

Contract Name: CashCats

Deployed on SmartBCH Mainnet

Compiler Version: 0.4.21+commit.dfe3193c
Optimization Enabled: No with 200 runs
Other Settings: default evmVersion

Creator: 0x6db3fe7A14eE1F00660a8ebc3DA8ba4C4E684cCb
Contract Address: 0x265bD28d79400D55a1665707Fa14A72978FA6043
Creation TX hash: 0x6e0309980b73f6e8c3ceefe8fc652cfeac421dce5f7f4ab84bb32d4d57be4be6

Transaction:

<https://www.smartscan.cash/transaction/0x6e0309980b73f6e8c3ceefe8fc652cfeac421dce5f7f4ab84bb32d4d57be4be6>

Contract:

<https://www.smartscan.cash/address/0x265bd28d79400d55a1665707fa14a72978fa6043>

Solidity Contract Code:

Original contract:

<https://ropsten.etherscan.io/address/0x52cc47d359e9a030c273e31215bb58db90be2b64#code>

```
/**
 *Submitted for verification at Etherscan.io on 2018-04-16
 */

pragma solidity ^0.4.18;

/**
 * @title SafeMath
 * @dev Math operations with safety checks that throw on error
 */
library SafeMath {

    /**
     * @dev Multiplies two numbers, throws on overflow.
     */
    function mul(uint256 a, uint256 b) internal pure returns (uint256) {
        if (a == 0) {
            return 0;
        }
    }
}
```

```

    }
    uint256 c = a * b;
    assert(c / a == b);
    return c;
}

/**
 * @dev Integer division of two numbers, truncating the quotient.
 */
function div(uint256 a, uint256 b) internal pure returns (uint256) {
    // assert(b > 0); // Solidity automatically throws when dividing by 0
    uint256 c = a / b;
    // assert(a == b * c + a % b); // There is no case in which this doesn't hold
    return c;
}

/**
 * @dev Subtracts two numbers, throws on overflow (i.e. if subtrahend is greater than minue
 */
function sub(uint256 a, uint256 b) internal pure returns (uint256) {
    assert(b <= a);
    return a - b;
}

/**
 * @dev Adds two numbers, throws on overflow.
 */
function add(uint256 a, uint256 b) internal pure returns (uint256) {
    uint256 c = a + b;
    assert(c >= a);
    return c;
}
}
/**
 * @title ERC20Basic
 * @dev Simpler version of ERC20 interface
 * @dev see https://github.com/ethereum/EIPs/issues/179
 */
contract ERC20Basic {
    function totalSupply() public view returns (uint256);
    function balanceOf(address who) public view returns (uint256);
    function transfer(address to, uint256 value) public returns (bool);
    event Transfer(address indexed from, address indexed to, uint256 value);
}

/**
 * @title ERC20 interface
 * @dev see https://github.com/ethereum/EIPs/issues/20
 */
contract ERC20 is ERC20Basic {
    function allowance(address owner, address spender) public view returns (uint256);
    function transferFrom(address from, address to, uint256 value) public returns (bool);
    function approve(address spender, uint256 value) public returns (bool);
    event Approval(address indexed owner, address indexed spender, uint256 value);
}

```

```

/**
 * @title Basic token
 * @dev Basic version of StandardToken, with no allowances.
 */
contract BasicToken is ERC20Basic {
    using SafeMath for uint256;

    mapping(address => uint256) balances;

    uint256 totalSupply_;

    /**
     * @dev total number of tokens in existence
     */
    function totalSupply() public view returns (uint256) {
        return totalSupply_;
    }

    /**
     * @dev transfer token for a specified address
     * @param _to The address to transfer to.
     * @param _value The amount to be transferred.
     */
    function transfer(address _to, uint256 _value) public returns (bool) {
        require(_to != address(0));
        require(_value <= balances[msg.sender]);

        // SafeMath.sub will throw if there is not enough balance.
        balances[msg.sender] = balances[msg.sender].sub(_value);
        balances[_to] = balances[_to].add(_value);
        Transfer(msg.sender, _to, _value);
        return true;
    }

    /**
     * @dev Gets the balance of the specified address.
     * @param _owner The address to query the the balance of.
     * @return An uint256 representing the amount owned by the passed address.
     */
    function balanceOf(address _owner) public view returns (uint256 balance) {
        return balances[_owner];
    }
}

contract tokenRecipient {function receiveApproval(address _from, uint256 _value, address _toke

/**
 * @title Standard ERC20 token
 *
 * @dev Implementation of the basic standard token.
 * @dev https://github.com/ethereum/EIPs/issues/20
 * @dev Based on code by FirstBlood: https://github.com/Firstbloodio/token/blob/master/smart_c

```

```

*/
contract StandardToken is ERC20, BasicToken {

    /* Public variables of the token */
    string public standard = 'ERC20';

    string public name;

    string public symbol;

    uint8 public decimals;

    uint256 public totalSupply;

    address public owner;

    mapping (address => mapping (address => uint256)) internal allowed;

    function StandardToken(
        uint256 initialSupply,
        string tokenName,
        uint8 decimalUnits,
        string tokenSymbol
    ) {
        balances[msg.sender] = initialSupply;
        // Give the creator all initial tokens
        totalSupply = initialSupply;
        // Update total supply
        name = tokenName;
        // Set the name for display purposes
        symbol = tokenSymbol;
        // Set the symbol for display purposes
        decimals = decimalUnits;
        // Amount of decimals for display purposes

        owner=msg.sender;
    }

    modifier onlyOwner {
        if (msg.sender != owner) throw;
        _;
    }

    /**
     * @dev Transfer tokens from one address to another
     * @param _from address The address which you want to send tokens from
     * @param _to address The address which you want to transfer to
     * @param _value uint256 the amount of tokens to be transferred
     */
    function transferFrom(address _from, address _to, uint256 _value) public returns (bool) {
        require(_to != address(0));
        require(_value <= balances[_from]);
        require(_value <= allowed[_from][msg.sender]);

        balances[_from] = balances[_from].sub(_value);
    }
}

