

# Thornado: Synopsis

Примеры объявления классов и INI-файлов

Абдураманов П.Ю., Окуловский Ю.С.

## Программный код:

```
using AIRLab.Thornado;  
  
namespace ShellTest.Tests  
{  
    public partial class IntTest  
    {  
        [Thornado]  
        public int Number;  
    }  
}
```

## Входной файл:

```
Number=12
```

## Прочитанный объект:

```
[]  
Number = 12
```

## Программный код:

```
using AIRLab.Thornado;  
  
namespace ShellTest.Tests  
{  
    public partial class StringTest  
    {  
        [Thornado]  
        public string String;  
    }  
}
```

## Входной файл:

```
String=dfdjskf
```

## Прочитанный объект:

```
[]  
String = dfdjskf
```

## Программный код:

```
using AIRLab.Thornado;  
  
namespace ShellTest.Tests  
{  
    public partial class StringTest  
    {  
        [Thornado]  
        public string String;  
    }  
}
```

## Входной файл:

String@Null

## Прочитанный объект:

[]

## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public enum SomethingEnum
    {
        OneVal,
        TwoVal,
        ThreeVal
    }

    public partial class EnumTest
    {
        [Thornado]
        public SomethingEnum Enum;
    }
}
```

## Входной файл:

Enum=TwoVal

## Прочитанный объект:

[]

Enum = TwoVal

## Программный код:

```
using AIRLab.Thornado;  
using System.Collections.Generic;  
  
namespace ShellTest.Tests  
{  
    public partial class ListIntTest  
    {  
        [Thornado]  
        public List<int> List;  
    }  
}
```

## Входной файл:

List=1|2|3

## Прочитанный объект:

```
[]  
[List]  
0 = 1  
1 = 2  
2 = 3
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListIntTest
    {
        [Thornado]
        public List<int> List;
    }
}
```

## Входной файл:

```
List.0=1
List.1=2
```

## Прочитанный объект:

```
[]
[List]
0 = 1
1 = 2
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListIntTest
    {
        [Thornado]
        public List<int> List;
    }
}
```

## Входной файл:

```
List.*=1
List.*=2
```

## Прочитанный объект:

```
[]
[List]
0 = 1
1 = 2
```



## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListIntTest
    {
        [Thornado]
        public List<int> List;
    }
}
```

## Входной файл:

```
List.2=1
List.*=2
```

## Прочитанный объект:

```
[]
[List]
0 = 0
1 = 0
2 = 1
3 = 2
```

## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public partial class ArrayIntTest
    {
        [Thornado]
        public int[] Array;
    }
}
```

## Входной файл:

```
Array=1|2|3
```

## Прочитанный объект:

```
[]
Array@Size=3
Array.0 = 1
Array.1 = 2
Array.2 = 3
```

## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public partial class ArrayIntTest
    {
        [Thornado]
        public int[] Array;
    }
}
```

## Входной файл:

```
Array.0=1
Array.1=2
```

## Прочитанный объект:

```
[]
Array@Size=2
Array.0 = 1
Array.1 = 2
```

## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public partial class ArrayIntTest
    {
        [Thornado]
        public int[] Array;
    }
}
```

## Входной файл:

```
Array.*=1
Array.*=2
```

## Прочитанный объект:

```
[]
Array@Size=2
Array.0 = 1
Array.1 = 2
```

## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public partial class ArrayIntTest
    {
        [Thornado]
        public int[] Array;
    }
}
```

## Входной файл:

```
Array.2=1
Array.*=2
```

## Прочитанный объект:

```
[]
Array@Size=4
Array.0 = 0
Array.1 = 0
Array.2 = 1
Array.3 = 2
```

## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public partial class ArrayIntTest
    {
        [Thornado]
        public int[] Array;
    }
}
```

## Входной файл:

```
Array@Size=5
Array.*=1
Array.*=2
```

## Прочитанный объект:

```
[]
Array@Size=5
Array.0 = 1
Array.1 = 2
Array.2 = 0
Array.3 = 0
Array.4 = 0
```

## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public class A
    {
        [Thornado]
        public int Number;
        [Thornado]
        public string String;
    }
    public partial class ClassTest
    {
        [Thornado]
        public A Class;
    }
}
```

## Входной файл:

```
Class.Number = 12
Class.String = ab
```

## Прочитанный объект:

```
[]
[Class]
Number = 12
String = ab
```

## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public class A
    {
        [Thornado]
        public int Number;
        [Thornado]
        public string String;
    }
    public partial class ClassTest
    {
        [Thornado]
        public A Class;
    }
}
```

## Входной файл:

```
Class = Number:12 String:ab
```

## Прочитанный объект:

