

# Правила лотереи



- Ограничение в 5 пользователей
- Пользователь должен заплатить 0,1 эфира, чтобы присоединиться к лотерее
- Один и тот же пользователь может присоединиться только один раз
- Владелец договора может присоединиться к лотерее
- Когда присоединяются 5 пользователей, выбирается победитель
- Победитель получает все деньги
- Новая лотерея начинается, когда выбран очередной победитель

### Функция присоединения к игре

```
function join() public payable {
    require(msg.value == 0.1 ether, "Must send 0.1 ether"); // проверка размера ставки
    require(joinedAlready(msg.sender) == false, "User already joined"); //проверка уникальности юзера
    participants[participantsCount] = msg.sender; //добавляем нового участника
    participantsCount++; //увеличиваем число участников
    if (participantsCount == 5) {
        selectWinner(); //если достигли 5 участников, то выбираем победителя
}
```

#### Проверка на уникальность юзера

```
function joinedAlready(address _participant) private view returns(bool)
    for(uint i = 0; i < participantsCount; i++) { //проходимся по всем
юзерам
      if (participants[i] == _participant) {
         return true; //нашли, выходим
    return false;//не нашли, выходим
```

### Выбор победителя

```
function selectWinner() private returns(address) {
    address winner = participants[randomNumber()]; //случайный
выбор пользователя
    winner.transfer(address(this).balance); //переводим все деньги
emy
    delete participants; //очищаем список участников
    participantsCount = 0; //обнуляем счетчик участников
    return winner; //возвращаем адрес победителя
```

### Полный код

```
contract Lottery {
                                                                                      return false;
  address[5] participants;
  uint8 participantsCount = 0;
                                                                                         function selectWinner() private returns(address) {
  uint randNonce = 0;
  function join() public payable {
                                                                                          address winner = participants[randomNumber()];
    require(msg.value == 0.1 ether, "Must send 0.1 ether");
                                                                                          winner.transfer(address(this).balance);
    require(joinedAlready(msg.sender) == false, "User already joined");
                                                                                          delete participants;
    participants[participantsCount] = msg.sender;
                                                                                          participantsCount = 0;
    participantsCount++;
                                                                                          return winner;
    if (participantsCount == 5) {
      selectWinner();
                                                                                          function randomNumber() private returns(uint) {
                                                                                          uint rand = uint(keccak256(abi.encodePacked(now, msg.sender, randNonce))) %
                                                                                     10;
                                                                                          randNonce++;
    function joinedAlready(address _participant) private view returns(bool) {
                                                                                          return rand;
    for(uint i = 0; i < participantsCount; i++) {</pre>
      if (participants[i] == _participant) {
        return true;
```

## Развертывание смарт-контракта

<pre>[vm] from:0xca3a733c to:Lottery.(constructor)</pre>	value:0 wei data:0x608d0029 logs:0 hash:0x9fbbefcd
status	0x1 Transaction mined and execution succeed
transaction hash	0x9fbbd1cd958911b6209b4ca767edcc06ec1ccd6e5d0d816c7b92433b06dbefcd
contract address	0x692a70d2e424a56d2c6c27aa97d1a86395877b3a 📋
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c 📋
to	Lottery.(constructor) 🖺
gas	3000000 gas
transaction cost	375108 gas 📋
execution cost	246160 gas 📋
hash	0x9fbbd1cd958911b6209b4ca767edcc06ec1ccd6e5d0d816c7b92433b06dbefcd
input	0x608d0029 🖺
decoded input	() 🗅
decoded output	- <u>ů</u>
logs	[] 🗂 🗂
value	0 wei

## Процесс игры

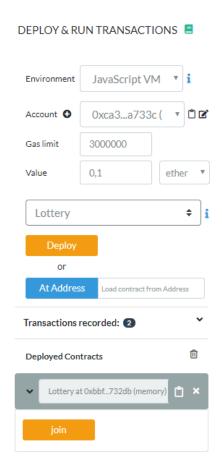
Аккаунты до вступления в лотерею

Oxca3...a733c (100 ether)
Ox147...c160c (100 ether)
Ox4b0...4d2db (100 ether)
Ox583...40225 (100 ether)
Oxdd8...92148 (100 ether)

Первый участник (автор контракта) вступил в игру

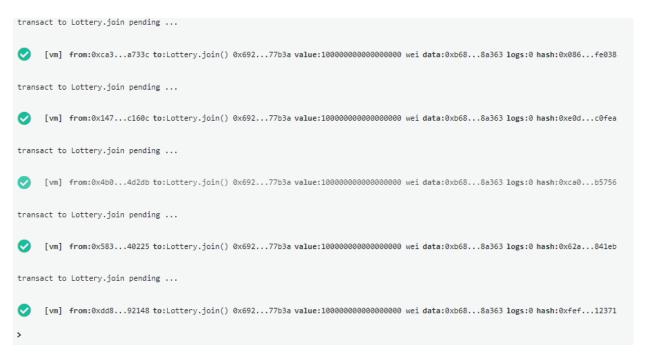
0xca3...a733c (99.8999999 0x147...c160c (100 ether) 0x4b0...4d2db (100 ether) 0x583...40225 (100 ether) 0xdd8...92148 (100 ether)

#### Транзакция со вступлением в игру



## Конец игры

#### Все игроки присоединились к игре



#### Второй игрок выиграл 0,5 эфира

Oxca3...a733c (99.899999999999540255 ether)

Ox147...c160c (100.399999999999951278 ether)

Ox4b0...4d2db (99.899999999999950671 ether)

Ox583...40225 (99.8999999999999950064 ether)

Oxdd8...92148 (99.8999999999999949656 ether)

