Сериализуемый класс: class F { int i1, i2, i3, i4, i5;}

код сериализации-десериализации:

using System.Text;

using System.Text.RegularExpressions;

namespace ReflectionOTUSHomework

{

public class MyReflection

{

public static string MySerializerToString<T>(T obj)

{

var sb = new StringBuilder();

sb.Append("{");

var properties = typeof(T).GetProperties();

foreach (var property in properties)

{

var name = property.Name;

var value = property.GetValue(obj).ToString();

sb.Append($"\"{name}\":{value},");

}

if (sb.Length > 0)

{

sb.Length--;

}

sb.Append("}");

return sb.ToString();

}

public static T MyDeserializerFromString<T>(string str) where T : new ()

{

var obj = new T();

var properties = typeof(T).GetProperties();

var regex = new Regex("\"([^\"]+)\":(\\d+)");

var matches = regex.Matches(str);

foreach (Match match in matches)

{

var propertyName = match.Groups[1].Value;

var propertyValue = match.Groups[2].Value;

// Находим свойство с таким именем в классе

foreach (var property in properties)

{

if (property.Name == propertyName && property.CanWrite)

{

if (int.TryParse(propertyValue, out int value))

{

property.SetValue(obj, value);

}

break;

}

}

}

return obj;

}

}

}

количество замеров: 100000 итераций

мой рефлекшен:  
  
Время на сериализацию = 88 мс  
  
Время на десериализацию = 772 мс  
  
стандартный механизм (NewtonsoftJson):  
  
Время на сериализацию = 364 мс  
  
Время на десериализацию = 317 мс