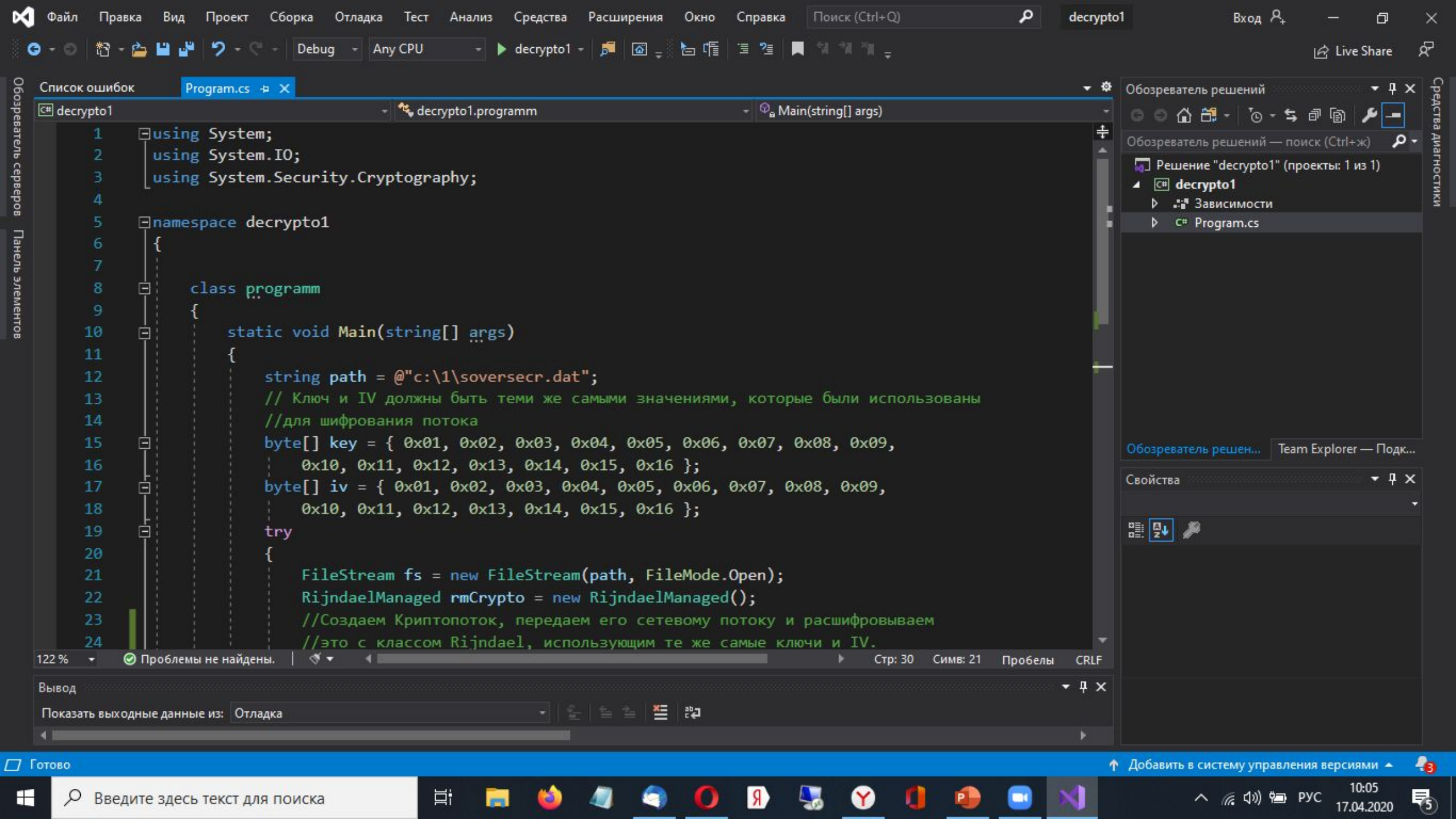
Расшифровка данных

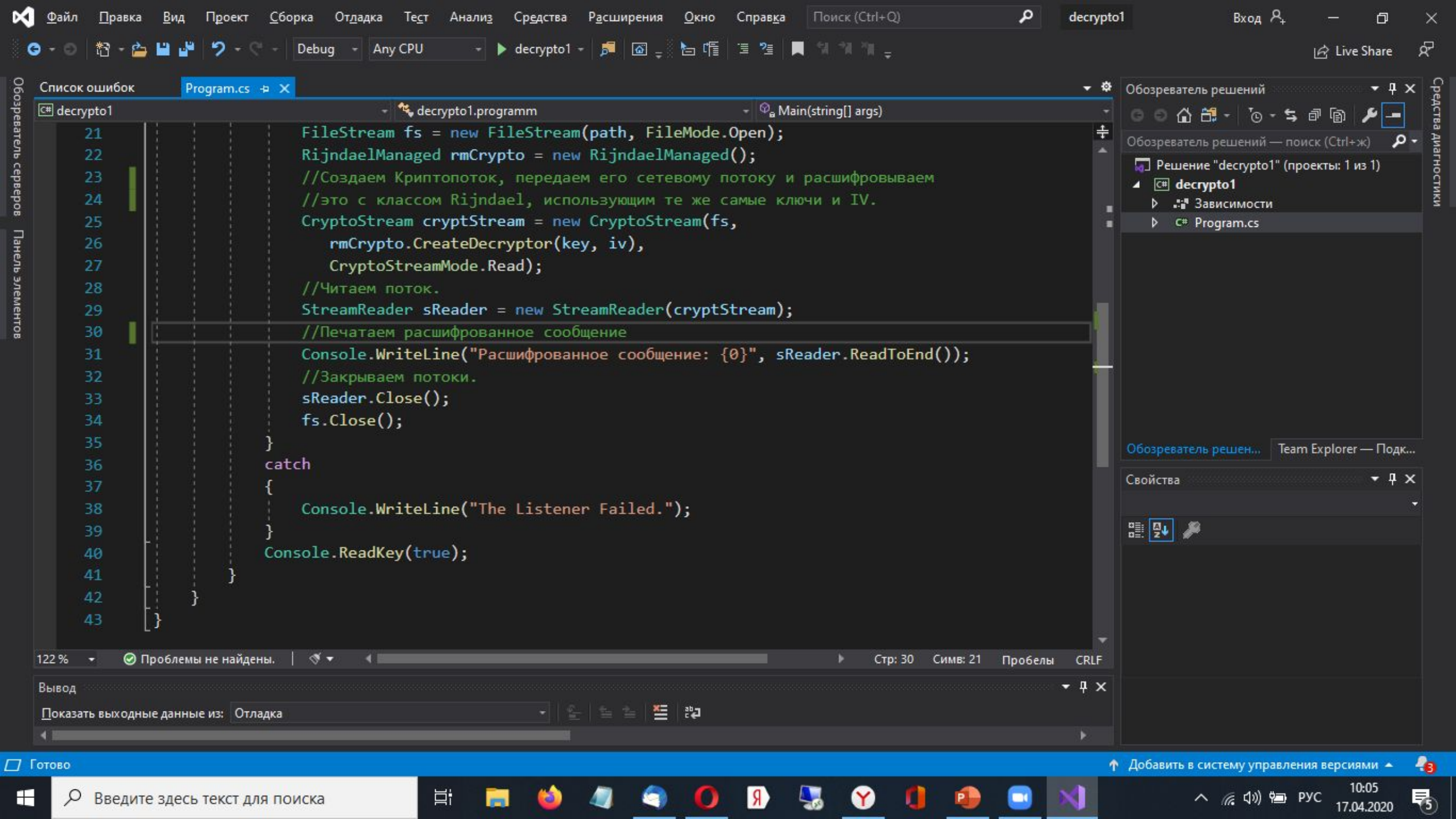
Расшифровка представляет собой операцию, обратную операции шифрования. Для шифрования с секретным ключом необходимо знать как ключ, так и вектор инициализации, которые использовались при шифровании данных. Для шифрования с открытым ключом необходимо знать либо открытый ключ (если данные были зашифрованы при помощи закрытого ключа), либо закрытый ключ (если данные были зашифрованы при помощи открытого ключа).