```
[]
[Class]
Number = 12
String = ab
```



## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public class A
    {
        [Thornado]
        public int Number;
        [Thornado]
        public string String;
    }
    public partial class ClassTest
    {
        [Thornado]
        public A Class;
    }
}
```

## Входной файл:

```
[Class]
Number=12
String=ab
```

## Прочитанный объект:

```
[]
[Class]
Number = 12
String = ab
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListListIntTest
    {
        [Thornado]
        public List<List<int>> List;
    }
}
```

## Входной файл:

```
[List]
0.0 = 1
0.1 = 2
1.0 = 1
1.1 = 2
```

## Прочитанный объект:

```
[]
[List]
[List.0]
0 = 1
1 = 2
[List]
[List.1]
0 = 1
1 = 2
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListListIntTest
    {
        [Thornado]
        public List<List<int>> List;
    }
}
```

## Входной файл:

```
[List.0]
0 = 1
1 = 2
[List.1]
0 = 1
1 = 2
```

## Прочитанный объект:

```
[]
[List]
[List.0]
0 = 1
1 = 2
[List]
[List.1]
0 = 1
1 = 2
[List]
```

## Программный код:

```
using AIRLab.Thornado;  
using System.Collections.Generic;  
  
namespace ShellTest.Tests  
{  
    public partial class ListListIntTest  
    {  
        [Thornado]  
        public List<List<int>> List;  
    }  
}
```

## Входной файл:

```
[List.0]  
* = 1  
* = 2  
[List.1]  
* = 1  
* = 2
```

## Прочитанный объект:

```
[]  
[List]  
[List.0]  
0 = 1  
1 = 2  
[List]  
[List.1]  
0 = 1  
1 = 2  
[List]
```

## Программный код:

```
using AIRLab.Thornado;  
using System.Collections.Generic;  
  
namespace ShellTest.Tests  
{  
    public partial class ListListIntTest  
    {  
        [Thornado]  
        public List<List<int>> List;  
    }  
}
```

## Входной файл:

```
[List.1]  
* = 1  
* = 2  
[List.*]  
* = 1  
* = 2
```

## Прочитанный объект:

```
[]  
[List]  
[List.0]  
[List]  
[List.1]  
0 = 1  
1 = 2  
[List]  
[List.2]  
0 = 1  
1 = 2  
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListListIntTest
    {
        [Thornado]
        public List<List<int>> List;
    }
}
```

## Входной файл:

```
List.0 = 1 | 2
List.1 = 3 | 4
```

## Прочитанный объект:

```
[]
[List]
[List.0]
0 = 1
1 = 2
[List]
[List.1]
0 = 3
1 = 4
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListListIntTest
    {
        [Thornado]
        public List<List<int>> List;
    }
}
```

## Входной файл:

```
List.*@Null
List.* = 1 | 2
List.* = 3 | 4
```

## Прочитанный объект:

```
[]
[List]
[List.1]
0 = 1
1 = 2
[List]
[List.2]
0 = 3
1 = 4
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListATest
    {
        [Thornado]
        public List<A> List;
    }
}
```

## Входной файл:

```
[List]
0.Number = 1
0.String = A
1.Number = 2
1.String = B
```

## Прочитанный объект:

```
[]
[List]
[List.0]
Number = 1
String = A
[List]
[List.1]
Number = 2
String = B
[List]
```



## Программный код:

```
using AIRLab.Thornado;  
using System.Collections.Generic;  
  
namespace ShellTest.Tests  
{  
    public partial class ListATest  
    {  
        [Thornado]  
        public List<A> List;  
    }  
}
```

## Входной файл:

```
[List.0]  
Number = 1  
String = A  
[List.1]  
Number = 2  
String = B
```

## Прочитанный объект:

```
[]  
[List]  
[List.0]  
Number = 1  
String = A  
[List]  
[List.1]  
Number = 2  
String = B  
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListATest
    {
        [Thornado]
        public List<A> List;
    }
}
```

## Входной файл:

```
[List.1]
Number = 1
String = A
[List.*]
Number = 2
String = B
```

## Прочитанный объект:

```
[]
[List]
[List.0]
Number = 0
[List]
[List.1]
Number = 1
String = A
[List]
[List.2]
Number = 2
String = B
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListATest
    {
        [Thornado]
        public List<A> List;
    }
}
```

## Входной файл:

```
List.0 = Number:1 String:A
List.1 = Number:2 String:B
```

## Прочитанный объект:

```
[]
[List]
[List.0]
Number = 1
String = A
[List]
[List.1]
Number = 2
String = B
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListATest
    {
        [Thornado]
        public List<A> List;
    }
}
```

