

Лабораторная работа №3. Устранение защиты по регистрации оконного приложения с использованием сторонних инструментов

Цель: ознакомиться с возможностями дизассемблеров для восстановления, анализа и модификации исходных кодов .Net-приложений

Инструменты и дополнительные файлы: дизассемблер [ILSpy](#) с плагином [Reflexil](#), учебное приложение [HackOfCode.exe](#) и файл лицензии к нему [e1f35abgds537dhe59yud.lic](#).

Задание 1. Знакомство с учебным примером защищенного приложения

1. Загрузите файл учебного приложения **HackOfCode.exe** и запустите его. В результате откроется окно приложения с надписью “Unregistered”, указывающее на то, что используется незарегистрированное приложение:



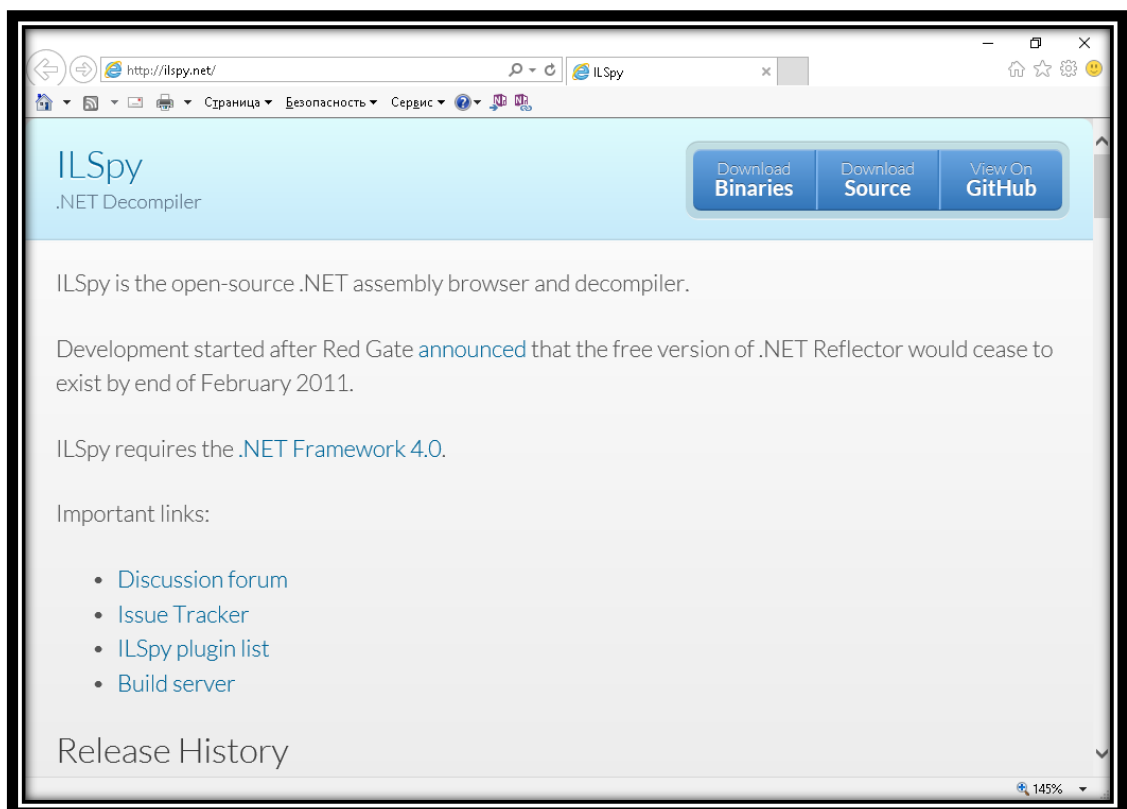
Функциональность приложения минимальна: оно отображает рисунок и завершает свою работу после нажатия кнопки «Ок». Пользователю предлагается зарегистрироваться и получить файл лицензии на право пользования приложением. При размещении этого файла в каталоге размещения приложения метка незарегистрированного приложения пропадает.