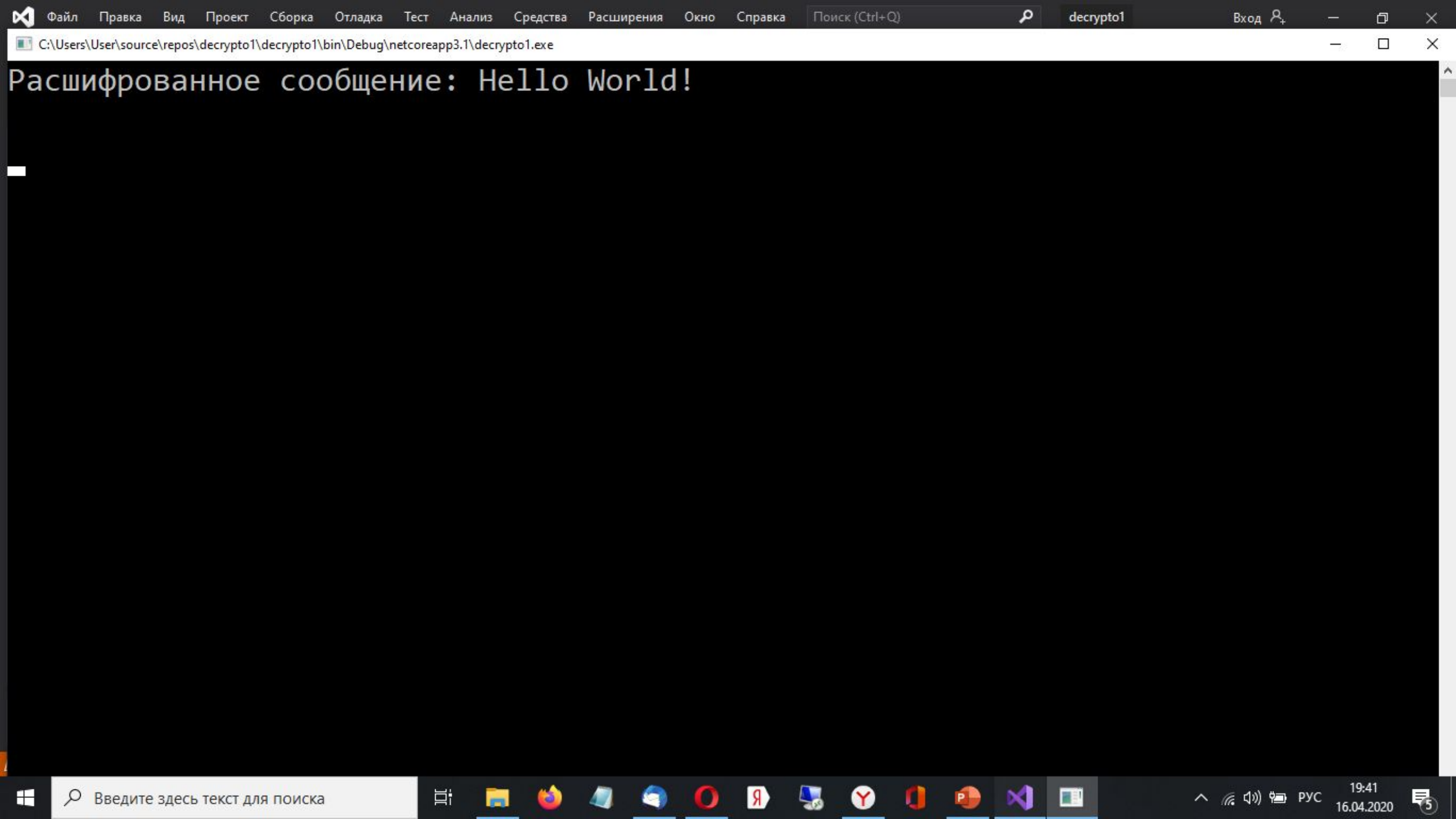
Симметричная расшифровка

Расшифровка данных, зашифрованных при помощи симметричных алгоритмов, похожа на процесс, используемый для шифрования данных при помощи симметричных алгоритмов.

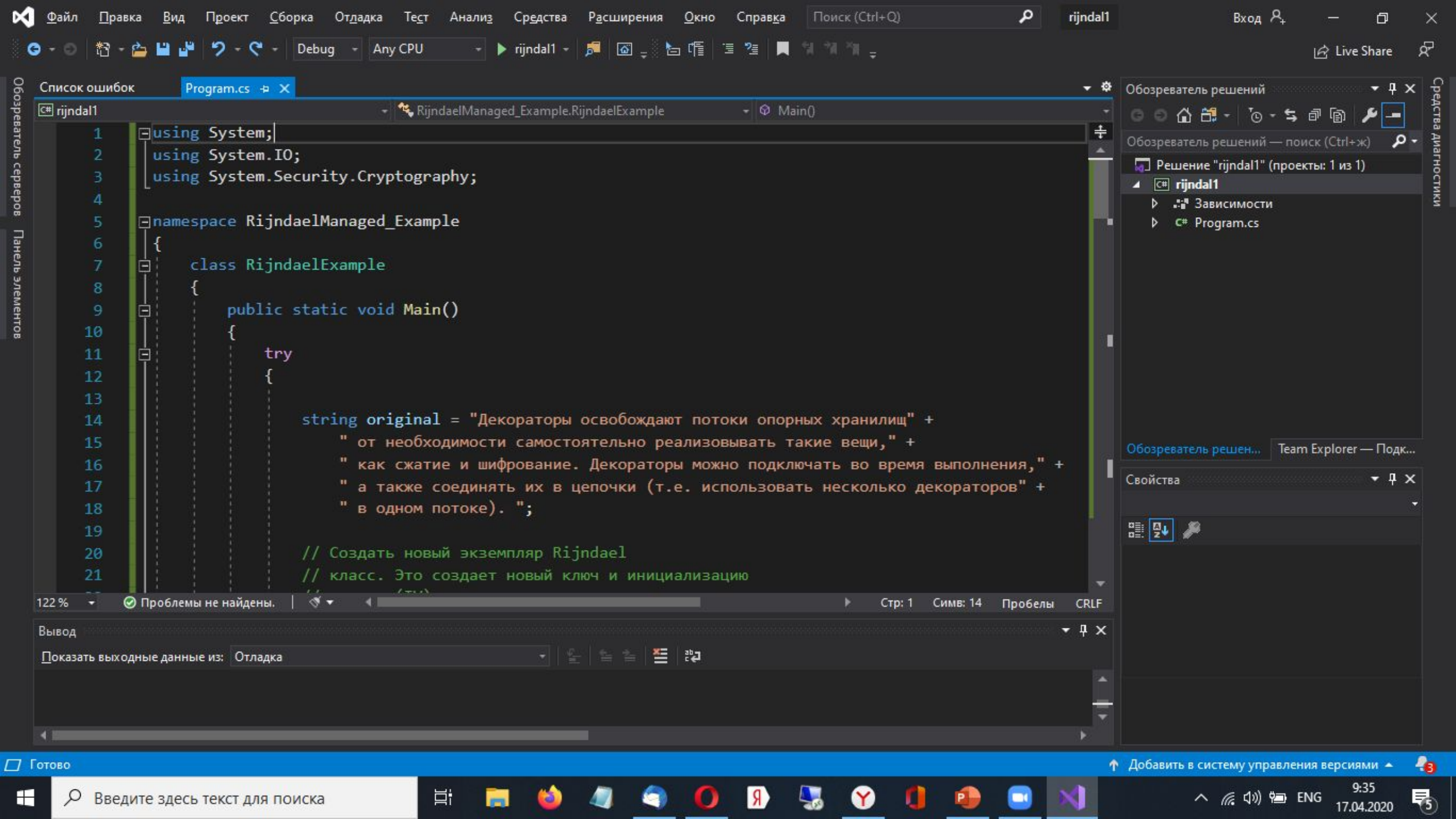
Класс CryptoStream используется с классами симметричного шифрования, предоставляемыми платформой .NET Framework, для расшифровки данных, считанных из любого управляемого объекта потока.

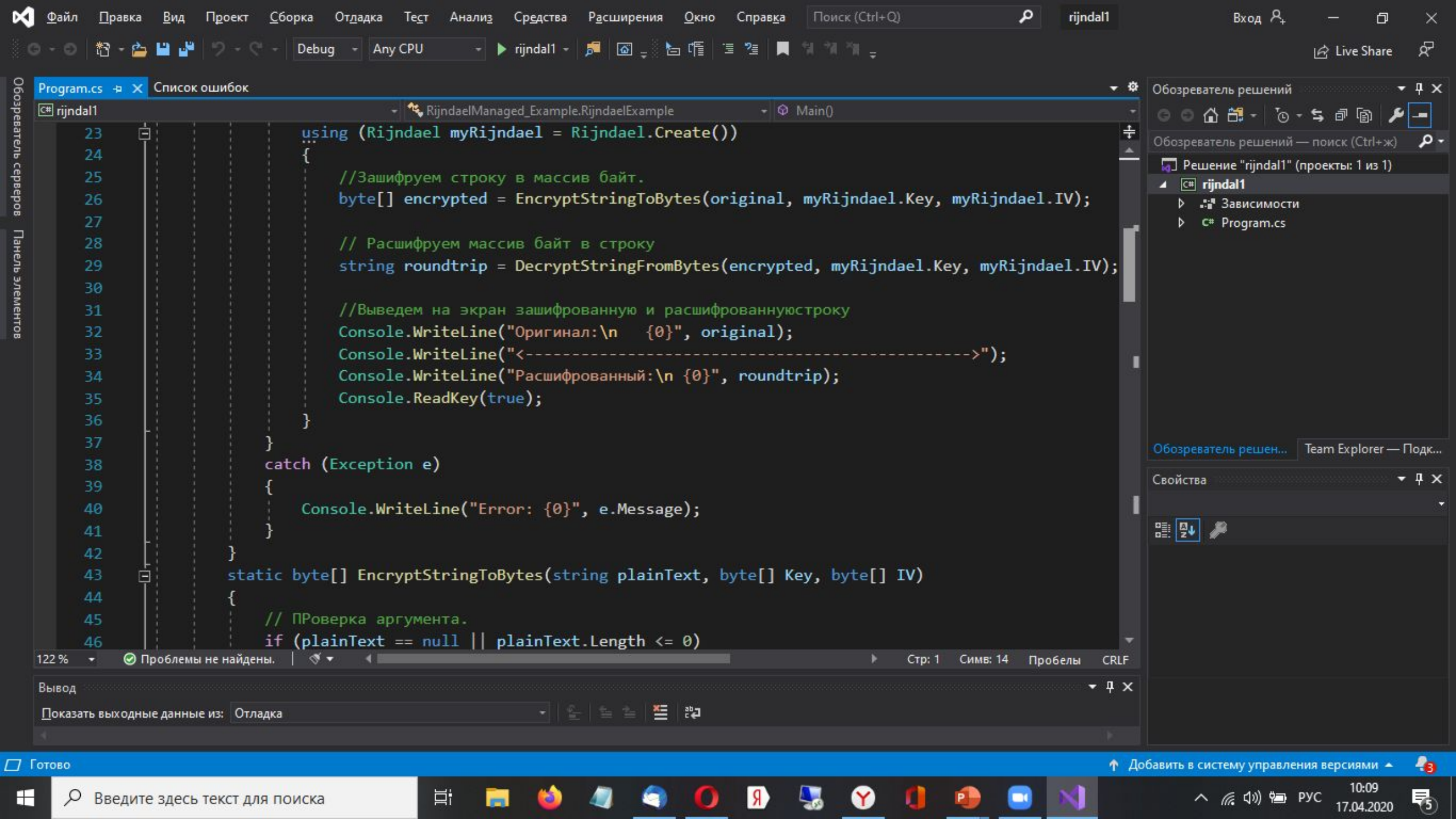

Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
decrypto1
Вход
Debug
Any CPU
decrypto1
Live Share
Список ошибок
Program.cs
decrypto1
decrypto1.programm
Main(string[] args)
using System;
using System.IO;
using System.Security.Cryptography;
namespace decrypto1
{
class programm
{
static void Main(string[] args)
{
string path = @"c:\1\soversecr.dat";
// Ключ и IV должны быть теми же самыми значениями, которые были использованы
//для шифрования потока
byte[] key = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09,
0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16 };
byte[] iv = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09,
0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16 };
try
{
FileStream fs = new FileStream(path, FileMode.Open);
RijndaelManaged rmCrypto = new RijndaelManaged();
//Создаем Криптопоток, передаем его сетевому потоку и расшифровываем
//это с классом Rijndael, использующим те же самые ключи и IV.
122 %
Проблемы не найдены.
Стр: 30
Симв: 21
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "decrypto1" (проекты: 1 из 1)
decrypto1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Средства диагностики
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:05
17.04.2020

