这是一个商品拍卖的智能合约。 有一个买家(获益的人) 和许多的竞价者, 竞价者对该商品进行竞价, 每次出价都会将相应的价值发送到智能合约中。 到拍卖结束的时候,

出价最高的人可以获得商品, 其余的人会拿回自己的钱。

据此,写出智能合约的 Solidity 代码:

```
pragma solidity ^0.4.24;
contract SimpleAuction{
   bytes32 public auction1Name;
   address public beneficiary;
   address public highestBidder;
   uint public auctionEnd;
   uint public highestOffer;11
   mapping(address=>uint) offers;
   bool ended;
   event HighestBidIncreased(address bidder,uint amount);
   event AuchtionEnded(address winner, uint amount);
   constructor(bytes32 _auctionlName, uint _biddingTime, address
_beneficiary) public{
       auctionlName = auctionlName;
       beneficiary = beneficiary;
       auctionEnd = block.timestamp + _biddingTime;
   function bid() public payable{
       require(block.timestamp <= auctionEnd);</pre>
       require(msg.value > highestOffer);
       if(highestBidder != 0){
           offers[highestBidder] += highestOffer;
       highestBidder = msg.sender;
       highestOffer = msg.value;
       HighestBidIncreased(msg.sender,msg.value);
   function withdraw() public view returns(bool){
       uint amount = offers[msg.sender];
       if (amount > 0){
          offers[msg.sender] = 0;
```

```
if (!msg.sender.send(amount)){
      offers[msg.sender] = amount;
      return false;
    }
    return true;
}

function auctionEnd() public{
    require(block.timestamp >= auctionEnd);
    require(!ended);
    ended = true;
    AuchtionEnded(highestBidder, highestOffer);
    beneficiary.transfer(highestOffer);
}
```

相关接口的解释:

function bid() public payable

竞价, 当需要对拍卖商品出价时调用

- 2. function withdraw() public view returns(bool)
 - 当某人的出价被其他人超过时,他可以选择调用该函数来撤销他之前的出价
- 3. function auctionEnd() public

当竞拍结束时,取消竞拍,将最高的出价交给拍卖品所有者

合约部署:

1. 通过 https://remix.ethereum.org, 获取智能合约的 ABI 和字节码: 字节码:

```
🥦 🖃 📵 ByteCode (~/Desktop) - gedit
      Open ▼ 💹 Save
                     ₽.
                        ← Undo → 🐰 🛅
ByteCode ×
{
      "linkReferences": {},
      "object":
"608060405234801561001057600080fd5b5060405160608061072b83398101806040528101908
      "opcodes": "PUSH1 0x80 PUSH1 0x40 MSTORE CALLVALUE DUP1 ISZERO PUSH2
0x10 JUMPI PUSH1 0x0 DUP1 REVERT JUMPDEST POP PUSH1 0x40 MLOAD PUSH1 0x60
DUP1 PUSH2 0x72B DUP4 CODECOPY DUP2 ADD DUP1 PUSH1 0x40 MSTORE DUP2 ADD
SWAP1 DUP1 DUP1 MLOAD SWAP1 PUSH1 0x20 ADD SWAP1 SWAP3 SWAP2 SWAP1 DUP1
MLOAD SWAP1 PUSH1 0x20 ADD SWAP1 SWAP3 SWAP2 SWAP1 DUP1 MLOAD SWAP1 PUSH1
0x20 ADD SWAP1 SWAP3 SWAP2 SWAP1 POP POP POP DUP3 PUSH1 0x0 DUP2 PUSH1 0x0
NOT AND SWAP1 SSTORE POP PUSH1 0x1 PUSH1 0x0 SWAP1 SLOAD SWAP1 PUSH2 0x100
0x1 PUSH1 0x0 PUSH2 0x100 EXP DUP2 SLOAD DUP2 PUSH20
0xffffffffffffffffffffffffffffffffff AND MUL OR SWAP1 SSTORE POP DUP2
TIMESTAMP ADD PUSH1 0x3 DUP2 SWAP1 SSTORE POP POP POP POP PUSH2 0x65B DUP1
PUSH2 0xD0 PUSH1 0x0 CODECOPY PUSH1 0x0 RETURN STOP PUSH1 0x80 PUSH1 0x40
MSTORE PUSH1 0x4 CALLDATASIZE LT PUSH2 0x83 JUMPI PUSH1 0x0 CALLDATALOAD
PUSH4 0xFFFFFFFF AND DUP1 PUSH4 0x1380C1F5 EQ PUSH2 0x88 JUMPI DUP1 PUSH4
0x1998AEEF EQ PUSH2 0xB3 JUMPI DUP1 PUSH4 0x2A24F46C EQ PUSH2 0xBD JUMPI
DUP1 PUSH4 0x38AF3EED EQ PUSH2 0xD4 JUMPI DUP1 PUSH4 0x3CCFD60B EQ PUSH2
Plain Text ▼ Tab Width: 8 ▼
                                               Ln 4, Col 80
```

