

Универзитет у Крагујевцу  
Природно-математички факултет  
Крагујевац

# IMPACT

## УВОД У ETHEREUM

**Ментори:**

Др Бобан Стојановић  
Филип Бојовић  
Андреја Живић  
Лазар Крстић

**Аутори:**

Немања Тракић



## Садржај

|  |   |
|--|---|
| УВОД У ETHEREUM.....                           | 1 |
| Шта је <i>Blockchain</i> ? .....               | 3 |
| <i>Blockchain</i> .....                        | 3 |
| Шта је <i>Ethereum</i> ? .....                 | 4 |
| Ethereum .....                                 | 4 |
| Паметни уговори – <i>Smart contracts</i> ..... | 4 |
| Ether.....                                     | 5 |
| DAPPS.....                                     | 5 |
| Имплементација Nethereum-a у .NET.....         | 6 |

## Шта је *Blockchain*?

### *Blockchain*

*Blockchain* је децентрализована база података која се не налази на једном месту већ је састављена од мањих база(блокова), које су међусобно повезане. У њима се налазе информације о дигиталним трансакцијама. Свакој трансакцији додељује се временска ознака. Трансакција се додаје у хронолошком реду већ постојећем блоку.

*Block* („Блок“) је фајл у коме се трајно евидентирају подаци који се односе на *Bitcoin* мрежу. Блок бележи најновије трансакције које још нису унете у неки од претходних блокова.

*Chain* („Ланац“) представља низ блокова који се криптографски односи на свог родитеља. Измена блока захтева промену свих наредних блокова што захтева сагласност целе мреже(консензус).

Недостатке *Bitcoin*а допуњује *Ethereum*. *Ethereum* представља велику и важну иновацију на тржишту криптовалута. Омогућено је да програмери дефинишу и покрећу паметне уговоре. *Ethereum* је, попут *Bitcoin*а, расподељена јавна *Blockchain* мрежа. Један од недостатака *Bitcoin*-а које је исправио *Ethereum*. На пример P2P (*Peer-to-peer*). *Ethereum* омогућава и извршавање програмског кода било које децентрализоване апликације.

Иновација која то омогућава је *Ethereum* виртуелна машина(*Ethereum Virtual Machine* – у даљем тексту *EVM*).

Нови блокови се емитују на чворове у мрежи, проверавају и верификују, ажурирајући стање за све.

Начин на који *blockchain* хешира податке, можете погледати [овде](#).

## Шта је *Ethereum*?

### Ethereum

*Ethereum* је тренутно базиран на методу Доказа о раду (*PoW*) као и Bitcoin, најављена је промена метода у Доказ о улогу (*PoS*). Рудари копају криптовалуту Ether која има исту функцију као и Bitcoin али се користи за извршавање трансакција у оквиру овог Blockchain система. Захваљујућу Ethereum-у процес је знатно олакшан уз помоћ алата који се нуде за прављење децентрализованих апликација, односно писање паметних уговора.

Било који програмер може створити паметни уговор и учинити га јавним за мрежу, користећи blockchain као свој слој података, уз накнаду плаћену мрежи. Тада сваки корисник може да позове паметни уговор да изврши свој код, опет уз накнаду плаћену мрежи.

### Паметни уговори – *Smart contracts*

Паметни уговори представљају најважнију функцију коју *Blockchain* систем омогућава. Они су аутоматски, „само-извршавајући“, дигитални уговори. Класични уговори су претворени у програмски код, расподељени у сачувани у оквиру целе мреже. Њих се придржавају и извршавају сви уређаји који учешћем прихватају све услове који су дефинисани у програмском коду.

Најважнији напредак који Blockchain доноси је у начину на који он елиминише посредника и чини уговор сигурнијим. Попут класичних уговора дефинисана су правила и казне које се аутоматски спроводе.

У пракси, учесници не пишу нови код сваки пут када желе да захтевају прорачун на EVM. Уместо тога, програмери апликација преносе програме (исечке кода за виšekратну употребу) у складиште EVM, а затим корисници подnose захтеве за извршавање ових исечака кода са различитим параметрима. Програме учитане у мрежу и извршене помоћу мреже називамо *smart contracts* („паметним уговорима“).

Паметни уговори пружају аутономију, поверење, уштеду, сигурности и ефикасност. Предности паметних уговора се огледају у томе што нема посредника (штеди време и новац), систем је шифрован и непристрасан.

Упркос томе што ова технологија пуно обећава и даље може бити склон проблемима. Такође, многа питања још увек нису регулисана законом. Већина проблема постоји искључиво зато што је ова технологија још увек млада, а она ће сигурно бити усавршена временом.

## Ether

*Ether* је валута *Ethereum-a*, Свака активност на овој мрежи се наплаћује *Ether-om*, а рудари који успешно генеришу нови блок у ланцу бивају награђени *Ether-om*. *Ether* се може претворити у доларе користећи Крипто-мењачнице.

Ако се *Ether* користи за плаћање накнада, а његова цена се мења из дана у дан, доведи до тога би цена услуге једног дана била веома висока, а већ следећег ниска. За ублажавање овог проблема користи се унутрашња валута *GAS*. Трошкови извршења и коришћења ресурса унапред су одређени у *Ethereum-u* преко *GAS-a*.

