1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. **Институт кибербезопасности и защиты информации**

**ЛАБОРАТОРНАЯ РАБОТА № 8**

1. «Основы стенографической защиты информации»
2. по дисциплине «Основы информационной безопасности»
3. Выполнил
4. студент гр. 4831001/20003 Черникова В.М.

1. Руководитель
2. асс. преподавателя Орел Е.М.
4. Санкт-Петербург

2023

# 1. Цель работы

Приобретение навыков исследования свойств стегоконтейнеров, разработки стегосистем и их применения для сокрытия данных при передаче с помощью графических изображений.

# 2. Формулировка задания

1. Создать произвольный графический файл формата ВМР, который будет использоваться в качестве стегоконтейнера. Для этого следует с помощью графического редактора (например, paint, входящего в штатный набор программ операционной системы Windows), создать произвольный рисунок и сохранить его в соответствующем формате.
2. Открыть файл изображения с помощью НЕХ-редактора (например, НIEW). Найти основные блоки, входящие в файл соответствующего формата. Отметить наличие или отсутствие глобальной палитры. При наличии глобальной палитры указать се размер.
3. Создать программу-кодер, реализующую стеганографическое сокрытие "секретного" текстового файла в созданный графический файл. Программа должна соответствовать требованиям:
   1. для сокрытия текста должен применяться метод замены наименее значимых бит в глобальной палитре и в содержательной части графического файла по алгоритму: в каждом байте палитры младший бит заменяется битом шифруемого сообщения и полученный байт записывается в результирующий файл;
   2. помимо самого "секретного" сообщения, графический файл, используемый в качестве контейнера, должен включать в себя необходимую для восстановления информацию о скрываемом сообщения (тип скрываемого файла и его размер);
   3. во входных параметрах программы должна быть обозначена степень упаковки скрываемого сообщения, которая указывала бы на количество бит, помещаемое в один байт графического файла, например, один бит сообщения на байт графической информации (три бита на точку в цветовой схеме RGB) или два бита на байт графической информации (шесть битов на точку);
   4. если размеры скрываемого текста не позволяют полностью поместить этот текст в графический файл, при заданной степени упаковки, то процесс скрытия должен быть прерван с указанием причины завершения работы программы, размера текущего сообщения, максимальной вместимости стегоконтейнера, вычисленной по параметрам данного графического файла. Расчет производится по размерам глобальной палитры, скрываемого сообщения и соответствующей степени упаковки в графический файл.
4. Запустить разработанную программу-кодер на выполнение, указывая различные значения степени упаковки в качестве параметра командной строки исполняемого файла. При этом визуально сравнить полученные графические файлы, содержащие упакованное "секретное" сообщение, с исходным изображением. Сравнить размер графического файла (стегоконтейнера) до упаковки в нем стегосообщения и после. Отметить влияние степени упаковки на качество изображения, получаемого в результате сокрытия в нем стегосообщения.
5. Создать программу-декодер, осуществляющую извлечение "секретного" сообщения из стегоконтейнера.
6. Запустить разработанную программу-декодер, используя в качестве входного параметра графический файл, который был получен в п. 4. Извлечь "секретное" сообщение. Сравнить его с исходным.
7. Запустить программу-кодер на выполнение, задав в качестве входных параметров текстовый файл, размер которого превышает объем стегоконтейнера. Отметить полученные результаты
8. Открыть с помощью графического редактора файл изображения, содержащий "секретное" сообщение. Выполнить преобразования изображения: растяжение, поворот, отражение. Отметить параметры преобразования, которые были выполнены над графическим файлом. Сохранить результирующее изображение в тот же файл.
9. Запустить программу-декодер, используя в качестве входного параметра графический файл, который был получен в результате преобразований в п. 8. Сравнить расшифрованное и исходное сообщения.

