

## Лабораторная работа №5

### Цель работы

Приобретение навыков Xml сериализации (десериализации) классов в среде .Net.

Сериализация представляет собой процесс преобразования объекта в поток байтов для хранения объекта или передачи его в память, базу данных или файл. Ее основное назначение — сохранить состояние объекта для того, чтобы иметь возможность воссоздать его при необходимости. Обратный процесс называется десериализацией.

### Упражнение 1. Создание сериализуемого класса

Воспользуйтесь проектом приложения из лабораторной работы №3. Пометим классы, наследуемые от **Shape** атрибутом **[Serializable]**, а поля атрибутами **[XmlElement(...)]**.

```
[Serializable]
public class Triangle : Shape
{
    private double _sideAb;
    private double _sideBc;
    private double _sideCa;

    [XmlElement("SideAb")]
    public double SideAb...

    [XmlElement("SideBc")]
    public double SideBc...

    [XmlElement("SideCa")]
    public double SideCa...

    [XmlElement("Area")]
    public override double Area...
```

Помимо прочего необходимо добавить конструктор без параметров в базовый класс **Shape** и в наследники.

### Упражнение 2. Сериализация класса

После того, как все атрибуты классов установлены, можно приступить к сериализации.

```

XmlSerializer xmlSerializer = new XmlSerializer(typeof(Shape.Triangle));
StringWriter stringWriter = new StringWriter();
xmlSerializer.Serialize(stringWriter, t1);
Console.WriteLine(stringWriter.ToString());

```

Результатом работы станет вывод на консоль xml – строки с сериализованным классом:

```

<?xml version="1.0" encoding="utf-16"?>
<Triangle xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Name>Треугольник</Name>
  <Area>6</Area>
  <Perimeter>0</Perimeter>
  <SideAb>3</SideAb>
  <SideBc>4</SideBc>
  <SideCa>5</SideCa>
</Triangle>

```

*Самостоятельно реализуйте запись содержимого строки в файл.*

### **Упражнение 3. Десериализация класса**

Для того чтобы десериализовать класс, необходимо прочитать содержимое xml-файла и передать его в качестве параметра методу **Deserialize** класса **xmlSerializer**.

```

var xmlDeSerializer = new XmlSerializer(typeof(Shape.Triangle));
var stringReader = new StringReader(stringWriter.ToString());
Shape.Triangle triangle = (Shape.Triangle)xmlSerializer.Deserialize(stringReader);

```

*Реализуйте приложение, которое в случае наличия файлов сериализованных классов сразу осуществляло бы вывод всех параметров созданных геометрических фигур. В случае, если файлы сериализации отсутствуют, запросите у пользователя все необходимые параметры геометрических фигур и перед закрытием приложения сериализуйте классы фигур.*