

# **ВВЕДЕНИЕ В ETHEREUM**

# НЕМНОГО ИСТОРИИ

Идея была предложена в конце 2013 г. (4 года после релиза Биткойна).

Первый релиз - 30 июня 2015 г.

Автор - Виталик Бутерин.

На данный момент используется 4-я версия протокола.

# ETHEREUM VS BITCON

| BTC                            | ETH   |
|--------------------------------|---|
| Граф транзакций                | Машина состояний                            |
| Общее число монет - 21 млн BTC | Общее число монет не ограничено             |
| Платежная система              | Платформа для децентрализованных приложений |

# БЛОКЧЕЙН

В блокчейне эфириума хранится история состояний одной большой машины состояний.

Для добавления блоков используется PoW, но планируется плавный переход на PoE.

# МАЙНИНГ

- Новые блоки генерируются сетью примерно раз в 15 секунд.
- Майнерам выплачивается вознаграждение, даже если блок не попал в блокчейн (блоки-оммеры)
- Скорость добычи монет не меняется в рамках одной версии кода. Общее число монет не ограничено.

# СЧЕТА

- Внешние счета (обычные)
- Счета смарт-контрактов

# ТРАНЗАКЦИИ

- Перевод средств
- Создание смарт контракта
- Отправка сообщения смарт контракту

# ОПЛАТА ТРАНЗАКЦИЙ

Транзакции оплачиваются *газом*:

- За перевод средств
- За хранение данных
- За создание смарт контракта
- За исполнение смарт контракта



# ГАЗ

Это просто условные единицы для оплаты транзакций.

Задаются два параметра:

- `gasPrice` - цена за единицу *газа*
- `gasLimit` - лимит единиц *газа*, которые можно потратить на транзакцию

*Газ* расходуется на разные операции с разным коэффициентом

# СМАРТ КОНТРАКТЫ

Небольшие программы, которые:

- хранятся в блокчейне
- хранят свои данные в блокчейне
- исполняются майнерами при генерации новых блоков (для блокчейна)

# СМАРТ КОНТРАКТЫ

Пишутся на языке **Solidity** и компилируются в  
байт-код

```
pragma solidity ^0.4.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

# СМАРТ КОНТРАКТЫ

Исполнение кода смарт контрактов оплачивается  
*газом*

Неизрасходованный *газ* возвращается

Если *газ* закончился в процессе исполнения - код  
не исполняется, но *газ* не возвращается

# DAO

1. Ребята из The DAO провели ICO, собрали 12 млн ETH
2. В коде их смарт контракта была уязвимость...
3. Кто-то ей воспользовался и **легально** спер  
~\$50 млн

# DAO

Результат:

- Ethereum выполнили хардфорк, откатив блокчейн к прежнему состоянию
- Те, кто не согласился с хардфорком, образовали новую валюту Ethereum Classic

**НА ЭТОМ ВСЕ**