# 3. Ход работы

Для начала работы создадим сам файл .bmp. Для простоты чтения там будет 3 цвета (RGB) и размером 30х30 пикселей. При сохранении рисунка в paint можно выбрать несколько режимов: 16-цветный, 256-цветный и монохромный рисунок. После сохранении одного рисунка как файлов у них разные размеры : 1, 2 , 1 Килобайты соответственно



Рис 1. Изображение

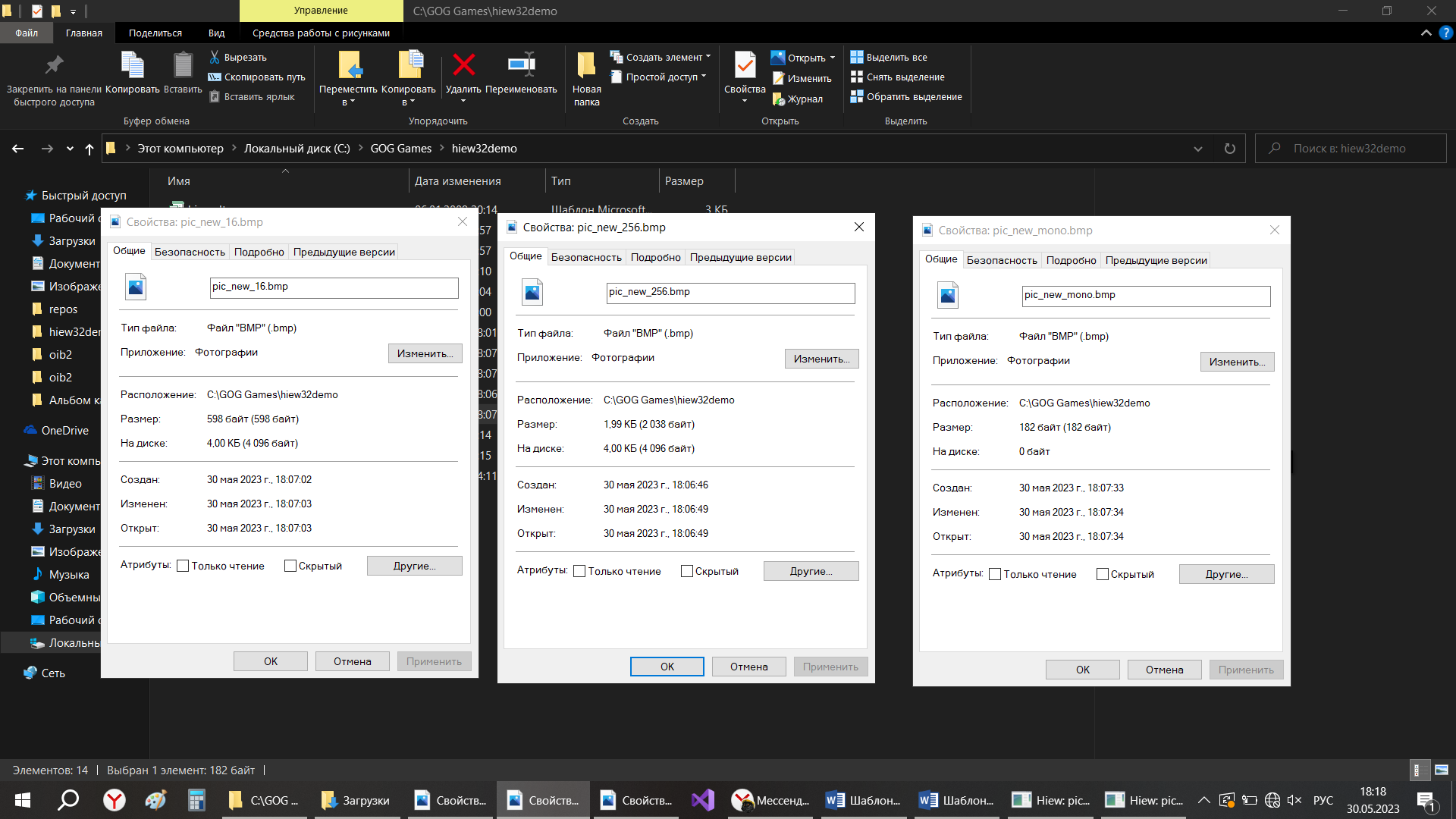


Рис 2. Общие свойства файлов

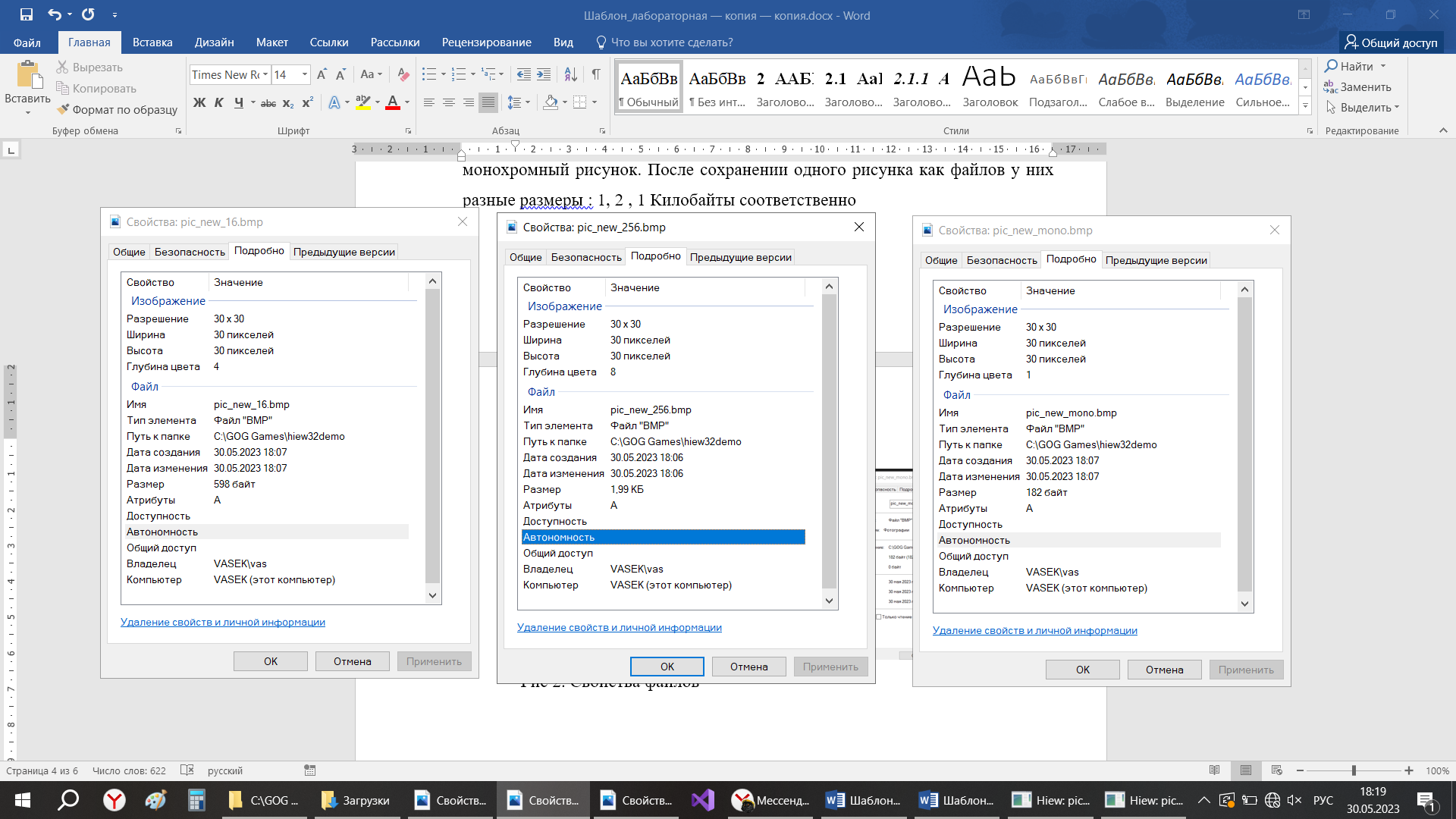


Рис 3. Подробные свойства файлов

Изображение, используемое для стегоконтейнера pic\_new\_24.bmp.

