**Практикум. Использование объекта Mutex для создания приложения, исполняемого в единственном экземпляре**

Создайте простое консольное приложение, в котором объект Mutex будет применяться для того, чтобы разрешить запуск не более одного экземпляра приложе­ния.

Если при выполнении упражнения у вас возникнут затруднения, обратитесь к го­товым проектам в папке Code

1. Создайте консольное приложение с именем **SingleInstance**.

2. В основном файле импортируйте пространство имен System. Threading.

3. В методе main консольного приложения создайте локальную переменную Mutex и присвойте ей значение null (или Nothing в Visual Basic).

4. Создайте строку-константу с именем общего объекта Mutex. Присвойте ей значение \*RUNMEONCE\*.

5. Создайте блок try/catch.

6. В секции try блока try/catch вызовите метод Mutex.OpenExisting, используя строку, на­значенную на шаге 4 именем объекта Mutex. Затем присвойте результат переменной Mutex, созданной на шаге 2.

7. В секции catch блока try/catch block перехватите исключение WaitHandleCannotBeOpene- dException, чтобы определить, существует ли объект Mutex с таким именем.

8. Проверьте равенство переменной Mutex, созданной на шаге 2, null (или Nothing в Visual Basic), чтобы узнать, можно ли найти объект Mutex.

9. Если найти объект Mutex не удалось, создайте его и назовите именем, представлен­ным строкой из шага 4.

10. Если объект Mutex найден, закройте переменную Mutex и выйдите из приложения. Полученный код выглядит примерно так:

// C#

using System Threading;

class Program

{

static void Main(string[j args)

{

Mutex oneMutex = null;

const String MutexName = "RUNMEONLYONCE";

try // Пытаемся открыть Mutex

{

oneMutex = Mutex.OpenExisting(MutexName);

}

catch (WaitHandleCannotBeOpenedException)

{

// He можем открыть Mutex, потому что он не существует

}

// Создаем его, если он не существует if (oneMutex == null)

{

oneMutex = new Mutex(true, MutexName);

}

else

{

// Закрываем Mutex и выходим из приложения,

// так как разрешается запуск только одного его экземпляра

oneMutex.Close();

return;

}

Console WriteLineC'Our Application"),

Console.Read(),

}

}

11. Соберите проект, исправьте ошибки. Убедитесь, что запускается только один экзем­пляр приложения одновременно.