2. Загрузите файл лицензии на право пользования учебным приложением **e1f35abgds537dhe59yud.lic** и разместите его в каталоге размещения приложения **HackOfCode.exe**.
3. Запустите приложение **HackOfCode.exe** и убедитесь, что метка “Unregistered” не отображается:



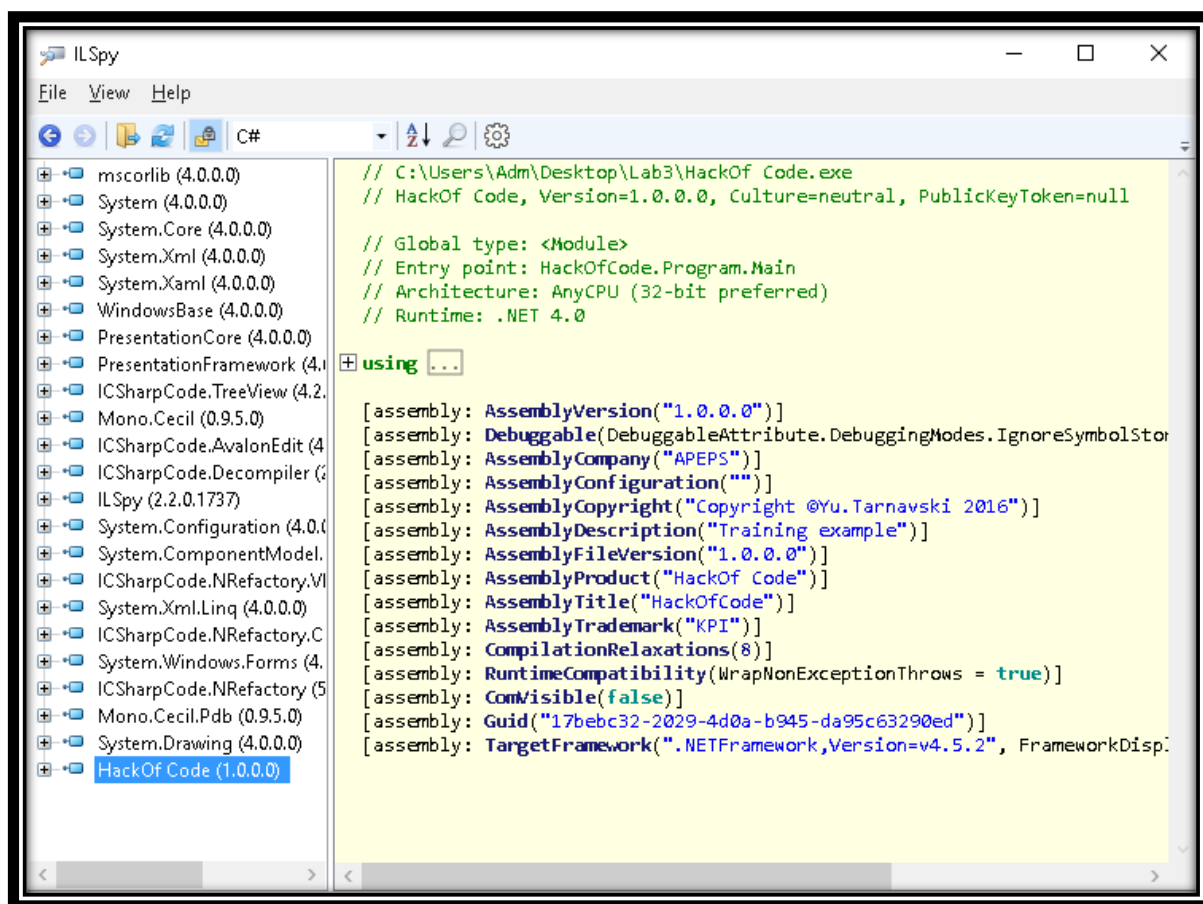
Задание 2. Дизассемблирование защищенного приложения с помощью ILSpy