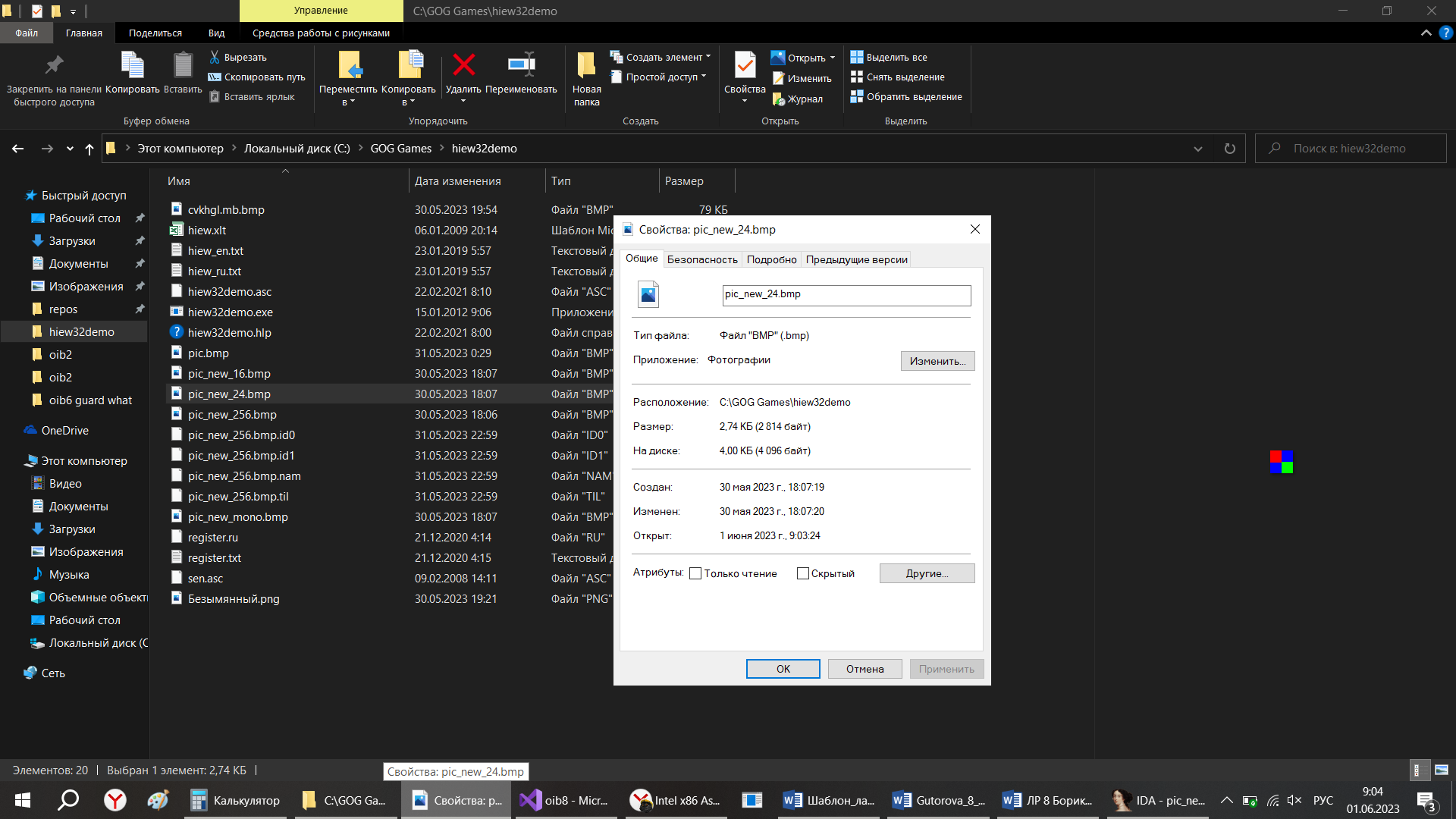