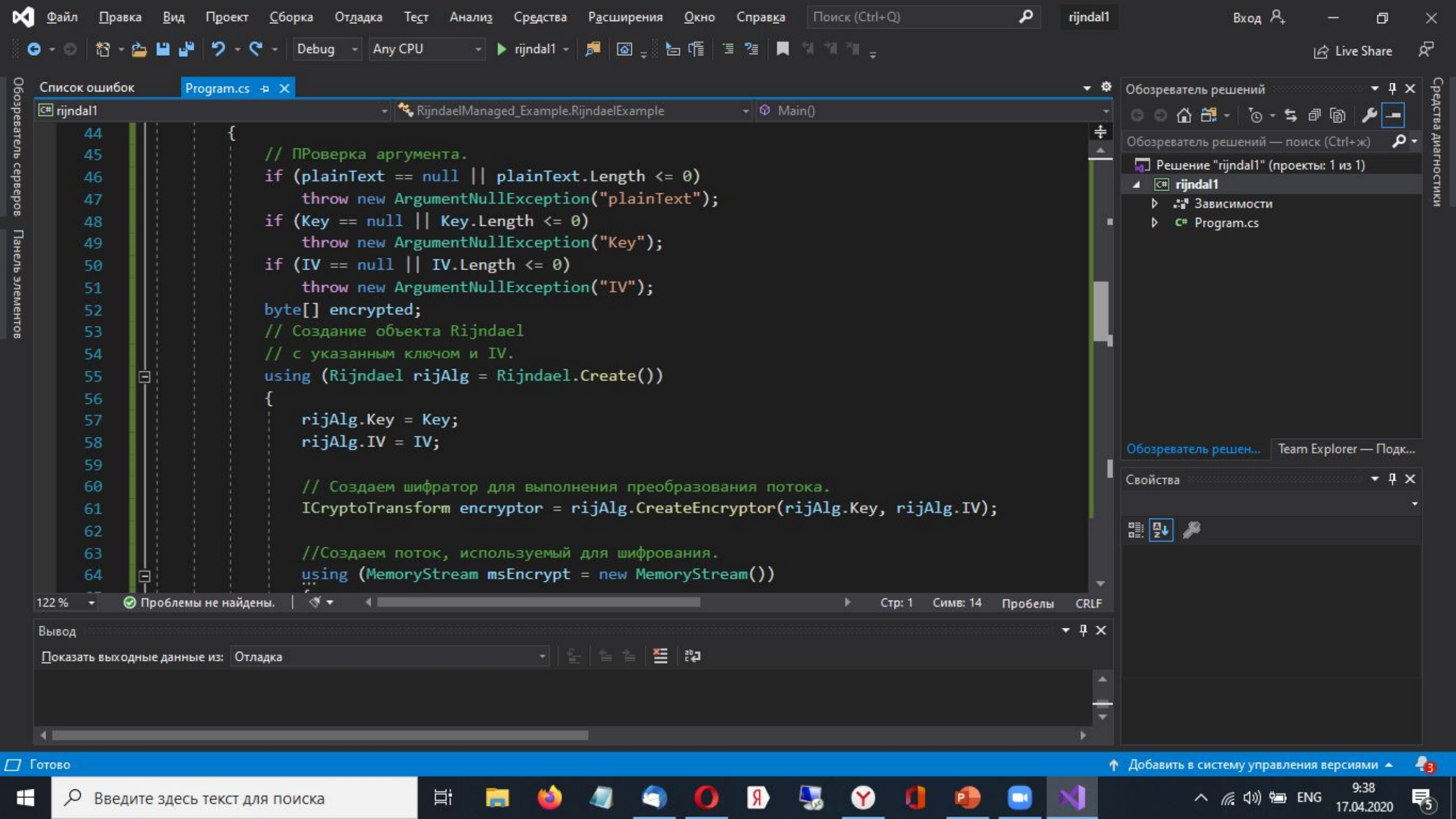

Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
decrypto1
Вход
Debug
Any CPU
decrypto1
Live Share
Список ошибок
Program.cs
decrypto1
decrypto1.programm
Main(string[] args)
FileStream fs = new FileStream(path, FileMode.Open);
RijndaelManaged rmCrypto = new RijndaelManaged();
//Создаем Криптопоток, передаем его сетевому потоку и расшифровываем
//это с классом Rijndael, использующим те же самые ключи и IV.
CryptoStream cryptStream = new CryptoStream(fs,
rmCrypto.CreateDecryptor(key, iv),
CryptoStreamMode.Read);
//Читаем поток.
StreamReader sReader = new StreamReader(cryptStream);
//Печатаем расшифрованное сообщение
Console.WriteLine("Расшифрованное сообщение: {0}", sReader.ReadToEnd());
//Закрываем потоки.
sReader.Close();
fs.Close();
}
catch
{
Console.WriteLine("The Listener Failed.");
}
Console.ReadKey(true);
}
}
}
122 %
Проблемы не найдены.
Стр: 30
Симв: 21
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "decrypto1" (проекты: 1 из 1)
decrypto1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Средства диагностики
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:05
17.04.2020

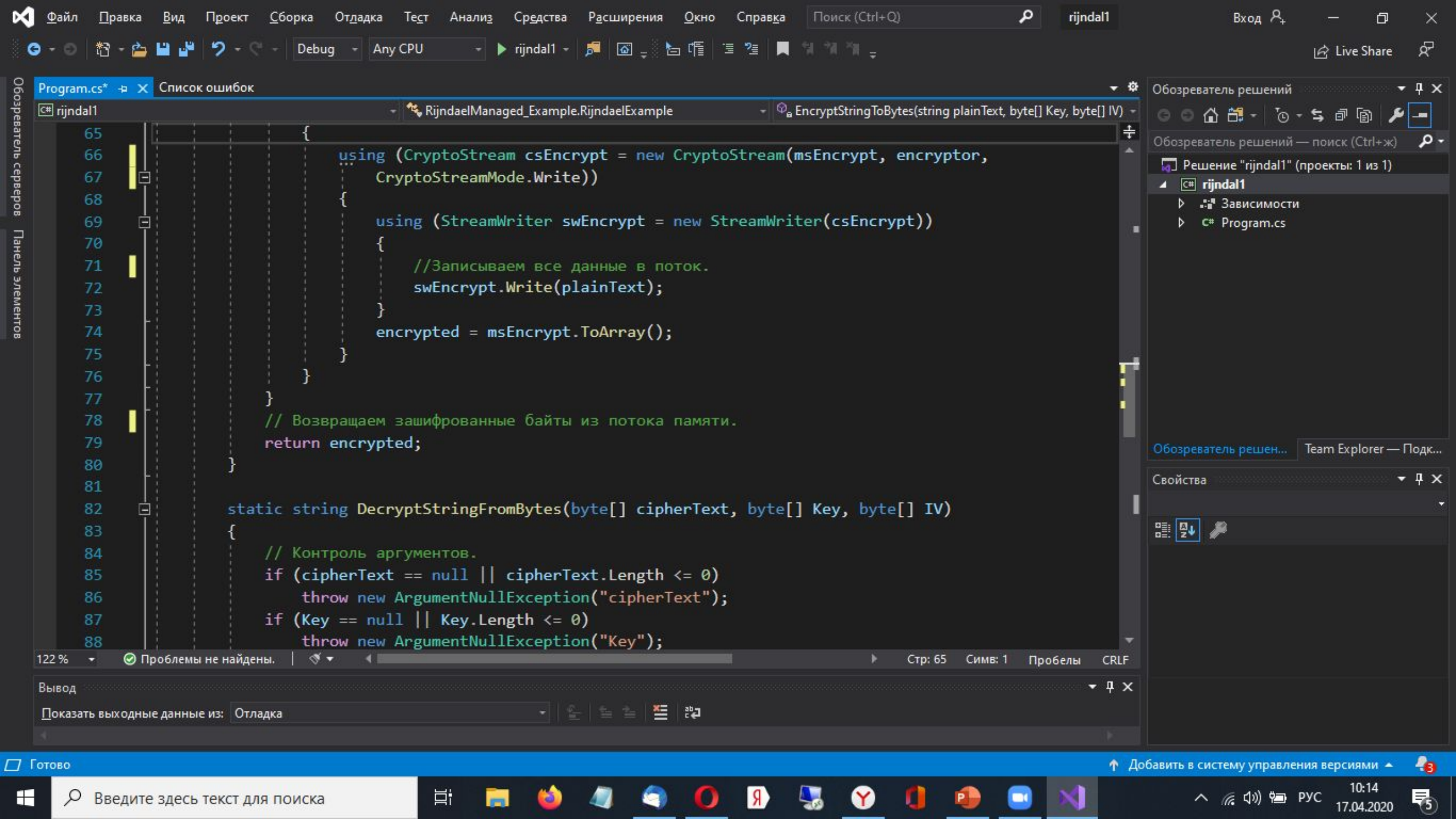

Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
decrypto1
Вход
C:\Users\User\source\repos\decrypto1\decrypto1\bin\Debug\netcoreapp3.1\decrypto1.exe
Расшифрованное сообщение: Hello World!
Введите здесь текст для поиска
РУС
19:41
16.04.2020