1. На сайте программы [ILSpy .NET Decompiler](http://ilspy.net/) ознакомьтесь с требованиями к установке, функциональными особенностями, списком плагинов, демонстрационным роликом и скриншотами программы:

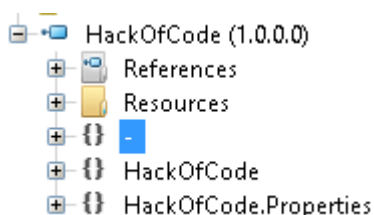


2. Загрузите последнюю версию дистрибутива **ILSpy**, поставляемую в виде zip-архива (**ILSpy_Master_2.4.0.1963_Binaries.zip**).
3. Распакуйте загруженный архив и запустите на выполнение файл **ILSpy.exe**.

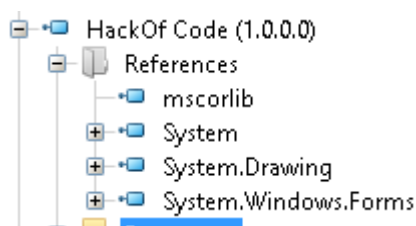
4. В окне приложения **ILSpy** откройте файл **HackOfCode.exe** (с помощью команды **File-Open...(Ctrl+O)**). Открываемый файл анализируется и помещается в древовидную структуру в левой части окна. При активизации узла в правой части окна можно ознакомиться с рядом свойств¹ исследуемой сборки, показывающие описание, правообладателя, организацию разработчика и т.п.:



5. Разверните узел анализируемого приложения, щелкнув по значку «+». В нем содержатся узлы ссылок (References), ресурсов (Resources) и кода приложения:

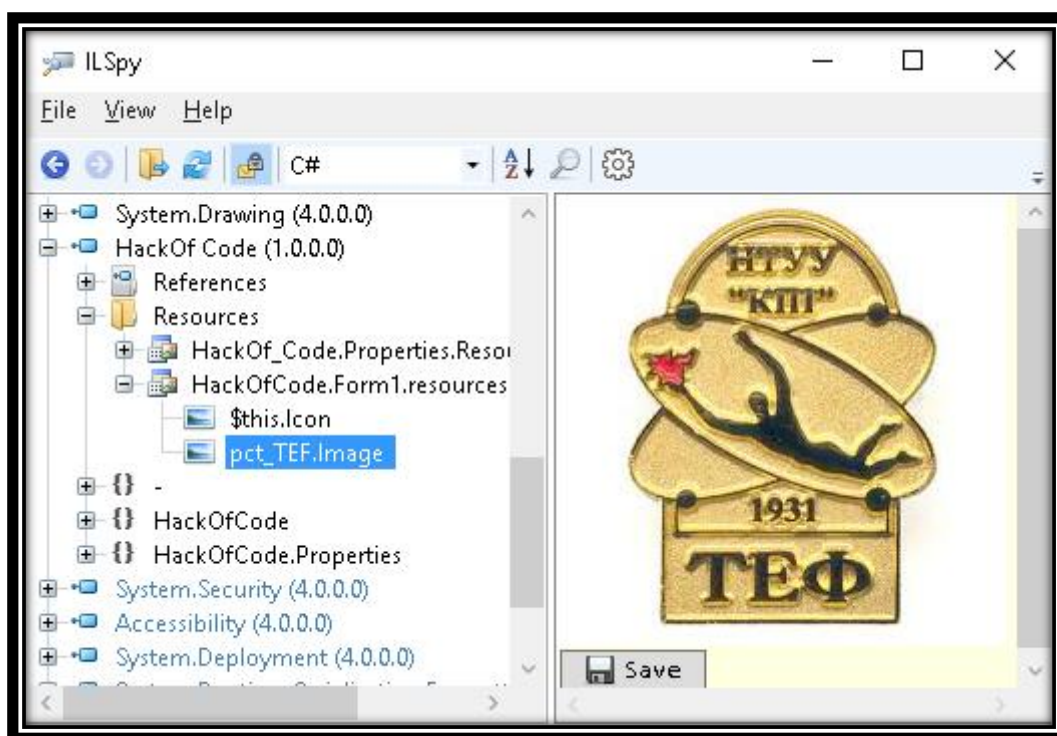


6. Разверните узел ссылок и ознакомьтесь со списком внешних сборок, которые используются приложением:

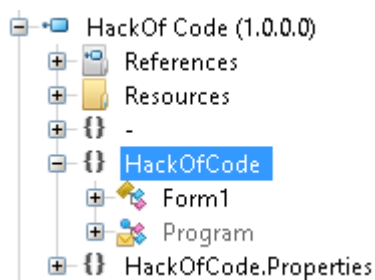



¹ Все эти свойства могут задаваться разработчиком в файле AssemblyInfo.cs проекта Visual Studio.

7. Разверните узел ресурсов и ознакомьтесь с тем, какие ресурсы используются приложением:

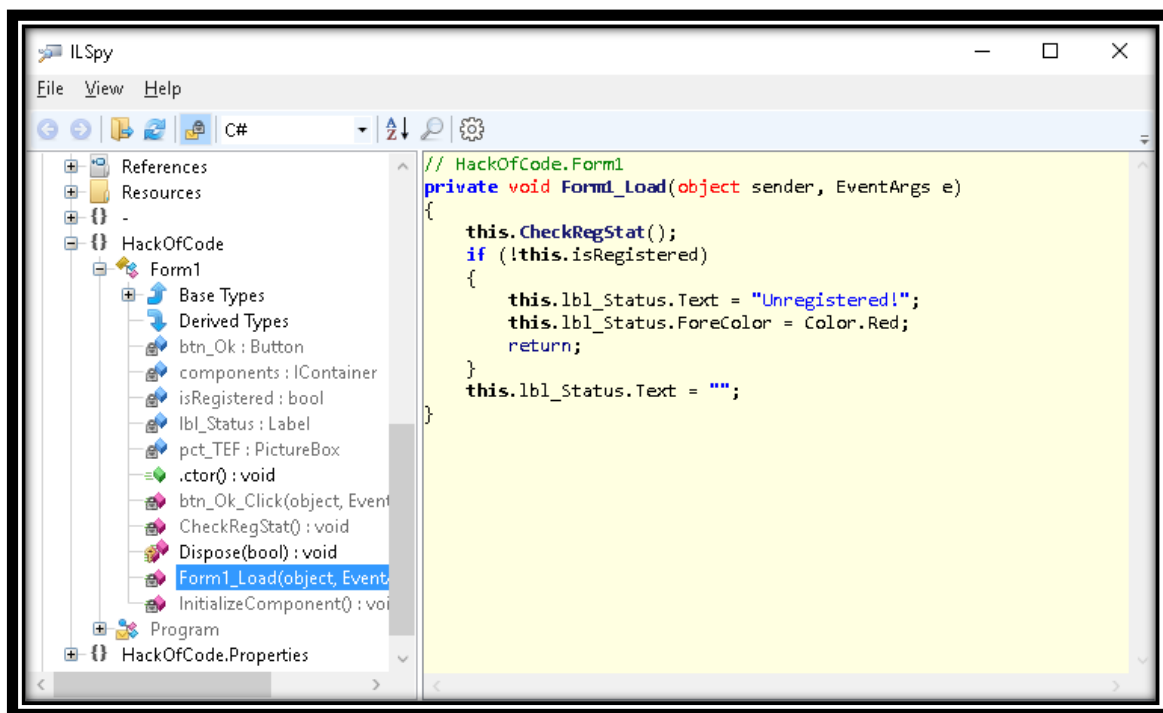


8. Разверните узел кода приложения **HackOfCode** и ознакомьтесь с тем, какие классы используются в приложении:

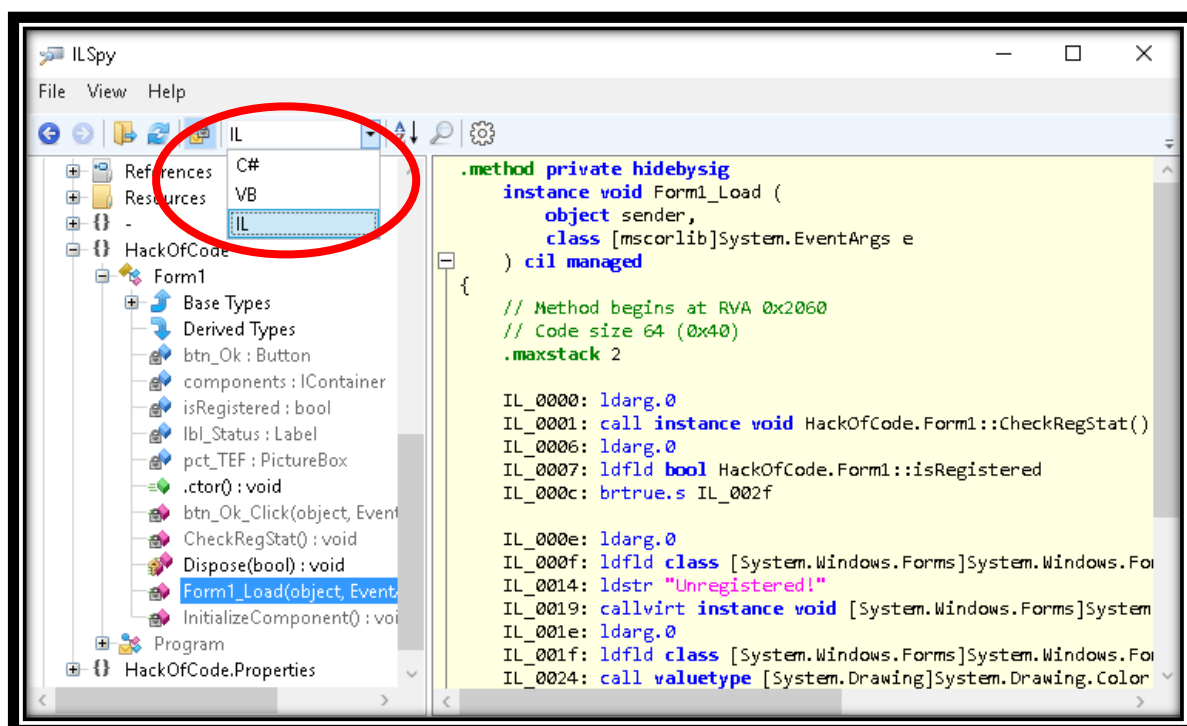


Здесь отображение классов с областью видения сборки (internal) можно включить/отключить с помощью команды **View>Show internal types and members** или с помощью кнопки  в панели инструментов.

9. Разверните узел кода класса **Form1** для просмотра его структуры. Код выбранного свойства или метода отображается в правой части окна:



10. По умолчанию для представления кода используется язык C#, но с помощью выпадающего списка в панели инструментов можно выбрать VB или IL:



11. В классе **Form1** выберите для просмотра метод **CheckRegStat ()**, в котором, как можно предположить, реализуется логика защиты приложения:

```
// HackOfCode.Form1
private void CheckRegStat()
{
    try
    {
        string[] files = Directory.GetFiles(Directory.GetCurrentDirectory(), "*.lic");
    }
}
```


```

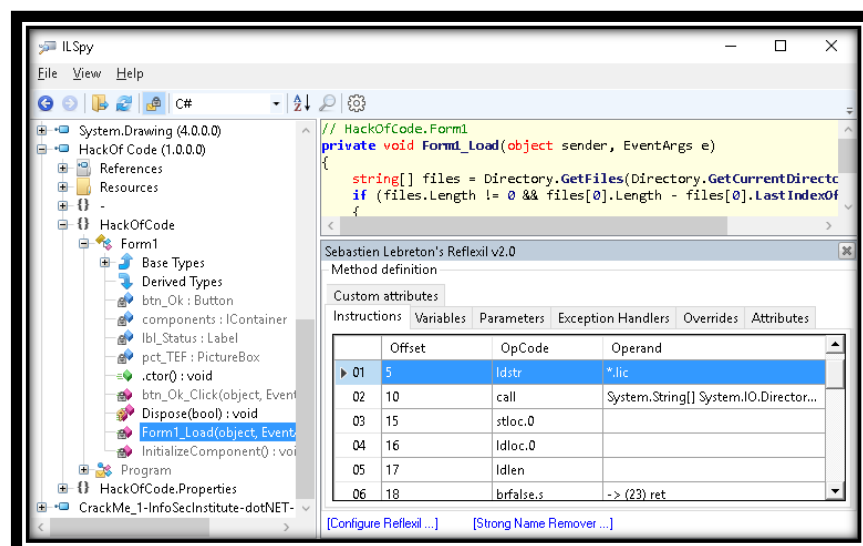
if (files.Length != 0)
{
    if (files[0].Length - files[0].LastIndexOf('h') == 11)
    {
        this.isRegistered = true;
    }
    else
    {
        MessageBox.Show("License wrong!", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
    }
}
else
{
    this.isRegistered = false;
}
}
catch (Exception arg_52_0)
{
    MessageBox.Show(arg_52_0.Message);
}
}

```

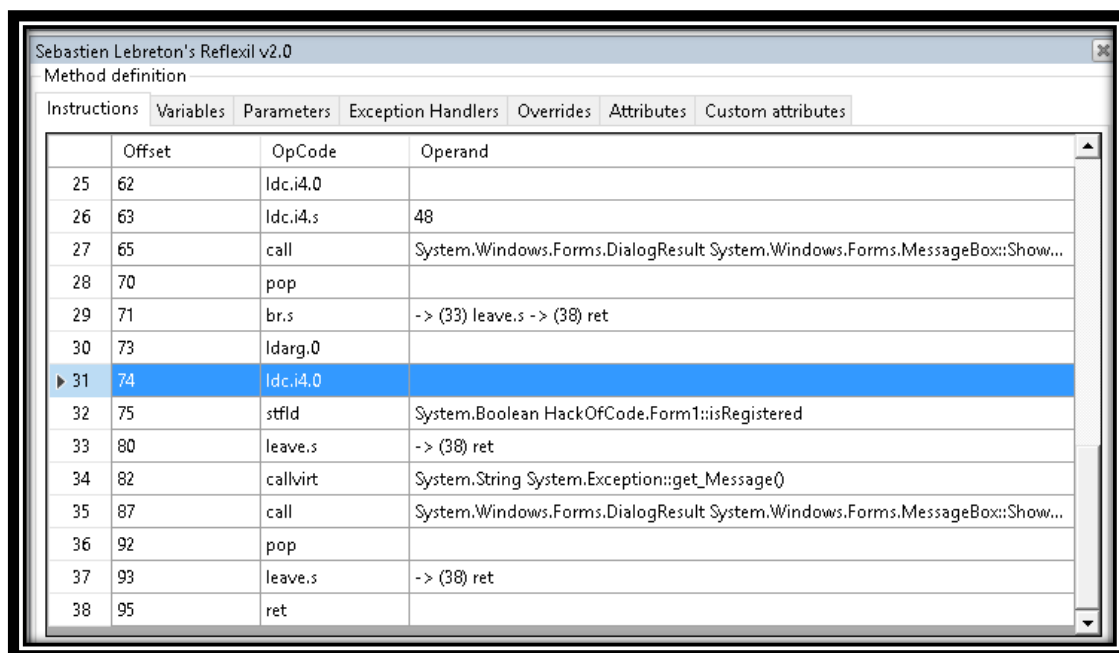
12. Проанализируйте логику работы метода **CheckRegStat ()** и убедитесь, что возможным способом обхода защиты есть замена во внешнем операторе **If-else** значения переменной **isRegistered** с **false** на **true**. Однако выполнить редактирование метода в окне **ILSpy** невозможно, если не установлен соответствующий плагин – **Reflexil**.