## DAPPS

Децентрализоване апликације(*DAPPS*, *dApps*) су компјутерске апликације које се извршавају на дистрибуираним системима. Популарност су добиле захваљујући *Ethereum-u* где се користе као паметни уговори.

Децентрализоване апликације се, за разлику од типичних апликација, извршавају на P2P мрежи. Обично су отвореног кода.

Најпознатије *DAPPS* апликације су Brave и Tor.

## Имплементација *Nethereum-a* у *.NET*

*Nethereum* је *.NET* библиотека отвореног кода за *Ethereum*, поједностављује управљање и интеракцију паметних уговорима са *Ethereum*-овим чворовима.

*Nethereum* се развија по стандарду *netstandard 1.1*, *net451*, а такође и као преносна библиотека, стога је компатибилан са свим популарним оперативним системима (*Windows*, *Linux*, *MacOS*, *Android* и *OSX*) и тестиран је на *cloud*, мобилним уређајима, десктопу, *Xbox -y*, *hololens -y* и *windows IoT*.

Предстојећа издања биће у складу са *Ethereum 2.0* (када се изда *Ethereum 2.0*) и укључиваће функције попут *DevP2P*, *Plasma* и микро-плаћања.

*Nethereum* ради на *.NET Core* и *.NET Framework* (од *V4.5.1*). Потребно је инсталирати *.NET SDK* ([Download .Net SDK](#)).

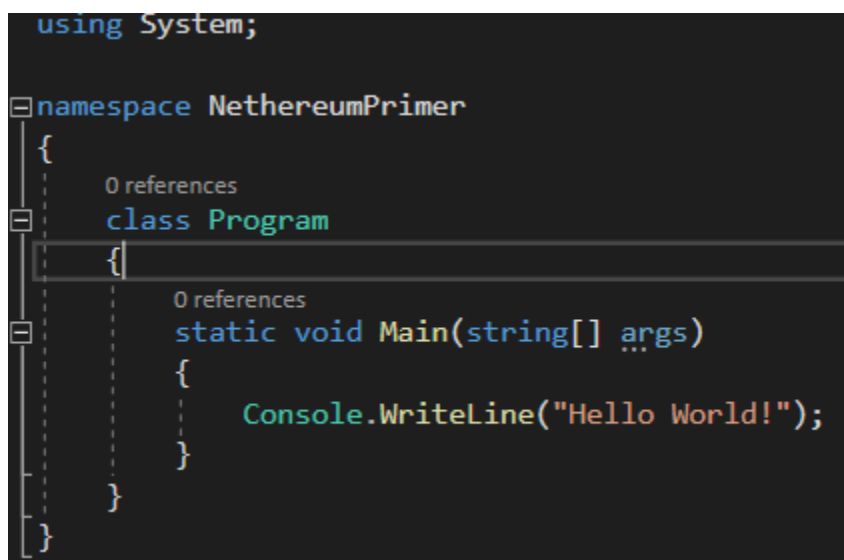
Креирање пројекта се може извршити уз помоћ *Visual Studio* или *CommandPrompt*-ом наредним командама:

```
dotnet new console -o NethereumPrimer  
cd NethereumPrimer
```

Додајте референцу на пакет *Nethereum.Web3* и ажурирајте пројектне пакете:

```
dotnet add package Nethereum.Web3  
dotnet restore
```

Покренути *Visual Studio*, датотеку *Program.cs* би требало изгледати овако:



```
using System;  
  
namespace NethereumPrimer  
{  
    0 references  
    class Program  
    {  
        0 references  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Hello World!");  
        }  
    }  
}
```

Следећи корак је додавањем наредне линије на почетку фајла.  
*using Nethereum.Web3;*

```
using System;
using System.Threading.Tasks;
using Nethereum.Web3;

namespace NethereumPrimer
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            GetAccountBalance().Wait();
            Console.ReadLine();
        }

        1 reference
        static async Task GetAccountBalance()
        {
            var web3 = new Web3("https://mainnet.infura.io/v3/7238211010344719ad14a89db874158c");
            var balance = await web3.Eth.GetBalance.SendRequestAsync("0xde0b295669a9fd93d5f28d9ec85e40f4cb697bae");
            Console.WriteLine($"Balance in Wei: {balance.Value}");

            var etherAmount = Web3.Convert.FromWei(balance.Value);
            Console.WriteLine($"Balance in Ether: {etherAmount}");
        }
    }
}
```

`var web3 = new Web3("https://mainnet.infura.io/v3/7238211010344719ad14a89db874158c");`

У овој линији је извршено инстанцирање променљиве *web3*.

За овај пример користићемо посебан *API* кључ али за пројекат је потребно да се пријавимо на [INFURA](https://infura.io/) и генеришемо наш кључ.

Износ је враћен у Wei формату, најмања јединица вредности. Можемо га конвертовати у Ether на следећи начин:

`var etherAmount = Web3.Convert.FromWei(balance.Value);`