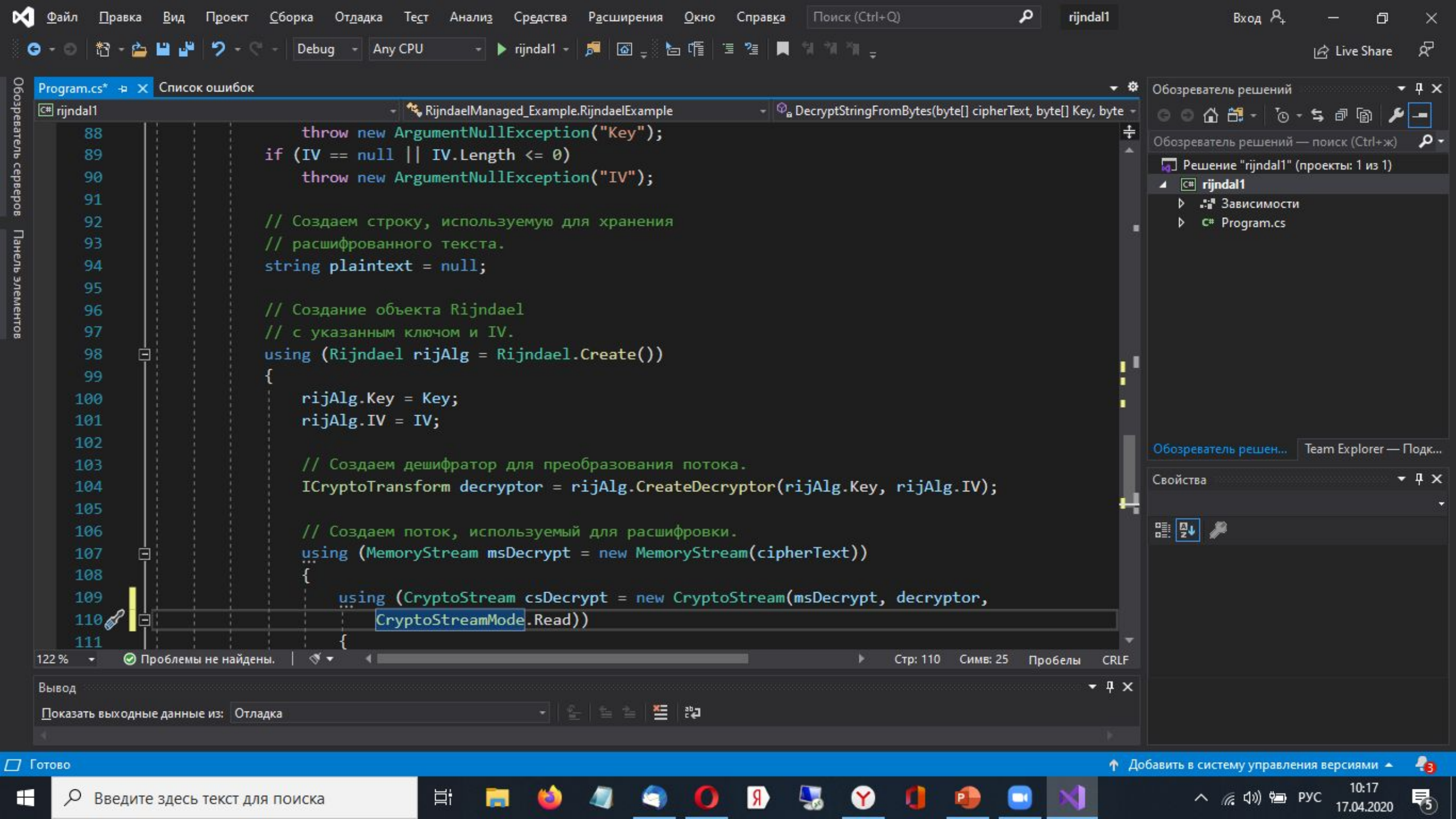
Пример единой программы реализующий метод шифрования Rijndail

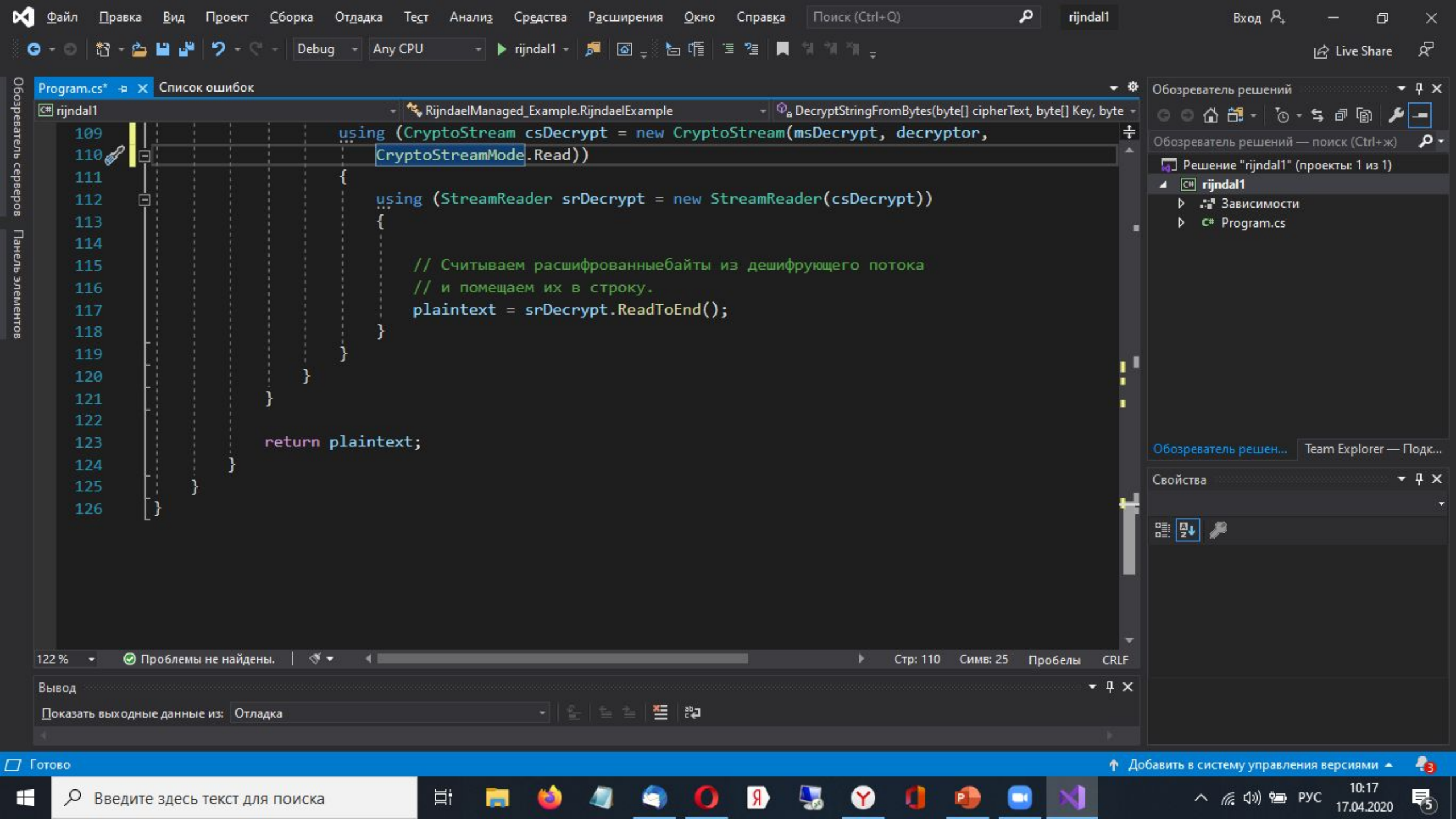

Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
rijndal1
Вход
Debug
Any CPU
rijndal1
Live Share
Список ошибок
Program.cs
rijndal1
RijndaelManaged_Example.RijndaelExample
Main()
using System;
using System.IO;
using System.Security.Cryptography;
namespace RijndaelManaged_Example
{
class RijndaelExample
{
public static void Main()
{
try
{
string original = "Декораторы освобождают потоки опорных хранилищ" +
" от необходимости самостоятельно реализовывать такие вещи," +
" как сжатие и шифрование. Декораторы можно подключать во время выполнения," +
" а также соединять их в цепочки (т.е. использовать несколько декораторов" +
" в одном потоке). ";
// Создать новый экземпляр Rijndael
// класс. Это создает новый ключ и инициализацию
122 %
Проблемы не найдены.
Стр: 1
Симв: 14
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "rijndal1" (проекты: 1 из 1)
rijndal1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Средства диагностики
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
ENG
9:35
17.04.2020


Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
rijndal1
Вход
Debug
Any CPU
rijndal1
Live Share
Program.cs
Список ошибок
rijndal1
RijndaelManaged_Example.RijndaelExample
Main()
using (Rijndael myRijndael = Rijndael.Create())
{
//Зашифруем строку в массив байт.
byte[] encrypted = EncryptStringToBytes(original, myRijndael.Key, myRijndael.IV);
// Расшифруем массив байт в строку
string roundtrip = DecryptStringFromBytes(encrypted, myRijndael.Key, myRijndael.IV);
//Выведем на экран зашифрованную и расшифрованнуюстроку
Console.WriteLine("Оригинал:\n {0}", original);
Console.WriteLine("<---------->");
Console.WriteLine("Расшифрованный:\n {0}", roundtrip);
Console.ReadKey(true);
}
}
catch (Exception e)
{
Console.WriteLine("Error: {0}", e.Message);
}
}
static byte[] EncryptStringToBytes(string plainText, byte[] Key, byte[] IV)
{
// ПРоверка аргумента.
if (plainText == null || plainText.Length <= 0)
122 %
Проблемы не найдены.
Стр: 1
Симв: 14
Пробелы
CRLF
Вывод
Показать выходные данные из: Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "rijndal1" (проекты: 1 из 1)
rijndal1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:09
17.04.2020


Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
rijndal1
Вход
Debug
Any CPU
rijndal1
Live Share
Список ошибок
Program.cs
rijndal1
RijndaelManaged_Example.RijndaelExample
Main()
{
// ПРоверка аргумента.
if (plainText == null || plainText.Length <= 0)
throw new ArgumentNullException("plainText");
if (Key == null || Key.Length <= 0)
throw new ArgumentNullException("Key");
if (IV == null || IV.Length <= 0)
throw new ArgumentNullException("IV");
byte[] encrypted;
// Создание объекта Rijndael
// с указанным ключом и IV.
using (Rijndael rijAlg = Rijndael.Create())
{
rijAlg.Key = Key;
rijAlg.IV = IV;
// Создаем шифратор для выполнения преобразования потока.
ICryptoTransform encryptor = rijAlg.CreateEncryptor(rijAlg.Key, rijAlg.IV);
//Создаем поток, используемый для шифрования.
using (MemoryStream msEncrypt = new MemoryStream())
122 %
Проблемы не найдены.
Стр: 1
Симв: 14
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "rijndal1" (проекты: 1 из 1)
rijndal1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Средства диагностики
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
ENG
9:38
17.04.2020


Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
rijndal1
Вход
Debug
Any CPU
rijndal1
Live Share
Program.cs*
Список ошибок
rijndal1
RijndaelManaged_Example.RijndaelExample
EncryptStringToBytes(string plainText, byte[] Key, byte[] IV)
{
using (CryptoStream csEncrypt = new CryptoStream(msEncrypt, encryptor,
CryptoStreamMode.Write))
{
using (StreamWriter swEncrypt = new StreamWriter(csEncrypt))
{
//Записываем все данные в поток.
swEncrypt.Write(plainText);
}
encrypted = msEncrypt.ToArray();
}
}
}
// Возвращаем зашифрованные байты из потока памяти.
return encrypted;
}
static string DecryptStringFromBytes(byte[] cipherText, byte[] Key, byte[] IV)
{
// Контроль аргументов.
if (cipherText == null || cipherText.Length <= 0)
throw new ArgumentNullException("cipherText");
if (Key == null || Key.Length <= 0)
throw new ArgumentNullException("Key");
122 %
Проблемы не найдены.
Стр: 65
Симв: 1
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "rijndal1" (проекты: 1 из 1)
rijndal1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:14
17.04.2020


Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
rijndal1
Вход
Debug
Any CPU
rijndal1
Live Share
Program.cs*
Список ошибок
rijndal1
RijndaelManaged_Example.RijndaelExample
DecryptStringFromBytes(byte[] cipherText, byte[] Key, byte
throw new ArgumentNullException("Key");
if (IV == null || IV.Length <= 0)
throw new ArgumentNullException("IV");
// Создаем строку, используемую для хранения
// расшифрованного текста.
string plaintext = null;
// Создание объекта Rijndael
// с указанным ключом и IV.
using (Rijndael rijAlg = Rijndael.Create())
{
rijAlg.Key = Key;
rijAlg.IV = IV;
// Создаем дешифратор для преобразования потока.
ICryptoTransform decryptor = rijAlg.CreateDecryptor(rijAlg.Key, rijAlg.IV);
// Создаем поток, используемый для расшифровки.
using (MemoryStream msDecrypt = new MemoryStream(cipherText))
{
using (CryptoStream csDecrypt = new CryptoStream(msDecrypt, decryptor,
CryptoStreamMode.Read))
{
122 %
Проблемы не найдены.
Стр: 110
Симв: 25
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "rijndal1" (проекты: 1 из 1)
rijndal1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:17
17.04.2020


Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
rijndal1
Вход
Debug
Any CPU
rijndal1
Live Share
Program.cs*
Список ошибок
rijndal1
RijndaelManaged_Example.RijndaelExample
DecryptStringFromBytes(byte[] cipherText, byte[] Key, byte
using (CryptoStream csDecrypt = new CryptoStream(msDecrypt, decryptor,
CryptoStreamMode.Read))
{
using (StreamReader srDecrypt = new StreamReader(csDecrypt))
{
// Считываем расшифрованныебайты из дешифрующего потока
// и помещаем их в строку.
plaintext = srDecrypt.ReadToEnd();
}
}
}
}
return plaintext;
}
}
}
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "rijndal1" (проекты: 1 из 1)
rijndal1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
122 %
Проблемы не найдены.
Стр: 110
Симв: 25
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель серверов
Панель элементов
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:17
17.04.2020

C:\Users\User\source\repos\rijndal1\rijndal1\bin\Debug\netcoreapp3.1\rijndal1.exe

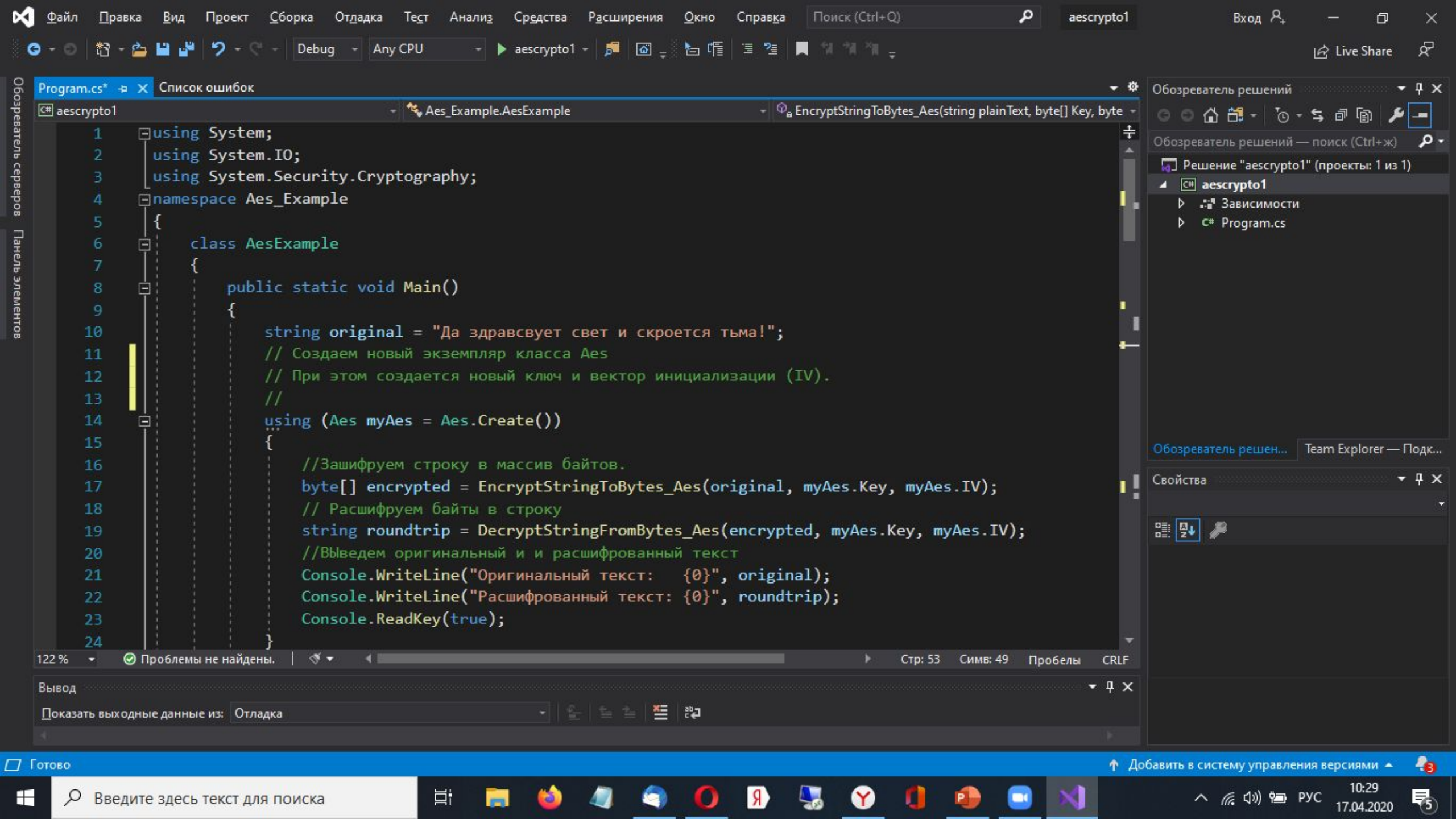
```
Оригинал:
   Декораторы освобождают потоки опорных хранилищ от необходимости самостоятель
но реализовывать такие вещи, как сжатие и шифрование. Декораторы можно подключа
ть во время выполнения, а также соединять их в цепочки (т.е. использовать неско
лько декораторов в одном потоке).
<---------------------------------------->
Расшифрованный:
 Декораторы освобождают потоки опорных хранилищ от необходимости самостоятельно
 реализовывать такие вещи, как сжатие и шифрование. Декораторы можно подключать
 во время выполнения, а также соединять их в цепочки (т.е. использовать несколь
ко декораторов в одном потоке).
```