Задание 3. Установка и применение плагина Reflexil

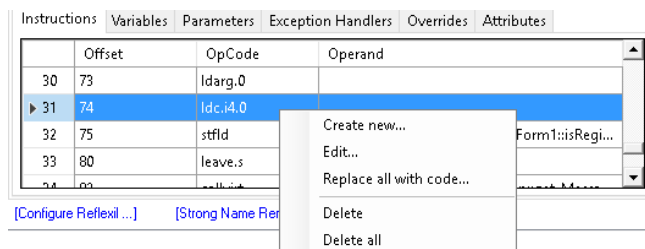
1. Зайдите на сайт плагина [Reflexil](#) и ознакомьтесь с его возможностями.
2. Загрузите zip-архив дистрибутива плагина.
3. Выполните установку плагина. Для этого извлеките из скачанного zip-архива файл библиотеки **Reflexil.ILSpy.Plugin.dll** и поместите его в папку установки **ILSpy**.
4. Вызовите установленный плагин, выбрав в главном меню **ILSpy** команду **View- Reflexil v2.0** или в панели инструментов кнопку .



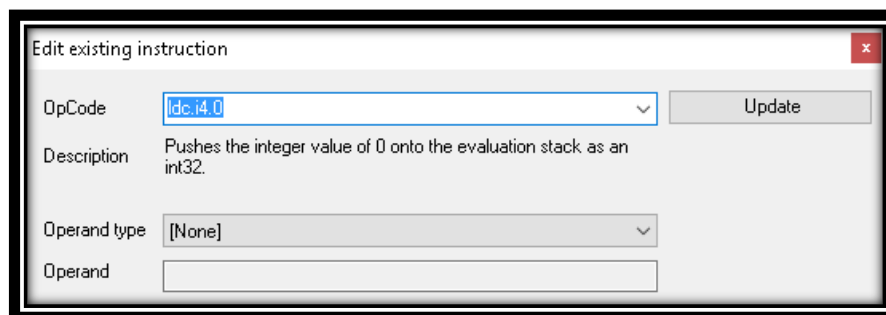
5. Выберите в дереве класса **Form1** метод **CheckRegStat ()** - в панели плагина на вкладке «**Instructions**» отобразятся II-команды.



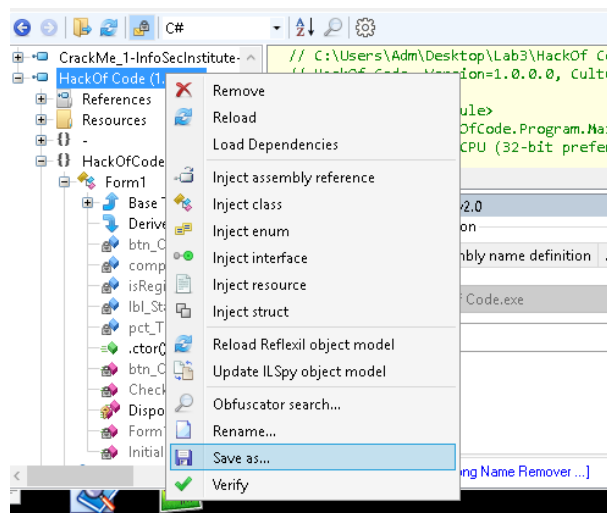
6. Найдите строку кода **31**, в которой инструкция **ldc.i4.0** задает значение **false** для переменной **isRegistered**.
7. Отредактируйте строку кода **31**, заменив инструкцию **ldc.i4.0** на **ldc.i4.1**. Для перехода к редактированию строки воспользуйтесь командой **Edit...** контекстного меню:



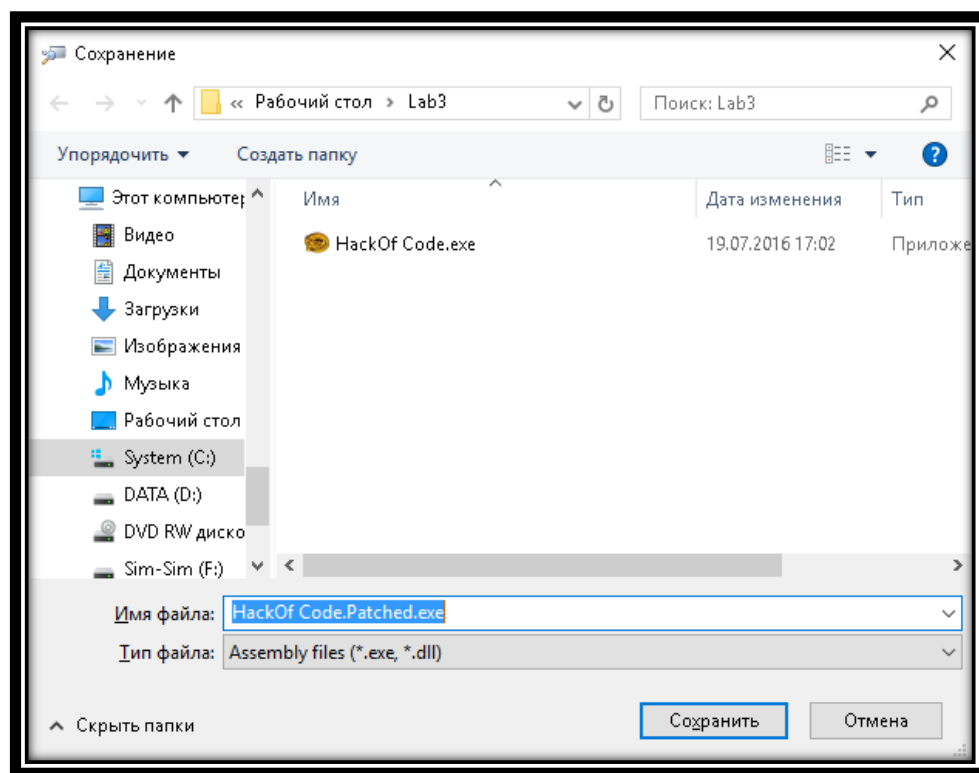
8. В окне редактирования инструкции выполните редактирование поля «**OpCode**» (заменяя **ldc.i4.0** на **ldc.i4.1**) и обновите его, нажав кнопку «**Update**»



9. Сохраните внесенные изменения в новом файле ***.exe**, воспользовавшись командой **Save as ...** контекстного меню:



10. Согласитесь с предлагаемым по умолчанию именем **HackOfCode.Patched.exe**:



11. Запустите файл **HackOfCode.Patched.exe** и убедитесь, что защита устранена.

Требование к отчету по лабораторной работе

Отчет должен содержать титульный лист с указанием темы и цели работы, результаты выполнения п.п.1,6-8,12 задания 2, п.п.1,1 задания 3, а также выводы по работе.

Контрольные вопросы

1. Каковы функциональные особенности программы **ILSpy**?
2. Какие требования к установке у программы **ILSpy**?
3. Как выполняется установка программы **ILSpy**?
4. Как просмотреть кода метода класса в **ILSpy**? Какие языки программирования могут использоваться для представления кода?
5. Допускается ли модификация кода в **ILSpy**?
6. Как устанавливается плагин **Reflexil** для **ILSpy**? Как и для чего он используется?