

# **Лабораторная работа №11**

## **Изучение методов авторизации и аутентификации в настольных приложениях**

### **1 Цель работы**

1.1 Познакомиться с методами авторизации и аутентификации в настольных приложениях.

### **2 Литература**

2.1 Зверева В. П., Сопровождение и обслуживание программного обеспечения компьютерных систем : учебник для студ. учреждений сред. проф. Образования / В. П. Зверева, А. В. Назаров. – М. : Издательский центр «Академия», 2018. – 256 с.

### **3 Подготовка к работе**

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

### **4 Основное оборудование**

4.1 Персональный компьютер.

### **5 Задание**

5.1 Разработать оконное приложение, использующее аутентификацию пользователя Windows

5.2 Разработать оконное приложение, использующее авторизацию пользователя через БД с распределением прав на три роли (роли определите самостоятельно)

5.3 Модифицировать программу из п.5.2 так, чтобы приложение использовало для авторизации вместо пароля файл ключа асимметричного шифрования, выбранного пользователем (см. п.9)

### **6 Порядок выполнения работы**

6.1 Повторить теоретический материал п. 3.1;

6.2 Выполнить задания 5.1-5.3

6.3 Ответить на контрольные вопросы п. 8;

6.4 Заполнить отчет п. 7.

### **7 Содержание отчета**

7.1 Титульный лист;

7.2 Цель работы;

7.3 Ответы на контрольные вопросы п. 6.3;

7.4 Вывод по проделанной работе.

### **8 Контрольные вопросы**

8.1 Что такое аутентификация?

## 8.2 Что такое авторизация?

## 9 Приложение

При регистрации пользователя используется алгоритм ассиметричного шифрования RSA. Публичным ключом шифруется некоторое сообщение и эти данные записываются в хранилище. Для авторизации необходимо расшифровать данное сообщение с использованием закрытого ключа, и если расшифровка прошла успешно, то пользователь авторизован.

### Примерная логика регистрации и авторизации:

```
// Генерация ключей
using (var rsa = RSA.Create())
{
    byte[] publicKey = rsa.ExportRSAPublicKey();
    byte[] privateKey = rsa.ExportRSAPrivateKey();

    // Сохранение приватного ключа в файл
    File.WriteAllBytes(privateKeyFilePath, privateKey);

    // Генерация тестового токена для проверки при авторизации
    byte[] testToken = RandomNumberGenerator.GetBytes(32);

    // Шифрование тестового токена публичным ключом
    rsa.ImportRSAPublicKey(publicKey, out _);
    byte[] encryptedToken = rsa.Encrypt(testToken,
RSAEncryptionPadding.Pkcs1);
    // Сохранение данных в базу
    SaveUserToDatabase(username, publicKey, encryptedToken);
    Console.WriteLine("Регистрация завершена. Сохраните
приватный ключ.");

    //Проверка валидности ключа путем расшифровки токена
    try{
        byte[] decryptedToken = rsa.Decrypt(testToken,
RSAEncryptionPadding.Pkcs1);
        Console.WriteLine("Успешная авторизация");
    }
    catch{
        Console.WriteLine("Ошибка авторизации");
    }
}
```