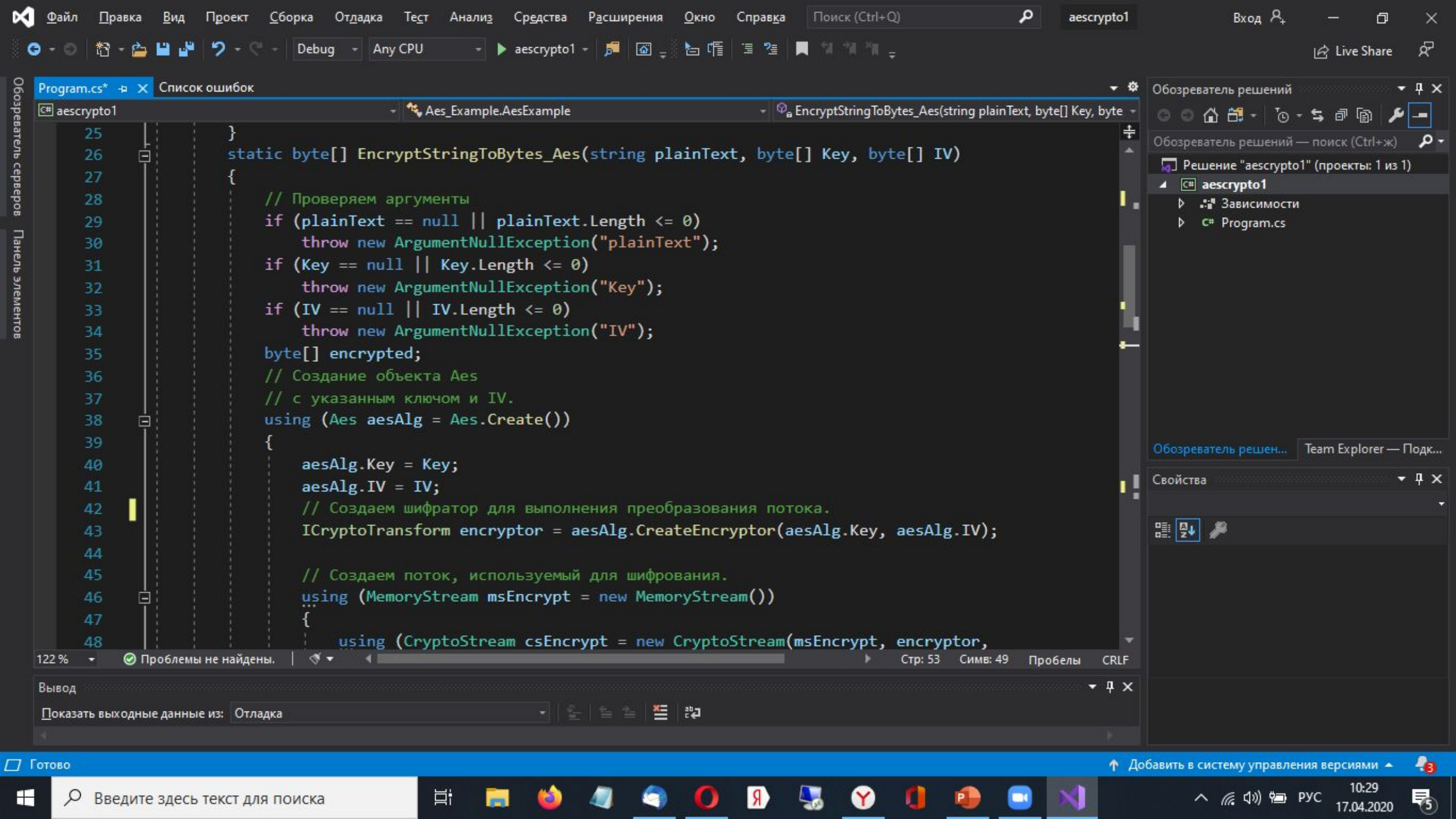
Методы класса AES

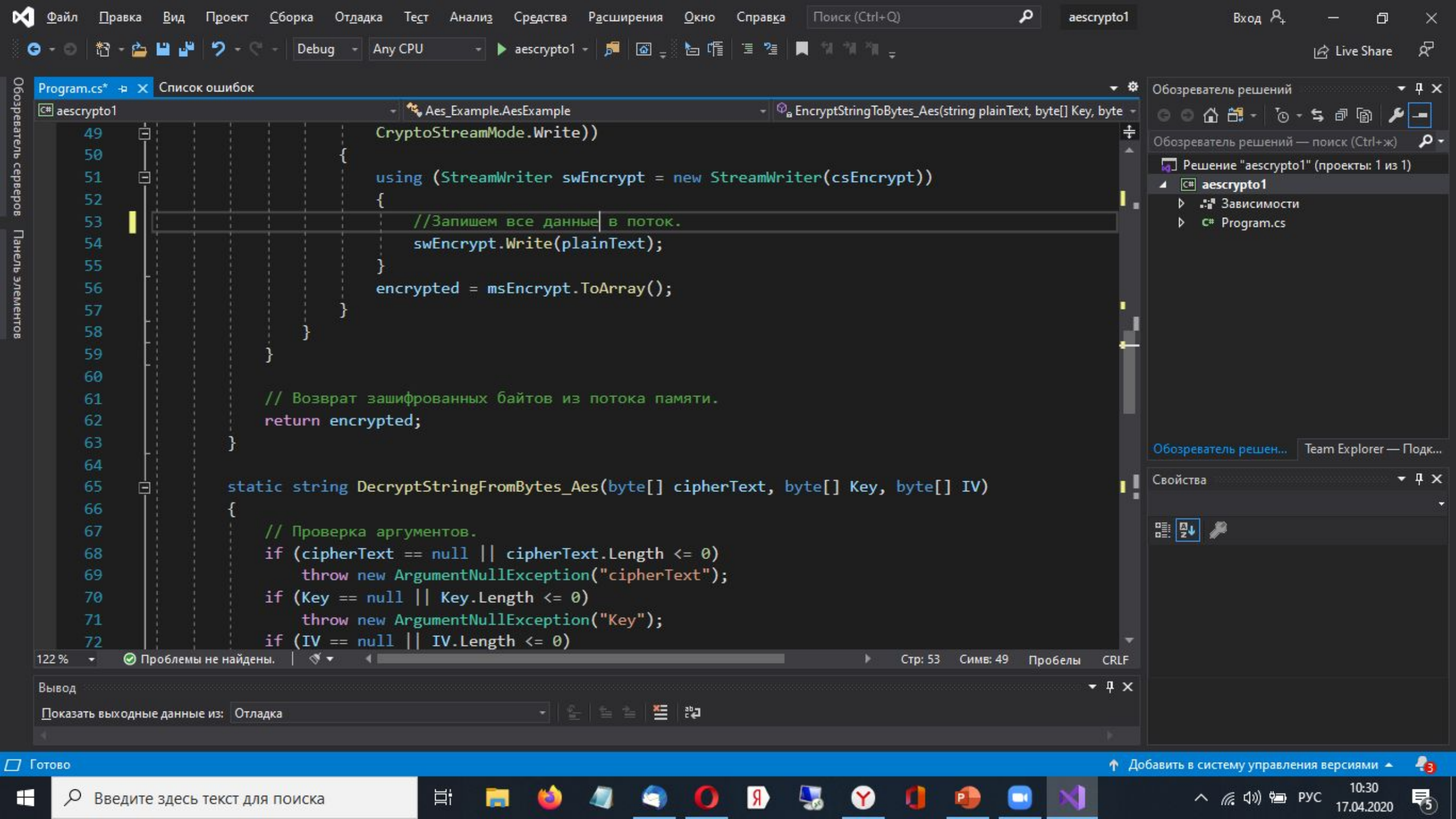
Clear()	Освобождает все ресурсы, используемые классом SymmetricAlgorithm. (Унаследовано от SymmetricAlgorithm)
Create()	Создает криптографический объект, используемый для выполнения симметричного алгоритма.
Create(String)	Создает криптографический объект, задающий реализацию AES для выполнения симметричного алгоритма.
CreateDecryptor()	Создает симметричный объект-дешифратор с текущим свойством Key и вектором инициализации (IV). (Унаследовано от SymmetricAlgorithm)
CreateDecryptor(Byte[], Byte[])	При переопределении в производном классе создает симметричный объект-дешифратор с указанным свойством Key и вектором инициализации (IV). (Унаследовано от SymmetricAlgorithm)
CreateEncryptor()	Создает симметричный объект-шифратор с текущим свойством Key и вектором инициализации (IV). (Унаследовано от SymmetricAlgorithm)
CreateEncryptor(Byte[], Byte[])	Если переопределено в производном классе, создает симметричный объект-шифратор с заданным свойством Key и вектором инициализации (IV). (Унаследовано от SymmetricAlgorithm)
Dispose()	Освобождает ресурсы, используемые текущим экземпляром