ABI:

```
🔞 🖨 🗊 🛮 ABI (~/Desktop) - gedit
       Open ▼
                 Save
                            굔
ByteCode × ABI ×
        {
                 "constant": true,
                 "inputs": [],
                 "name": "highestOffer",
                 "outputs": [
                         {
                                  "name": ""
                                  "type": "uint256"
                         }
                 "payable": false,
"stateMutability": "view",
                 "type": "function"
        },
                 "constant": false,
                 "inputs": [],
                 "name": "bid",
                 "outputs": [],
                 "payable": true,
                 "stateMutability": "payable",
                 "tuno". "function
                                 Plain Text ▼ Tab Width: 8 ▼
                                                                Ln 1, Col 1
                                                                               INS
```

2. 通过 ABI 创建一个合约:

命令:

结果:

```
> myContract
  abi: [{
        constant: true,
        inputs: [],
name: "highestOffer",
outputs: [{...}],
payable: false,
        stateMutability: "view",
        type: "function"
        constant: false,
        inputs: [],
name: "bid",
outputs: [],
        payable: true,
        stateMutability: "payable",
        type: "function'
  }, {
        constant: false,
        inputs: [],
name: "auctionEnd",
        outputs: [],
payable: false,
stateMutability: "nonpayable",
        type: "function'
  }, {
        constant: true,
        inputs: [],
name: "beneficiary",
outputs: [{...}],
payable: false,
        stateMutability: "view",
        type: "function"
  }, {
        constant: false,
        inputs: [],
name: "withdraw",
outputs: [{...}],
        payable: false,
        stateMutability: "nonpayable",
        type: "function"
```

```
cype:
        anonymous: false,
        inputs: [{...}, {...}],
       name: "AuchtionEnded",
type: "event"
  }],
eth: {
accounts: ["0x78a851e5d57a3be6a72db2e4b4675f9b3f4bee03", "0x1566f3a17a1b89bd
99a88f1ffe5bdcca12b9f8cb", "0x668a62d20c07b48d1d98d6099ba60d1a230470fe", "0x91d9
1640ea6a55df669cac45a64a576263a0b780", "0xab5610050281c73be3f8415cf3a5a49ec6d7cf
f0", "0x2481a4867811e6d2d8b96aecef0839bfb464f563"],
    coinbase: "0x78a851e5d57a3be6a72db2e4b4675f9b3f4bee03",
       lll: function(),
       serpent: function(),
solidity: function()
     defaultAccount: undefined,
    defaultBlock: "latest",
     gasPrice:
     hashrate:
    mining: false,
     pendingTransactions: [{
          blockHash: null
          blockNumber: null,
          from: "0x78a851e5d57a3be6a72db2e4b4675f9b3f4bee03",
          gas:
          gasPrice:
          hash: "0xae395a9aed678955f1a8628192a1d9518fc6150805885e3536de5d288195351
          input: "0x1998aeef",
          nonce:
          r: "0x1155a05601f490324bd08c3cfc59858690b7fdf516c131e9492d49b75a64ebe9"
```

```
transactionIndex: 0,
      value:
 }],
 protocolVersion: "0x3f",
  syncing: false,
 call: function(),
 chainId: function(),
contract: function(abi),
 estimateGas: function(),
 filter: function(options, callback, filterCreationErrorCallback),
getAccounts: function(callback),
  getBalance: function(),
 getBlock: function(),
getBlockNumber: function(callback),
  getBlockTransactionCount: function(),
 getBlockUncleCount: function(),
 getCode: function(),
getCoinbase: function(callback),
 getCompilers: function(),
getGasPrice: function(callback),
getHashrate: function(callback),
 getMining: function(callback),
getPendingTransactions: function(callback),
 getProtocolVersion: function(callback),
 getRawTransaction: function(),
  getRawTransactionFromBlock: function(),
 getStorageAt: function(),
getSyncing: function(callback),
 getTransaction: function(),
 getTransactionCount: function(),
 getTransactionFromBlock: function(),
 getTransactionReceipt: function(),
 getUncle: function(),
 getWork: function(),
  iban: function(iban),
 icapNamereg: function(),
isSyncing: function(callback),
 namereg: function(),
 resend: function(),
  sendIBANTransaction: function(),
  sendRawTransaction: function(),
  sendTransaction: function(),
  sendTransaction: function(),
  sign: function(),
signTransaction: function(),
  submitTransaction: function(),
  submitWork: function()
at: function(address, callback),
getData: function(),
new: function()
```

