Примеры разработки #1

Смарт-Контракт Simple Billboard

Смарт-контракт Simple Billboard будет хранить рекламное объявление, которое будет доступно для всех, а редактирование объявления сможет редактировать только владелец контракта, то есть тот, кто его выполнил в сети Ethereum.

Для смарт-контракта мы создадим две переменные:

```
address owner;
string ads;
```

Переменная owner будет хранить адрес владельца смарт-контракта, а переменная ads текст рекламного объявления.

При создании смарт-контракта мы должна сразу указать нашего владельца. Это необходимо сделать в функции-конструкторе:

```
constructor() public {
    owner = msg.sender;}
```

При работе со смарт-контрактом можно использовать глобальную переменную msg Именно в этой переменной есть свойства, который передают при отправке сообщения в сеть Ethereum. Переменная msg содержит следующие свойства:

msg.sender - отправитель сообщения в блокчейн

msg.value - количество отправленного эфира в wei

msg.data - произвольные данные

При вызове конструктора смарт-контракта в переменную, в которой должен хранится адрес владельца, необходимо передать msg.sender. Так будет назначен владелец смарт-контракта. Очень важно не забывать добавлять владельца к смарт-контракту, чтобы иметь над ним полный контроль.

Для просмотра рекламного сообщения в сети Ethereum будет использоваться функцию getBillboard:

```
      function getBillboard() view public
returns(string) {
          return ads;}
```

Для функции указано ключевое слово view (для простого чтения данных из сети Etherum), public (функция видима всем) и возврат значения returns с типом string. В теле функции одна строка кода - возврат переменной контракта ads.

В смарт-контракте должна быть предусмотрена возможность редактирования сообщения, но с ограничением доступа - ей может воспользоваться только владелец контракта. Функция для редактирования текста рекламного сообщения будет следующей:

```
        function setBillboard(string _ads) public
{
            require(owner == msg.sender);
            ads = _ads;}
```

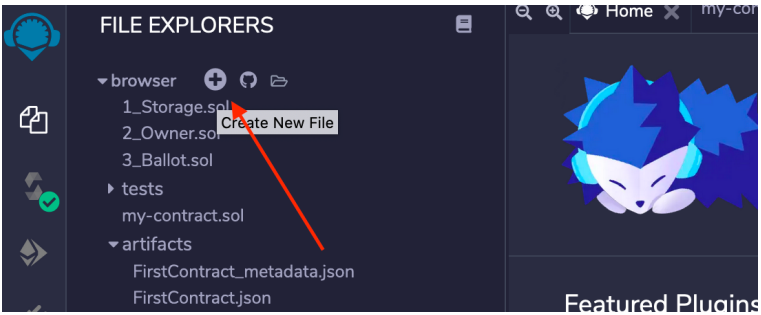
Функция setBillboard будет принимать строковую переменную, которая будет записана в глобальную переменную смарт-контракта ads.

Первая строка функции - проверит является ли отправитель сообщения владельцем, данного смарт-контракта. Все что будет описано ниже строчки с require возможно лишь в том случае, если условие, заключенное

в require выполнено. Например, при отправке сообщения через Metamask будет выведено предупреждение, что при выполнении транзакции будет ошибка.

Для проверки работы смарт-контракта его необходимо запустить в среде разработки Remix. Для этого нужно создать новый файл:

В окне необходимо ввести имя нового файла - пусть это будет SimpleBillboard.sol и нажимаем Ok.

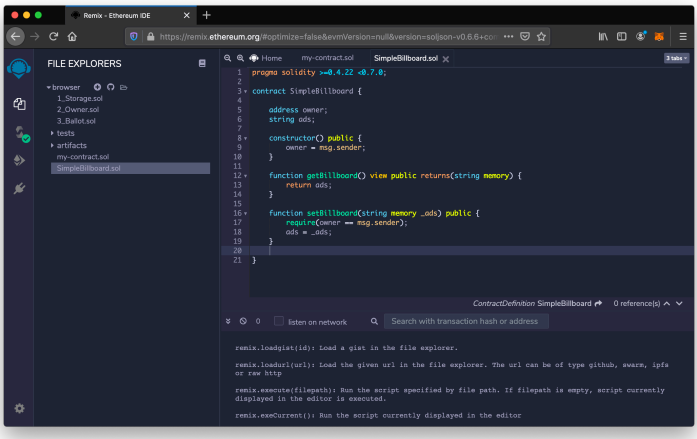


В вводим код смарт-контракта в редакторе.

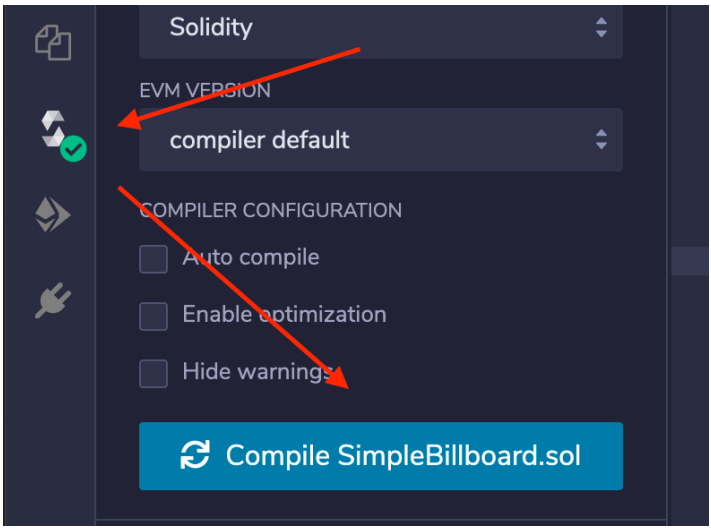
Полный код нашего смарт-контракта:

```
pragma solidity >=0.4.22 <0.7.0;

contract SimpleBillboard {

    address owner;
    string ads;
```

```
        constructor() public {
            owner = msg.sender;
        }

        function getBillboard() view public
returns(string memory) {
            return ads;
        }

        function setBillboard(string memory _ads)
public {
            require(owner == msg.sender);
            ads = _ads;
        }

    }
```