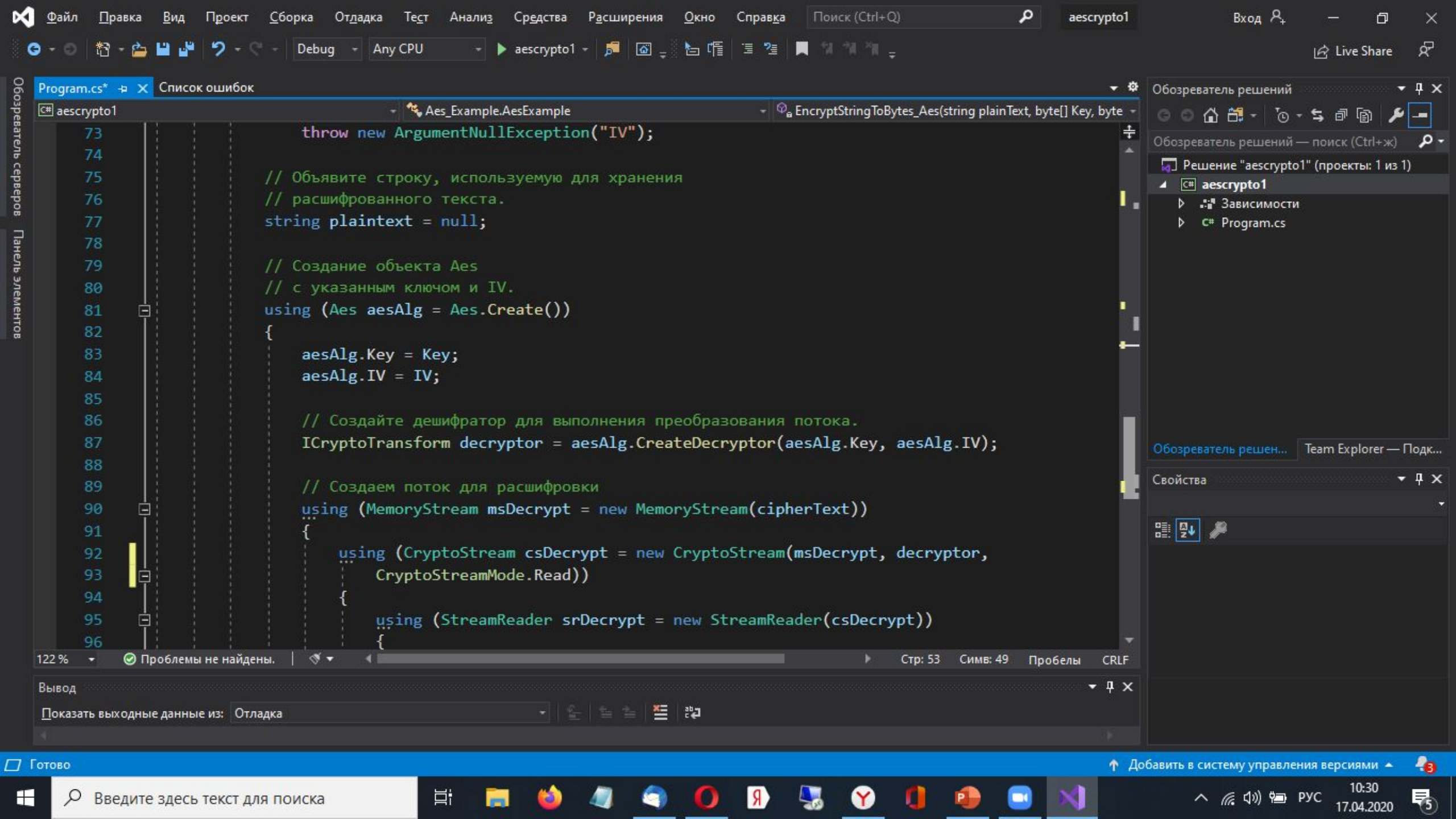
Методы класса AES

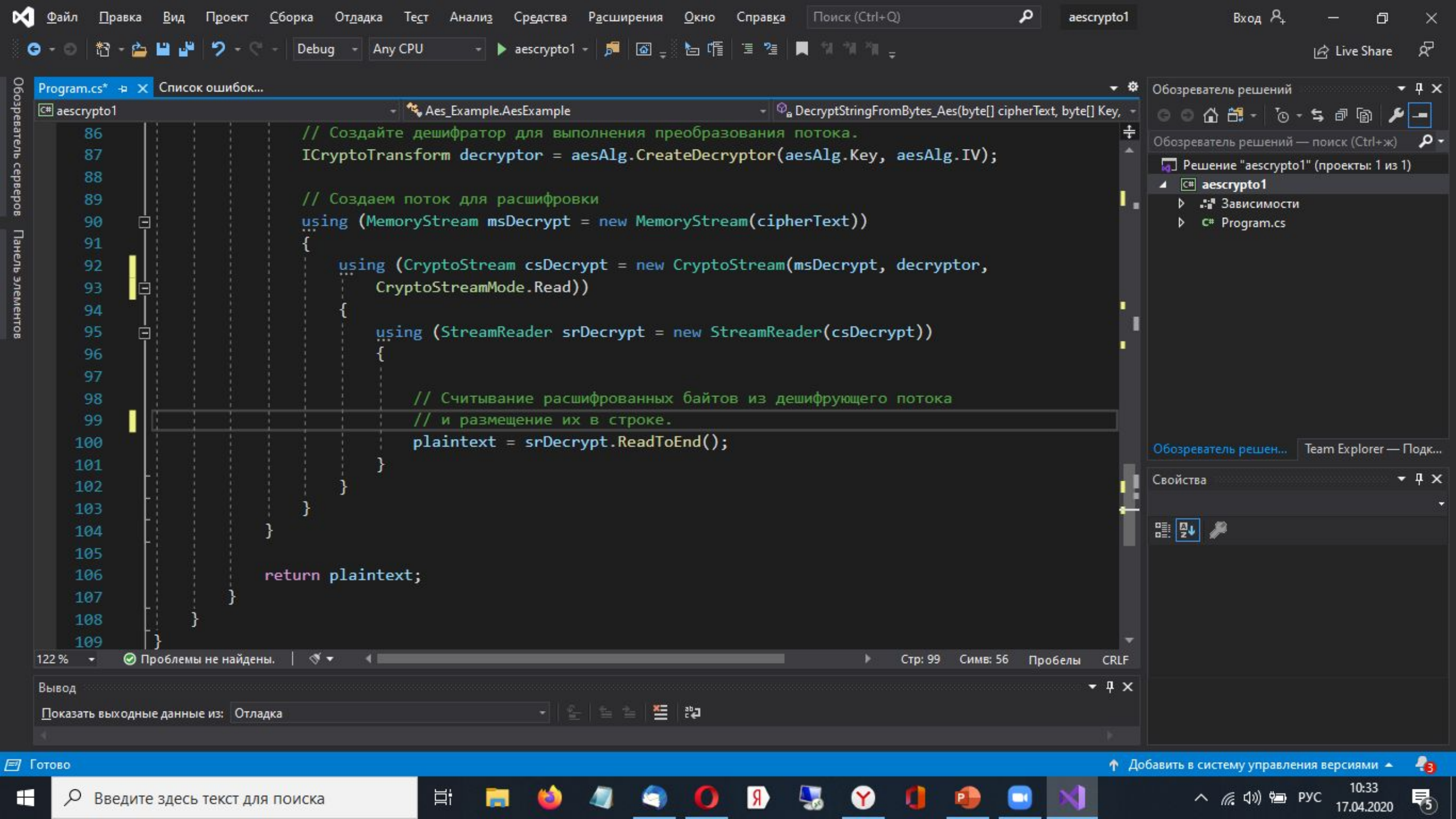
Dispose(Boolean)	Освобождает неуправляемые ресурсы, используемые SymmetricAlgorithm, и дополнительно освобождает управляемые ресурсы. (Унаследовано от SymmetricAlgorithm)
Equals(Object)	Определяет, равен ли указанный объект текущему объекту. (Унаследовано от Object)
GenerateIV()	Если переопределено в производном классе, создает произвольный вектор инициализации (IV), используемый для алгоритма. (Унаследовано от SymmetricAlgorithm)
GenerateKey()	Если переопределено в производном классе, генерирует произвольный ключ (Key), используемый для алгоритма. (Унаследовано от SymmetricAlgorithm)
GetHashCode()	Служит в качестве хэш-функции по умолчанию. (Унаследовано от Object)
GetType()	Возвращает объект Type для текущего экземпляра. (Унаследовано от Object)
MemberwiseClone()	Создает неполную копию текущего объекта Object. (Унаследовано от Object)
ToString()	Возвращает строку, представляющую текущий объект. (Унаследовано от Object)
ValidKeySize(Int32)	Определяет допустимость указанного размера ключа для текущего алгоритма. (Унаследовано от SymmetricAlgorithm)

