

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММИРОВАНИЯ

**ЛАБОРАТОРНАЯ РАБОТА № 5**

Выполнил студент:

Ершов Михаил Николаевич  
группа: М3206

Санкт-Петербург, 2019 г.

## Лабораторная работа №5

Создать класс Точка. Создать класс Треугольник (состоит из трех Точек).

Написать собственный сериализатор XML для класса Треугольник.

Проверить работоспособность сериализатора.

Написать код демонстрирующий сериализацию в бинарный файл, и десериализацию файла.

Написать собственную реализацию сохранения объекта типа Треугольник в реляционную базу данных, и восстановление объекта. Продемонстрировать работу реализации.

# 1 Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab5
{
    public class Program
    {
        static void Main(string[] args)
        {
            Triangle triangle = new Triangle() {
                A = new Point(0, 0),
                B = new Point(1, 2),
                C = new Point(10, 0)
            };
            Console.WriteLine($"{triangle.B.X} {triangle.B.Y}");
            triangle.saveToDB(@"Data Source = LAPTOP-ERSOVMN\SQLEXPRESS02;
                Initial Catalog = tehnarenok; Integrated Security = True;
                MultipleActiveResultSets = True");
            triangle.Serialize("input.xml");
            triangle.SaveToBinary("test");
            Triangle tr = Triangle.Deserialize("input.xml");
            Triangle trFromDb = Triangle.loadFromDB(@"Data Source = LAPTOP-
                ERSOVMN\SQLEXPRESS02; Initial Catalog = tehnarenok; Integrated
                Security = True; MultipleActiveResultSets = True", 1);
            Triangle trFromBinary = Triangle.loadFromBinary("test");
            triangle.Serialize("i.xml");
            Console.WriteLine($"{tr.B.X} {tr.B.Y}");
            Console.WriteLine($"{triangle.B.X} {triangle.B.Y}");
            Console.ReadKey();
        }
    }
}
```

## 2 Point.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab5
{
    [Serializable]
    public class Point
    {
        private int x, y;
        public Point(int x, int y)
        {
            this.x = x;
            this.y = y;
        }

        public Point()
        {
            this.x = 0;
            this.y = 0;
        }

        public int X { get => x; set => x = value; }
        public int Y { get => y; set => y = value; }
    }
}
```

### 3 Triangle.cs

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.IO;
using System.Linq;
using System.Runtime.Serialization.Formatters.Binary;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Xml.Serialization;

namespace lab5
{
    [Serializable]
    public class Triangle
    {
        public Point A { get; set; } = new Point();
        public Point B { get; set; } = new Point();
        public Point C { get; set; } = new Point();

        public void Serialize(string name)
        {
            XmlSerializer xmlSerializer = new XmlSerializer(typeof(Triangle));
            FileStream file = new FileStream(name, FileMode.OpenOrCreate);
            xmlSerializer.Serialize(file, this);
            file.Close();
        }

        public static Triangle Deserialize(string name)
        {
            XmlSerializer xmlSerializer = new XmlSerializer(typeof(Triangle));
            FileStream file = new FileStream(name, FileMode.OpenOrCreate);
            Triangle triangle = (Triangle)xmlSerializer.Deserialize(file);
            file.Close();
            return triangle;
        }

        public void SaveToBinary(string name)
        {
            BinaryFormatter binaryFormatter = new BinaryFormatter();
            FileStream file = new FileStream(name, FileMode.OpenOrCreate);
            binaryFormatter.Serialize(file, this);
            file.Close();
        }

        public static Triangle loadFromBinary(string name)
        {
            BinaryFormatter binaryFormatter = new BinaryFormatter();
            FileStream file = new FileStream(name, FileMode.OpenOrCreate);
            Triangle triangle = (Triangle)binaryFormatter.Deserialize(file);
            file.Close();
            return triangle;
        }
    }
}
```

```

public void saveToDB(string connectionString)
{
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();
    string command = $"INSERT INTO Triangle (A_x, A_y, B_x, B_y, C_x,
        C_y) VALUES ({A.X}, {A.Y}, {B.X}, {B.Y}, {C.X}, {C.Y})";
    SqlCommand sqlCommand = new SqlCommand(command, connection);
    sqlCommand.ExecuteNonQuery();
    connection.Close();
}

public static Triangle loadFromDB(string connectionString, int id)
{
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();
    string command = $"SELECT * FROM Triangle WHERE TriangleID = {id}";
    SqlCommand sqlCommand = new SqlCommand(command, connection);
    SqlDataReader sqlDataReader = sqlCommand.ExecuteReader();
    if (sqlDataReader.HasRows)
    {
        sqlDataReader.Read();
        var result = new Triangle
        {
            A = new Point
            {
                X = (int)sqlDataReader.GetValue(1),
                Y = (int)sqlDataReader.GetValue(2)
            },
            B = new Point
            {
                X = (int)sqlDataReader.GetValue(3),
                Y = (int)sqlDataReader.GetValue(4)
            },
            C = new Point
            {
                X = (int)sqlDataReader.GetValue(5),
                Y = (int)sqlDataReader.GetValue(6)
            }
        };
        sqlDataReader.Close();
        connection.Close();
        return result;
    }
    return null;
}
}
}

```