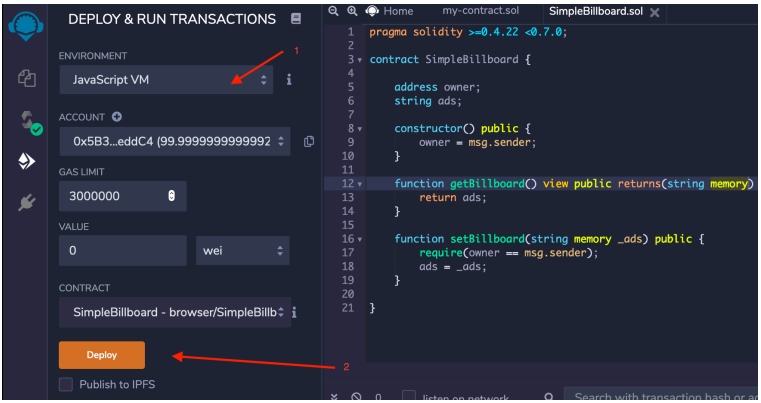
Теперь необходимо выполнить компиляцию контракта — это происходит на закладке Solidity compiler. Для компиляции нужно нажать кнопку **Compile SimpleBillboard.sol**.



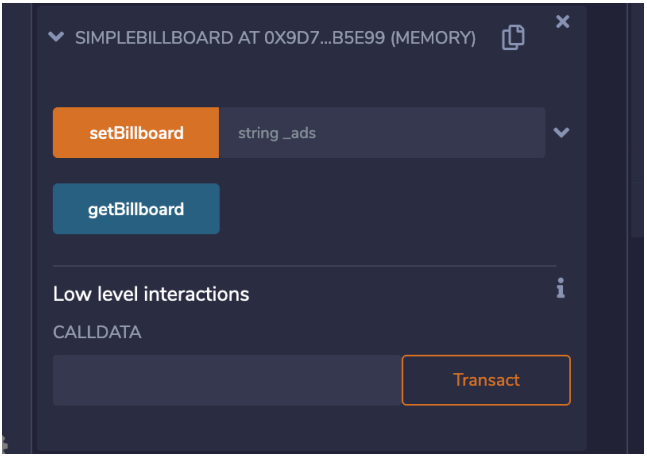
Если после нажатия кнопки не было ошибок, то можно уже деплоить смарт-контракт в сеть Ethereum. Для этого нужно перейти на закладку Deploy & run transactions.

Для тестирование лучше выбирать виртуальную машину JavaScript, а уже потом выполнять смарт-контракт в основной сети Ethereum. Поэтому на закладке

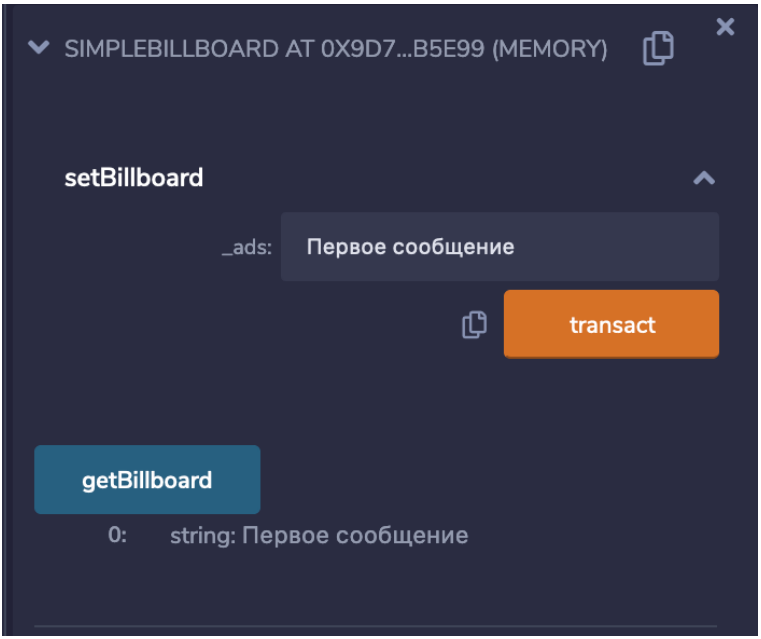
Run в поле Enviroment нужно переключиться на значение JavaScript VM (стрелка 1) и выполнить смарт-контракт с помощью кнопки Deploy (стрелка 2).



После выполнения смарт-контракта под кнопкой Deploy появятся функции смарт-контракта.



Для выполнения функции SetBillboard необходимо указать строку с текстом рекламного объявления и нажать кнопку setBillboard. А для просмотра текста рекламного объявления, не нужно указывать никаких переменных для передачи в функции - нужно выполнить функция getBillboard, которая вернет значение из сети Ethereum.



В нижней части экрана можно увидеть транзакции и вызовы функций смарт-контракта.