## Входной файл:

```
List.*@Null
List.* = Number:1 String:A
List.* = Number:2 String:B
```

## Прочитанный объект:

```
[]
[List]
[List.1]
Number = 1
String = A
[List]
[List.2]
Number = 2
String = B
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListATest
    {
        [Thornado]
        public List<A> List;
    }
}
```

## Входной файл:

```
List = Number:1 String:A | Number:2 String:B
```

## Прочитанный объект:

```
[]
[List]
[List.0]
Number = 1
String = A
[List]
[List.1]
Number = 2
String = B
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class ListATest
    {
        [Thornado]
        public List<A> List;
    }
}
```

## Входной файл:

```
List = Number:1 String:A | Number:2 String:B
List.* = Number:3 String:C
```

## Прочитанный объект:

```
[]
[List]
[List.0]
Number = 1
String = A
[List]
[List.1]
Number = 2
String = B
[List]
[List.2]
Number = 3
String = C
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public class B : A
    {
        [Thornado]
        public int Num;
        [Thornado]
        public string Str;
    }
    public partial class ListObjectTest
    {
        [Thornado]
        public List<A> List;
    }
}
```

## Входной файл:

```
[List.0]
Number=1
String=A
[List.1]
@Type=B
Number=1
String=A
Num=2
Str=B
```

## Прочитанный объект:

```
[]
[List]
[List.0]
Number = 1
String = A
[List]
[List.1]
@Type = ShellTest.Tests.B
Num = 2
Str = B
Number = 1
String = A
[List]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class DictIntIntTest
    {
        [Thornado]
        public Dictionary<int, int> Dict;
    }
}
```

## Входной файл:

```
Dict.123 = 1
Dict.13 = 2
Dict.55 = 3
```

## Прочитанный объект:

```
[]
[Dict]
123 = 1
13 = 2
55 = 3
```



## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class DictIntIntTest
    {
        [Thornado]
        public Dictionary<int, int> Dict;
    }
}
```

## Входной файл:

```
Dict = 123:1 13:2 55:3
```

## Прочитанный объект:

```
[]
[Dict]
123 = 1
13 = 2
55 = 3
```

## Программный код:

```
using AIRLab.Thornado;  
using System.Collections.Generic;  
  
namespace ShellTest.Tests  
{  
    public partial class DictIntIntTest  
    {  
        [Thornado]  
        public Dictionary<int, int> Dict;  
    }  
}
```

## Входной файл:

```
[Dict]  
123=1  
13=2  
55=3
```

## Прочитанный объект:

```
[]  
[Dict]  
123 = 1  
13 = 2  
55 = 3
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class DictStringIntTest
    {
        [Thornado]
        public Dictionary<string, int> Dict;
    }
}
```

## Входной файл:

```
Dict.A = 1
Dict.B = 2
Dict.C = 3
```

## Прочитанный объект:

```
[]
[Dict]
A = 1
B = 2
C = 3
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class DictStringIntTest
    {
        [Thornado]
        public Dictionary<string, int> Dict;
    }
}
```

## Входной файл:

```
Dict = A:1 B:2 C:3
```

## Прочитанный объект:

```
[]
[Dict]
A = 1
B = 2
C = 3
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class DictStringIntTest
    {
        [Thornado]
        public Dictionary<string, int> Dict;
    }
}
```

## Входной файл:

```
[Dict]
A=1
B=2
C=3
```

## Прочитанный объект:

```
[]
[Dict]
A = 1
B = 2
C = 3
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class DictIntATest
    {
        [Thornado]
        public Dictionary<int, A> Dict;
    }
}
```

## Входной файл:

```
Dict.123 = Number:1 String:A
Dict.2 = Number:2 String:B
Dict.55 = Number:3 String:C
```

## Прочитанный объект:

```
[]
[Dict]
[Dict.123]
Number = 1
String = A
[Dict]
[Dict.2]
Number = 2
String = B
[Dict]
[Dict.55]
Number = 3
String = C
[Dict]
```

## Программный код:

```
using AIRLab.Thornado;
using System.Collections.Generic;

namespace ShellTest.Tests
{
    public partial class DictIntATest
    {
        [Thornado]
        public Dictionary<int, A> Dict;
    }
}
```

## Входной файл:

```
[Dict.123]
Number=1
String=A
[Dict.2]
Number=2
String=B
[Dict.55]
Number=3
String=C
```

## Прочитанный объект:

```
[]
[Dict]
[Dict.123]
Number = 1
String = A
[Dict]
[Dict.2]
Number = 2
String = B
[Dict]
[Dict.55]
Number = 3
String = C
[Dict]
```

## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public partial class ColorTest
    {
        [Thornado]
        public System.Drawing.Color Color;
    }
}
```

