

stringstream.cm

```
/*  
  
    Copyright (c) 2012–2016 Seppo Laakko  
    http://sourceforge.net/projects/cmajor/  
  
    Distributed under the GNU General Public License, version 3 (GPLv3).  
    (See accompanying LICENSE.txt or http://www.gnu.org/licenses/gpl.html  
    )  
  
*/  
  
// Copyright (c) 1994  
// Hewlett–Packard Company  
// Copyright (c) 1996  
// Silicon Graphics Computer Systems, Inc.  
// Copyright (c) 2009 Alexander Stepanov and Paul McJones  
  
using System;  
  
namespace System.IO  
{  
    public class InputStringStream : InputStream  
    {  
        public nothrow InputStringStream(): str(), pos(0)  
        {  
        }  
        public nothrow InputStringStream(const string& str_): str(str_),  
            pos(0)  
        {  
        }  
        suppress InputStringStream(const InputStringStream&);  
        suppress void operator=(const InputStringStream&);  
        public default InputStringStream(InputStringStream&&);  
        public default void operator=(InputStringStream&&);  
        public nothrow void SetStr(const string& str_)  
        {  
            str = str_;  
            pos = 0;  
        }  
        public nothrow const string& GetStr() const  
        {  
            return str;  
        }  
        public override string ReadLine()  
        {  
            string line;  
            while (pos < str.Length())  
            {  
                char c = str[pos++];  

```

```

        if (c == '\n')
        {
            return line;
        }
        line.Append(c);
    }
    return line;
}
public override string ReadToEnd()
{
    int p = pos;
    pos = str.Length();
    return str.Substring(p);
}
public nothrow override bool EndOfStream() const
{
    return pos == str.Length();
}
private string str;
private int pos;
}

public class OutputStringStream: OutputStream
{
    public OutputStringStream(): str()
    {
    }
    public OutputStringStream(const string& str_): str(str_)
    {
    }
    suppress OutputStringStream(const OutputStringStream&);
    suppress void operator=(const OutputStringStream&);
    public default OutputStringStream(OutputStringStream&&);
    public default void operator=(OutputStringStream&&);
    public const string& GetStr() const
    {
        return str;
    }
    public void SetStr(const string& str_)
    {
        str = str_;
    }
    public override void Write(const char* s)
    {
        str.Append(s);
    }
    public override void Write(const string& s)
    {
        str.Append(s);
    }
    public override void Write(char c)
    {
        str.Append(c);
    }
}

```

```

}
public override void Write(byte b)
{
    str.Append(ToString(b));
}
public override void Write(sbyte b)
{
    str.Append(ToString(b));
}
public override void Write(short s)
{
    str.Append(ToString(s));
}
public override void Write(ushort u)
{
    str.Append(ToString(u));
}
public override void Write(int i)
{
    str.Append(ToString(i));
}
public override void Write(uint i)
{
    str.Append(ToString(i));
}
public override void Write(long l)
{
    str.Append(ToString(l));
}
public override void Write(ulong u)
{
    str.Append(ToString(u));
}
public override void Write(bool b)
{
    str.Append(ToString(b));
}
public override void Write(float f)
{
    str.Append(ToString(f));
}
public override void Write(double d)
{
    str.Append(ToString(d));
}
public override void WriteLine()
{
    str.Append( '\n' );
}
public override void WriteLine(const char* s)
{
    str.Append(s);
    str.Append( '\n' );
}

```

```

}
public override void WriteLine(const string& s)
{
    str.Append(s);
    str.Append( '\n' );
}
public override void WriteLine(char c)
{
    str.Append(c);
    str.Append( '\n' );
}
public override void WriteLine(byte b)
{
    str.Append( ToString(b) );
    str.Append( '\n' );
}
public override void WriteLine(sbyte b)
{
    str.Append( ToString(b) );
    str.Append( '\n' );
}
public override void WriteLine(short s)
{
    str.Append( ToString(s) );
    str.Append( '\n' );
}
public override void WriteLine(ushort u)
{
    str.Append( ToString(u) );
    str.Append( '\n' );
}
public override void WriteLine(int i)
{
    str.Append( ToString(i) );
    str.Append( '\n' );
}
public override void WriteLine(uint i)
{
    str.Append( ToString(i) );
    str.Append( '\n' );
}
public override void WriteLine(long l)
{
    str.Append( ToString(l) );
    str.Append( '\n' );
}
public override void WriteLine(ulong u)
{
    str.Append( ToString(u) );
    str.Append( '\n' );
}
public override void WriteLine(bool b)
{

```

```

        str.Append(ToString(b));
        str.Append( '\n' );
    }
    public override void WriteLine(float f)
    {
        str.Append(ToString(f));
        str.Append( '\n' );
    }
    public override void WriteLine(double d)
    {
        str.Append(ToString(d));
        str.Append( '\n' );
    }
    private string str;
}
}

```