3. 使用字节码预估手续费:

```
> web3.eth.estimateGas({data:bytecode})
527325
```

4. 部署合约:

```
> contractInstance = myContract.new({data: bytecode gas: 10000000, from: user1},
    function(e, contract){
        if(!e){
            console.log("Contract transaction send: Transaction Hash: "+c
        ontract.transactionHash+" waiting to be mined...");
        }else{
            console.log("Contract mined! Address: "+contract.address);
            console.log(contract);
            }
            console.log(contract);
        }
}
```

部署后输出如下:

```
abi: [{
    constant: true,
    inputs: [],
    name: "highestOffer",
outputs: [{...}],
payable: false,
    stateMutability: "view",
    type: "function
}, {
    constant: false,
    inputs: [],
    name: "bid
    outputs: [],
    payable: true,
    stateMutability: "payable",
    type: "function'
}, {
    constant: false,
    inputs: [],
name: "auctionEnd",
    outputs: [], payable: false,
    stateMutability: "nonpayable",
    type: "function
}, {
    constant: true,
    inputs: [],
    name: "beneficiary",
outputs: [{...}],
payable: false,
    stateMutability: "view",
    type: "function
}, {
    constant: false,
    inputs: [],
    name: "withdraw", outputs: [{...}],
    payable: false,
    stateMutability: "nonpayable",
    type: "function'
}, {
    constant: true,
```

合约开始等待挖矿。挖矿之后合约写入,:

```
INFO [11-27|02:57:04.724] Commit new mining work number=204 se alhash=fbb489...186bbd uncles=0 txs=0 gas=0 fees=0 elapsed=30.783ms Contract mined! Address: 0xf3eddd2391da2118c928b1de758073b13331afa9
```

可以查询到合约的地址:

> eth.getCode(contractInstance.address) 000000000000000900463ffffffff1680631380c1f5146100885780631998aeef146100b3578063 a57806391f901571461018d575b600080fd5b34801561009457600080fd5b5061009d6101e4565b6 080fd5b506100d2610374565b005b3480156100e057600080fd5b506100e96104b8565b604051808 fffffff16815260200191505060405180910390f35b34801561013757600080fd5b506101406104c e565b604051808215151515815260200191505060405180910390f35b34801561016657600080fd5 b5061016f610603565b60405180826000191660001916815260200191505060405180910390f35b3 4801561019957600080fd5b506101a2610609565b604051808273ffffffffffffffffffffffffffff 180910390f35b60045481565b60035442111515156101fb57600080fd5b6004543411151561020b fffffffff160217905550346004819055507ff4757a49b326036464bec6fe419a4ae38c8a02ce3e6 3fffffffffffffffffffffffffffffffffffff168152602001828152602001925050506040518 8427f8e12ab180b45dfe1dfd11a4ec97d0eb454d0b5a34b192cf9bbfcf67c1960026000905490610 26020018281526020019250505060405180910390a1600160009054906101000a900473fffffffff 8fc6004549081150290604051600060405180830381858888f193505050501580156104b5573d600 ffffffffffffffffffffffffffffffffff6815260200190815260200160002054905060008 ffffffffffffffffffffffffffffffff66108fc8290811502906040516000604051808303818 8704ccbc057e1c6644069c59ab27811622d793642dba478b804130029'

合约已经具有地址, 说明合约部署成功。