

```
#region Copyright (c) 2019 TRUST 47 Fund. All Rights Reserved.
//-----//
//          TRUST 47 Fund          //
//          Copyright (c) 2019 All Rights reserved //
//          //                       //
//          //                       //
// This file and its contents are protected by United States and //
// International copyright laws. Unauthorized reproduction and/or //
// distribution of all or any portion of the code contained herein //
// is strictly prohibited and will result in severe civil and criminal //
// penalties. Any violations of this copyright will be prosecuted //
// to the fullest extent possible under law. //
//          //                       //
// THE SOURCE CODE CONTAINED HEREIN AND IN RELATED FILES IS PROVIDED //
// TO THE REGISTERED DEVELOPER FOR THE PURPOSES OF EDUCATION AND //
// TROUBLESHOOTING. UNDER NO CIRCUMSTANCES MAY ANY PORTION OF THE SOURCE //
// CODE BE DISTRIBUTED, DISCLOSED OR OTHERWISE MADE AVAILABLE TO ANY //
// THIRD PARTY WITHOUT THE EXPRESS WRITTEN CONSENT OF TRUST 47 Fund //
//          //                       //
// UNDER NO CIRCUMSTANCES MAY THE SOURCE CODE BE USED IN WHOLE OR IN //
// PART, AS THE BASIS FOR CREATING A PRODUCT THAT PROVIDES THE SAME, OR //
// SUBSTANTIALLY THE SAME, FUNCTIONALITY AS ANY TRUST 47 Fund PRODUCT. //
//          //                       //
// THE REGISTERED DEVELOPER ACKNOWLEDGES THAT THIS SOURCE CODE //
// CONTAINS VALUABLE AND PROPRIETARY TRADE SECRETS OF TRUST 47 Fund //
// THE REGISTERED DEVELOPER AGREES TO EXPEND EVERY EFFORT TO //
// INSURE ITS CONFIDENTIALITY. //
//          //                       //
// THE END USER LICENSE AGREEMENT (EULA) ACCOMPANYING THE PRODUCT //
// PERMITS THE REGISTERED DEVELOPER TO REDISTRIBUTE THE PRODUCT IN //
// EXECUTABLE FORM ONLY IN SUPPORT OF APPLICATIONS WRITTEN USING //
// THE PRODUCT. IT DOES NOT PROVIDE ANY RIGHTS REGARDING THE //
// SOURCE CODE CONTAINED HEREIN. //
//          //                       //
// THIS COPYRIGHT NOTICE MAY NOT BE REMOVED FROM THIS FILE. //
//-----//
#endregion Copyright (c) 2019 TRUST 47 Fund. All Rights Reserved.
```

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

#region No Warranty
/*=====
 * NO WARRANTY
 * Mediacrypt/IT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED,
 * INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OR MERCHANTABILITY
 * AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SOFTWARE
 * AND THE ACCOMPANYING WRITTEN MATERIALS. NO LIABILITY FOR CONSEQUENTIAL DAMAGES.
 * IN NO EVENT SHALL Mediacrypt/IT OR ITS SUPPLIERS BE LIABLE
 * FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION,
 * DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION,
 * LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS)
 * ARISING OUT OF THE USE OF OR INABILITY TO USE THIS Mediacrypt/IT PRODUCT,
 * EVEN IF Mediacrypt/IT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 *=====*/
#endregion No Warranty

namespace NXTPBlocks.Blockchain.Demo02
{
    public class Blockchain
    {
        public IList<Block> Chain { set; get; }

        public Blockchain()
        {
            InitializeChain();
            AddGenesisBlock();
        }

        public void InitializeChain()
        {
            Chain = new List<Block>();
        }
    }
}
```

```
public Block CreateGenesisBlock()
{
    return new Block(DateTime.Now, null, "{}");
}

public void AddGenesisBlock()
{
    Chain.Add(CreateGenesisBlock());
}

public Block GetLatestBlock()
{
    return Chain[Chain.Count - 1];
}

public void AddBlock(Block block)
{
    Block latestBlock = GetLatestBlock();
    block.Index = latestBlock.Index + 1;
    block.PreviousHash = latestBlock.Hash;
    block.Hash = block.CalculateHash();
    Chain.Add(block);
}

public bool IsValid()
{
    for (int i = 1; i < Chain.Count; i++)
    {
        Block currentBlock = Chain[i];
        Block previousBlock = Chain[i - 1];

        if (currentBlock.Hash != currentBlock.CalculateHash())
        {
            return false;
        }

        if (currentBlock.PreviousHash != previousBlock.Hash)
```

```
        {
            return false;
        }
    }
}
```