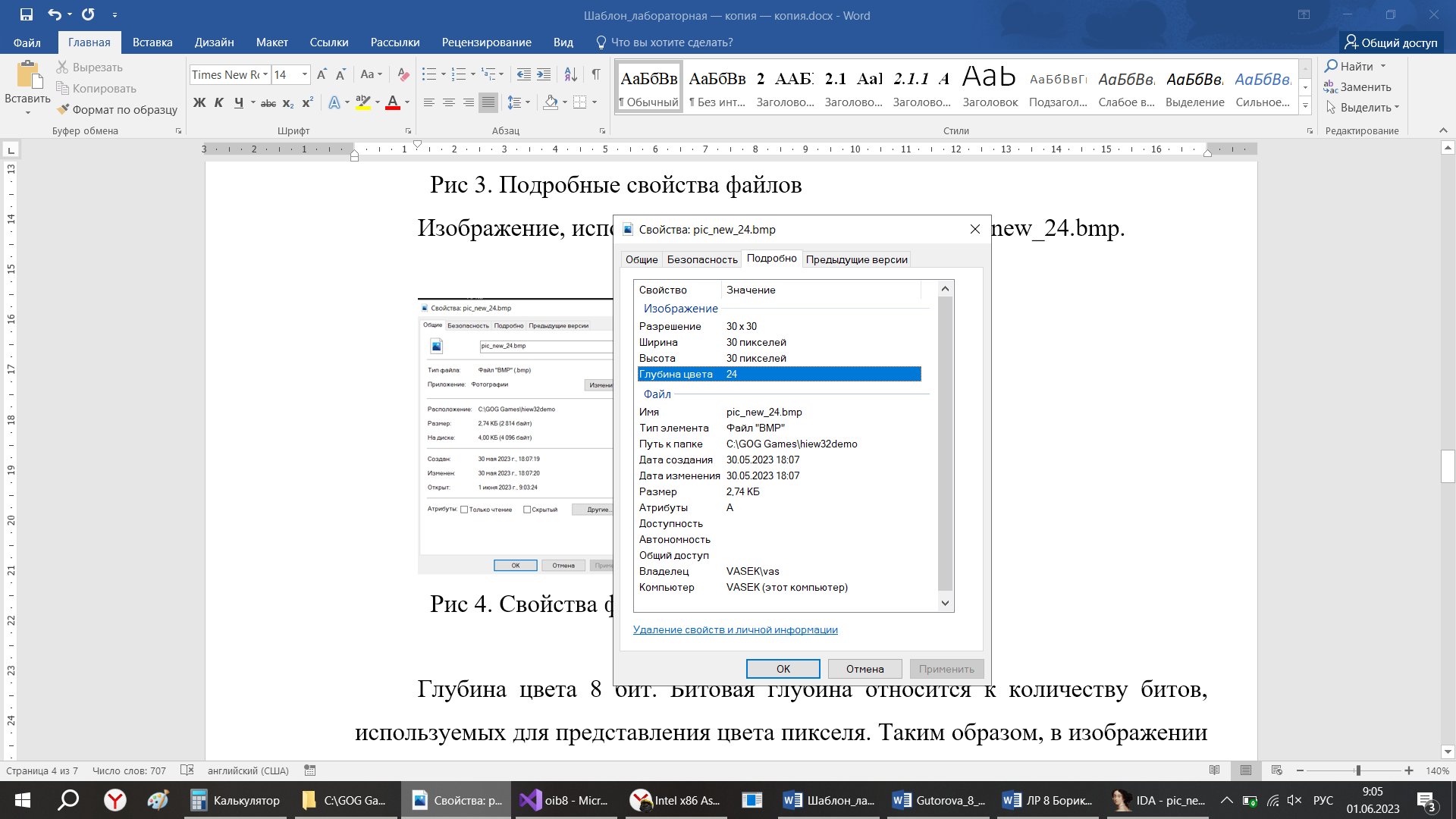
 