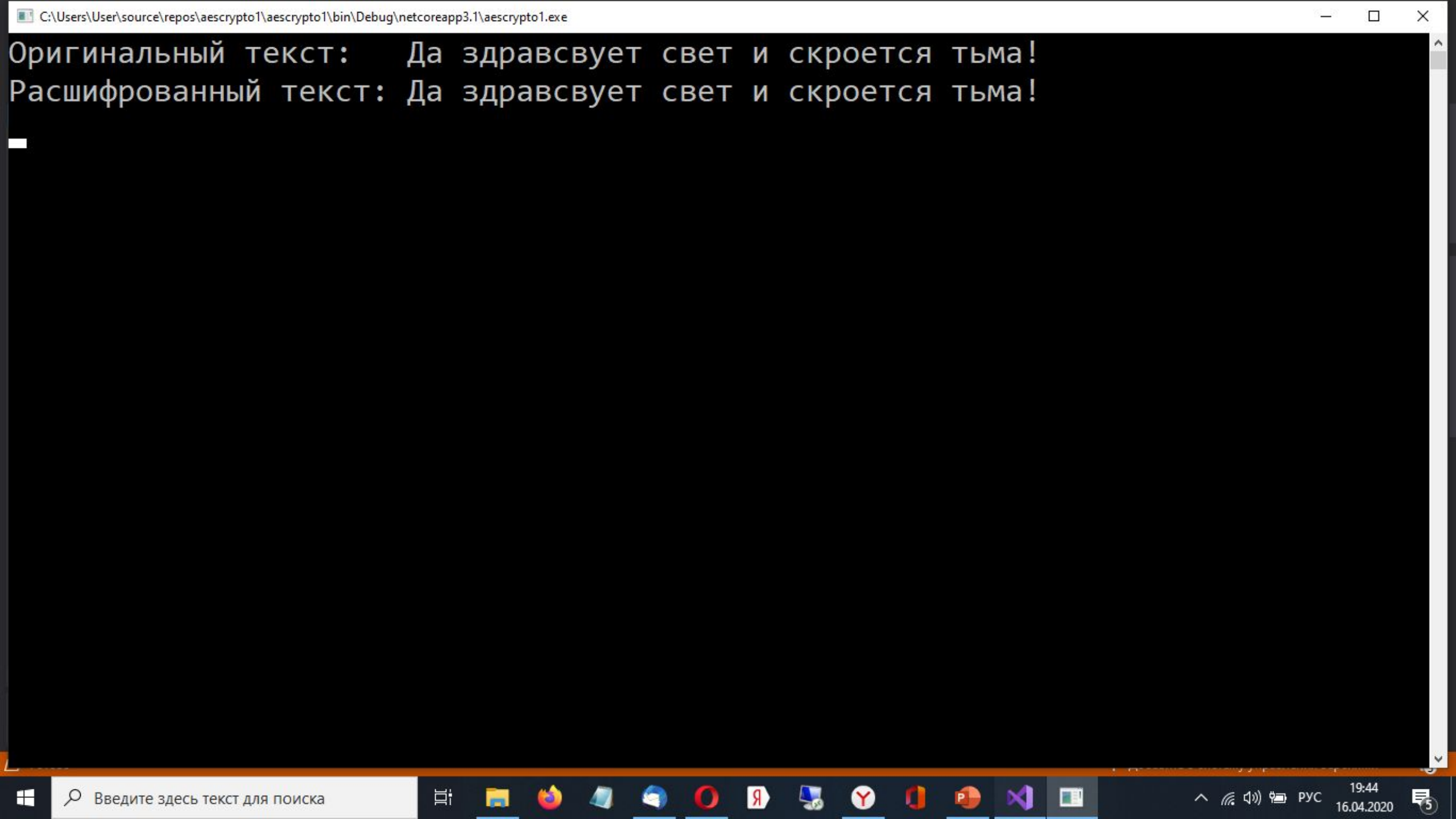

Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
aescrypto1
Вход
Debug
Any CPU
aescrypto1
Live Share
Program.cs*
Список ошибок
aescrypto1
Aes_Example.AesExample
EncryptStringToBytes_Aes(string plainText, byte[] Key, byte
using System;
using System.IO;
using System.Security.Cryptography;
namespace Aes_Example
{
class AesExample
{
public static void Main()
{
string original = "Да здравсвует свет и скроется тьма!";
// Создаем новый экземпляр класса Aes
// При этом создается новый ключ и вектор инициализации (IV).
//
using (Aes myAes = Aes.Create())
{
//Зашифруем строку в массив байтов.
byte[] encrypted = EncryptStringToBytes_Aes(original, myAes.Key, myAes.IV);
// Расшифруем байты в строку
string roundtrip = DecryptStringFromBytes_Aes(encrypted, myAes.Key, myAes.IV);
//Выведем оригинальный и и расшифрованный текст
Console.WriteLine("Оригинальный текст: {0}", original);
Console.WriteLine("Расшифрованный текст: {0}", roundtrip);
Console.ReadKey(true);
}
122 %
Проблемы не найдены.
Стр: 53
Симв: 49
Пробелы
CRLF
Вывод
Показать выходные данные из: Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "aescrypto1" (проекты: 1 из 1)
aescrypto1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:29
17.04.2020


Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
aescrypto1
Вход
Debug
Any CPU
aescrypto1
Live Share
Program.cs*
Список ошибок
aescrypto1
Aes_Example.AesExample
EncryptStringToBytes_Aes(string plainText, byte[] Key, byte
}
static byte[] EncryptStringToBytes_Aes(string plainText, byte[] Key, byte[] IV)
{
// Проверяем аргументы
if (plainText == null || plainText.Length <= 0)
throw new ArgumentNullException("plainText");
if (Key == null || Key.Length <= 0)
throw new ArgumentNullException("Key");
if (IV == null || IV.Length <= 0)
throw new ArgumentNullException("IV");
byte[] encrypted;
// Создание объекта Aes
// с указанным ключом и IV.
using (Aes aesAlg = Aes.Create())
{
aesAlg.Key = Key;
aesAlg.IV = IV;
// Создаем шифратор для выполнения преобразования потока.
ICryptoTransform encryptor = aesAlg.CreateEncryptor(aesAlg.Key, aesAlg.IV);
// Создаем поток, используемый для шифрования.
using (MemoryStream msEncrypt = new MemoryStream())
{
using (CryptoStream csEncrypt = new CryptoStream(msEncrypt, encryptor,
122 %
Проблемы не найдены.
Стр: 53
Симв: 49
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "aescrypto1" (проекты: 1 из 1)
aescrypto1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:29
17.04.2020


Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
aescrypto1
Вход
Debug
Any CPU
aescrypto1
Live Share
Program.cs*
Список ошибок
aescrypto1
Aes_Example.AesExample
EncryptStringToBytes_Aes(string plainText, byte[] Key, byte
CryptoStreamMode.Write))
{
using (StreamWriter swEncrypt = new StreamWriter(csEncrypt))
{
//Запишем все данные в поток.
swEncrypt.Write(plainText);
}
encrypted = msEncrypt.ToArray();
}
}
}
// Возврат зашифрованных байтов из потока памяти.
return encrypted;
}
static string DecryptStringFromBytes_Aes(byte[] cipherText, byte[] Key, byte[] IV)
{
// Проверка аргументов.
if (cipherText == null || cipherText.Length <= 0)
throw new ArgumentNullException("cipherText");
if (Key == null || Key.Length <= 0)
throw new ArgumentNullException("Key");
if (IV == null || IV.Length <= 0)
122 %
Проблемы не найдены.
Стр: 53
Симв: 49
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "aescrypto1" (проекты: 1 из 1)
aescrypto1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:30
17.04.2020


Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
aescrypto1
Вход
Debug
Any CPU
aescrypto1
Live Share
Program.cs*
Список ошибок
aescrypto1
Aes_Example.AesExample
EncryptStringToBytes_Aes(string plainText, byte[] Key, byte
73 throw new ArgumentNullException("IV");
74
75 // Объявите строку, используемую для хранения
76 // расшифрованного текста.
77 string plaintext = null;
78
79 // Создание объекта Aes
80 // с указанным ключом и IV.
81 using (Aes aesAlg = Aes.Create())
82 {
83 aesAlg.Key = Key;
84 aesAlg.IV = IV;
85
86 // Создайте дешифратор для выполнения преобразования потока.
87 ICryptoTransform decryptor = aesAlg.CreateDecryptor(aesAlg.Key, aesAlg.IV);
88
89 // Создаем поток для расшифровки
90 using (MemoryStream msDecrypt = new MemoryStream(cipherText))
91 {
92 using (CryptoStream csDecrypt = new CryptoStream(msDecrypt, decryptor,
93 CryptoStreamMode.Read))
94 {
95 using (StreamReader srDecrypt = new StreamReader(csDecrypt))
96 {
122 %
Проблемы не найдены.
Стр: 53
Симв: 49
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "aescrypto1" (проекты: 1 из 1)
aescrypto1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:30
17.04.2020


Файл Правка Вид Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
aescrypto1
Вход
Debug
Any CPU
aescrypto1
Live Share
Program.cs*
Список ошибок...
aescrypto1
Aes_Example.AesExample
DecryptStringFromBytes_Aes(byte[] cipherText, byte[] Key,
// Создайте дешифратор для выполнения преобразования потока.
ICryptoTransform decryptor = aesAlg.CreateDecryptor(aesAlg.Key, aesAlg.IV);
// Создаем поток для расшифровки
using (MemoryStream msDecrypt = new MemoryStream(cipherText))
{
using (CryptoStream csDecrypt = new CryptoStream(msDecrypt, decryptor,
CryptoStreamMode.Read))
{
using (StreamReader srDecrypt = new StreamReader(csDecrypt))
{
// Считывание расшифрованных байтов из дешифрующего потока
// и размещение их в строке.
plaintext = srDecrypt.ReadToEnd();
}
}
}
}
return plaintext;
}
}
}
122 %
Проблемы не найдены.
Стр: 99
Симв: 56
Пробелы
CRLF
Вывод
Показать выходные данные из:
Отладка
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "aescrypto1" (проекты: 1 из 1)
aescrypto1
Зависимости
Program.cs
Обозреватель решен...
Team Explorer — Подк...
Свойства
Обозреватель серверов
Панель элементов
Готово
Добавить в систему управления версиями
Введите здесь текст для поиска
РУС
10:33
17.04.2020


C:\Users\User\source\repos\aescrypto1\aescrypto1\bin\Debug\netcoreapp3.1\aescrypto1.exe
Оригинальный текст: Да здравсвует свет и скроется тьма!
Расшифрованный текст: Да здравсвует свет и скроется тьма!
Введите здесь текст для поиска
РУС
19:44
16.04.2020

Практическое задание

Разработать интерактивную программу, которая шифрует файлы и расшифровывает их, выполняющую следующие операции:

- Выбрать файл (вести путь с клавиатуры)
- Ввести имя зашифрованного файла
- Вести путь для размещения зашифрованного файла
- Создать зашифрованный файл, расширение файла должно соответствовать выбранному классу шифрования (AES, Rij).
- Выбрать файл для расшифровки файла.
- Ввести путь для расшифровываемого файла.
- Программа должна автоматически по расширению выбирать алгоритм для расшифровки.

СПАСИБО ЗА ВНИМАНИЕ!!!!