## Входной файл:

```
[Color]
R=200
G=100
B=200
```

## Прочитанный объект:

```
[]
Color.R = 200
Color.G = 100
Color.B = 200
```



## Программный код:

```
using AIRLab.Thornado;

namespace ShellTest.Tests
{
    public partial class ColorTest
    {
        [Thornado]
        public System.Drawing.Color Color;
    }
}
```

## Входной файл:

Color = R:200 G:100 B:200

## Прочитанный объект:

```
[]
Color.R = 200
Color.G = 100
Color.B = 200
```

## Программный код:

```
using AIRLab.Thornado;  
  
namespace ShellTest.Tests  
{  
    public partial class ColorTest  
    {  
        [Thornado]  
        public System.Drawing.Color Color;  
    }  
}
```

## Входной файл:

Color = R:200 G:100

## Прочитанный объект:

Ошибка: [Error]Чтение файла,  
линия 1: Не все Immutable  
заполнены.

Адреса:

Color

## Программный код:

```
using AIRLab.Thornado;
using System;

namespace ShellTest.Tests
{
    public struct Imm
    {
        [Thornado]
        public int X;
        [Thornado]
        public int Y;
        public Imm(int x, int y)
        {
            X = x;
            Y = y;
        }
        static Imm()
        {
            ImmutableShell<Imm>.AddConstructor(new Func<int,int, Imm>((X, Y) => new Imm(X, Y)));
        }
    }

    public partial class ImmutableTest
    {
        [Thornado]
        public Imm Immutable;
    }
}
```

## Входной файл:

```
Immutable.X = 1
Immutable.Y = 2
```

## Прочитанный объект:

```
[]
Immutable.X = 1
Immutable.Y = 2
ImmutableShell<Imm>.AddConstructor(new Func<int,int, Imm>((X, Y) => new Imm(X, Y)));
```

## Программный код:

```
using AIRLab.Thornado;  
using System;
```

```
namespace ShellTest.Tests
```

```
{  
    public struct Imm  
    {  
        [Thornado]  
        public int X;  
        [Thornado]  
        public int Y;  
        public Imm(int x, int y)  
        {  
            X = x;  
            Y = y;  
        }  
        static Imm()  
        {  
            ImmutableShell<Imm>.AddConstructor(new Func<int,int, Imm>((X, Y) => new Imm(X, Y)));  
        }  
    }  
  
    public partial class ImmutableTest  
    {  
        [Thornado]  
        public Imm Immutable;  
    }  
}
```

## Входной файл:

```
Immutable = X:1 Y:2
```

## Прочитанный объект:

```
[]  
Immutable.X = 1  
Immutable.Y = 2
```

## Программный код:

```
using AIRLab.Thornado;
using System;

namespace ShellTest.Tests
{
    public struct Imm
    {
        [Thornado]
        public int X;
        [Thornado]
        public int Y;
        public Imm(int x, int y)
        {
            X = x;
            Y = y;
        }
        static Imm()
        {
            ImmutableShell<Imm>.AddConstructor(new Func<int, int, Imm>((X, Y) => new Imm(X, Y)));
        }
    }

    public partial class ImmutableTest
    {
        [Thornado]
        public Imm Immutable;
    }
}
```

## Входной файл:

```
[Immutable]
X = 1
Y = 2
```

## Прочитанный объект:

```
[]
Immutable.X = 1
Immutable.Y = 2
```

## Программный код:

```
using AIRLab.Thornado;
using System;

namespace ShellTest.Tests
{
    public struct Imm
    {
        [Thornado]
        public int X;
        [Thornado]
        public int Y;
        public Imm(int x, int y)
        {
            X = x;
            Y = y;
        }
        static Imm()
        {
            ImmutableShell<Imm>.AddConstructor(new Func<int,int, Imm>((X, Y) => new Imm(X, Y)));
        }
    }

    public partial class ImmutableTest
    {
        [Thornado]
        public Imm Immutable;
    }
}
```

## Входной файл:

Immutable.X = 1

## Прочитанный объект:

Ошибка: [Error]Чтение файла,  
линия 1: Не все Immutable  
заполнены.  
Адреса:  
Immutable

## Программный код:

```
using AIRLab.Thornado;  
using System;
```

```
namespace ShellTest.Tests
```

```
{  
    public class DateFormat : ValueFormat<TimeSpan>  
    {  
        public DateFormat()  
            : base(a => a.TotalSeconds.ToString(), str => TimeSpan.FromSeconds(ulong.Parse(str)), () => TimeSpan.FromTicks(1))  
        {}  
    }  
  
    public partial class FormatTest  
    {  
        [Thornado]  
        public TimeSpan Date;  
    }  
}
```

Входной файл:

```
Date = 10
```

Прочитанный объект:

```
[]
```

```
Date = 10
```