Рис 4. Свойства файла pic\_new\_24.bmp

Глубина цвета 24 бит. Информация о цвете пикселя представлена ​​24 битами, то есть для трех основных цветов BRG каждый цвет представлен байтовыми (8) битами Таким образом, в изображении может быть использовано различных цветов, поэтому глобальной палитры не существует, что также видно в дизассемблер-редакторе. Размер файла 3 Килобайта. Размер содержательной части файла: байт. Следовательно, размер стегоконтейнера 2700/8=337 байт.

Программа декодера не работает с повернутыми изображениями (а также перевернутыми, отраженными и так далее) и извлечь из него сообщение невозможно, так как практически любой способ обработки изображения приводит к разрушению сообщения.

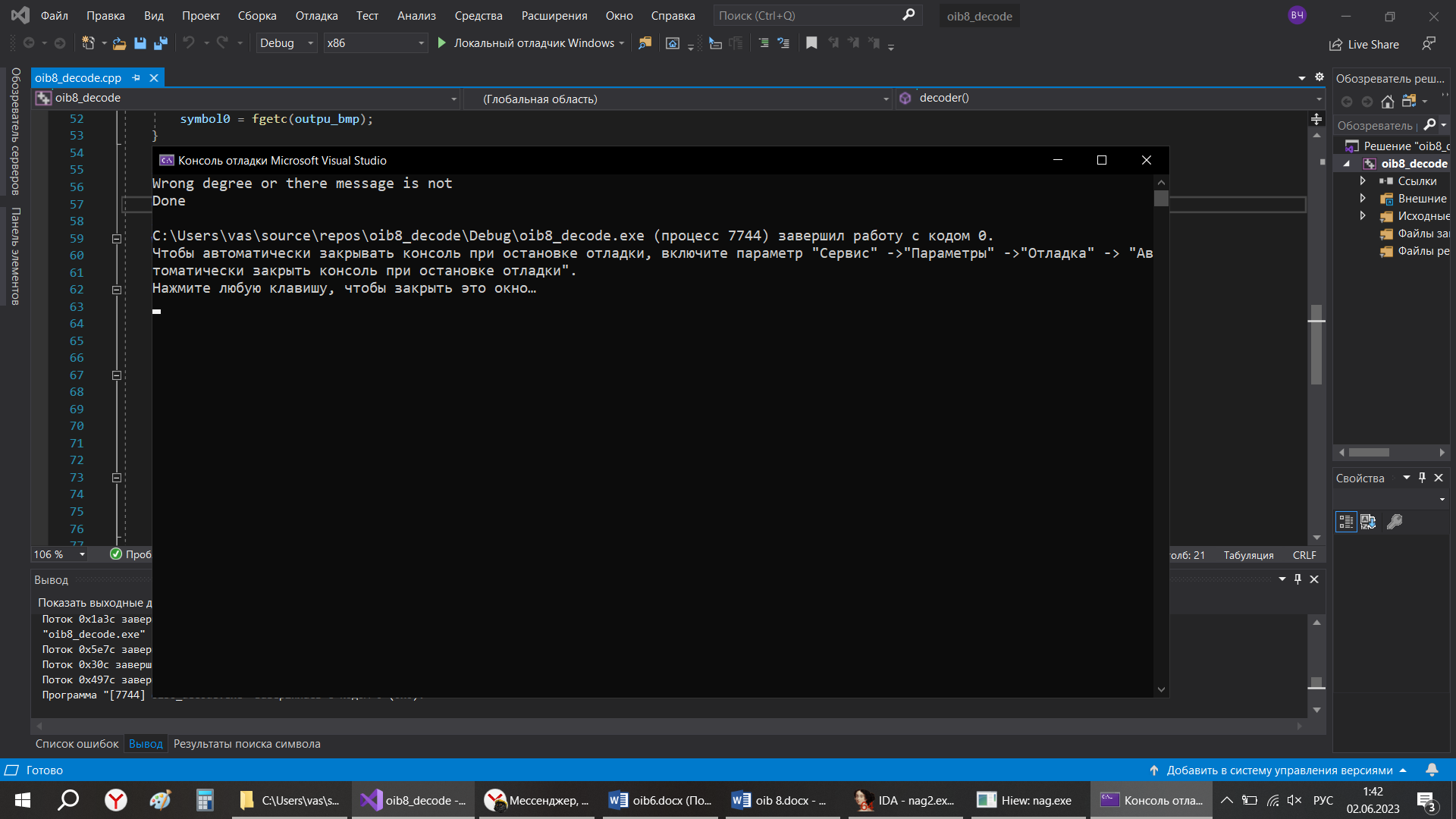


Рис 5. Сообщение декодера про измененном изображении

# 4. Результаты работы

В данной работе удалось скрыть информацию из текстового файла и прочитать ее, декодировав стегосообщение из стегоконтейнера. При декодировании были считаны малые биты из стегоконтейнера и записаны в созданный программой файл. Итоговый файл имел тот же размер, что и исходный, то есть был идентичен, что соответствует задаче.

# 5. Вывод

В ходе данной работы были приобретены навыки исследования свойств стегоконтейнеров, разработки стегосистем и их применения для сокрытия данных при передаче с помощью графических изображений.

**6. Дополнительные вопросы**

*Как используется глобальная палитра?*

Палитра присутствует только в BMP-файлах, содержащих палитровые изображения (с глубиной пикселей 8 бит и менее). К 8-битным изображениям прикладывается палитра, состоящая из не более чем 256 элементов. Седьмой бит (старший) указывает, есть ли глобальная палитра, а биты с 0 по 2 определяют её размер (от 21 до 28). Если флажок глобальной палитры выставлен в 0, её просто нет. А если есть, то размер всего блока в байтах равен количеству цветов в палитре помножить на три. Принцип заполнения: RGBRGBRGB.