```

```

        balances[_to] = balances[_to].add(_value);
        allowed[_from][msg.sender] = allowed[_from][msg.sender].sub(_value);
        Transfer(_from, _to, _value);
        return true;
    }

/**
 * @dev Approve the passed address to spend the specified amount of tokens on behalf of ms
 *
 * Beware that changing an allowance with this method brings the risk that someone may use
 * and the new allowance by unfortunate transaction ordering. One possible solution to mit
 * race condition is to first reduce the spender's allowance to 0 and set the desired valu
 * https://github.com/ethereum/EIPs/issues/20#issuecomment-263524729
 * @param _spender The address which will spend the funds.
 * @param _value The amount of tokens to be spent.
 */
function approve(address _spender, uint256 _value) public returns (bool) {
    allowed[msg.sender][_spender] = _value;
    Approval(msg.sender, _spender, _value);
    return true;
}

function multiApprove(address[] _spender, uint256[] _value) public returns (bool){
    require(_spender.length == _value.length);
    for(uint i=0;i<=_spender.length;i++){
        allowed[msg.sender][_spender[i]] = _value[i];
        Approval(msg.sender, _spender[i], _value[i]);
    }
    return true;
}

/**
 * @dev Function to check the amount of tokens that an owner allowed to a spender.
 * @param _owner address The address which owns the funds.
 * @param _spender address The address which will spend the funds.
 * @return A uint256 specifying the amount of tokens still available for the spender.
 */
function allowance(address _owner, address _spender) public view returns (uint256) {
    return allowed[_owner][_spender];
}

/**
 * @dev Increase the amount of tokens that an owner allowed to a spender.
 *
 * approve should be called when allowed[_spender] == 0. To increment
 * allowed value is better to use this function to avoid 2 calls (and wait until
 * the first transaction is mined)
 * From MonolithDAO Token.sol
 * @param _spender The address which will spend the funds.
 * @param _addedValue The amount of tokens to increase the allowance by.
 */
function increaseApproval(address _spender, uint _addedValue) public returns (bool) {
    allowed[msg.sender][_spender] = allowed[msg.sender][_spender].add(_addedValue);
    Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
    return true;
}

```

```

function multiIncreaseApproval(address[] _spender, uint[] _addedValue) public returns (bool) {
    require(_spender.length == _addedValue.length);
    for(uint i=0;i<=_spender.length;i++){
        allowed[msg.sender][_spender[i]] = allowed[msg.sender][_spender[i]].add(_addedValue[i]);
        Approval(msg.sender, _spender[i], allowed[msg.sender][_spender[i]]);
    }
    return true;
}
/**
 * @dev Decrease the amount of tokens that an owner allowed to a spender.
 *
 * approve should be called when allowed[_spender] == 0. To decrement
 * allowed value is better to use this function to avoid 2 calls (and wait until
 * the first transaction is mined)
 * From MonolithDAO Token.sol
 * @param _spender The address which will spend the funds.
 * @param _subtractedValue The amount of tokens to decrease the allowance by.
 */
function decreaseApproval(address _spender, uint _subtractedValue) public returns (bool) {
    uint oldValue = allowed[msg.sender][_spender];
    if (_subtractedValue > oldValue) {
        allowed[msg.sender][_spender] = 0;
    } else {
        allowed[msg.sender][_spender] = oldValue.sub(_subtractedValue);
    }
    Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
    return true;
}

function multiDecreaseApproval(address[] _spender, uint[] _subtractedValue) public returns (bool) {
    require(_spender.length == _subtractedValue.length);
    for(uint i=0;i<=_spender.length;i++){
        uint oldValue = allowed[msg.sender][_spender[i]];
        if (_subtractedValue[i] > oldValue) {
            allowed[msg.sender][_spender[i]] = 0;
        } else {
            allowed[msg.sender][_spender[i]] = oldValue.sub(_subtractedValue[i]);
        }
        Approval(msg.sender, _spender[i], allowed[msg.sender][_spender[i]]);
    }
    return true;
}

/* Approve and then communicate the approved contract in a single tx */
function approveAndCall(address _spender, uint256 _value, bytes _extraData)
returns (bool success) {
    tokenRecipient spender = tokenRecipient(_spender);
    if (approve(_spender, _value)) {
        spender.receiveApproval(msg.sender, _value, this, _extraData);
        return true;
    }
}
}

```


104ac5760080fd5b6104e1600480803573fff16906020019091908035
9060200190919050506112b4565b604051808215151515815260200191505060405180910390f35b341561050657600
080fd5b610532600480803573fff16906020019091905050611545565b
6040518082815260200191505060405180910390f35b341561055357600080fd5b6105e060048080359060200190820
18035906020019080806020026020016040519081016040528093929190818152602001838360200280828437820191
505050505091908035906020019082018035906020019080806020026020016040519081016040528093929190818
15260200183836020028082843782019150505050509190505061158d565b60405180821515151581526020019150
5060405180910390f35b341561060557600080fd5b61060d6118ee565b604051808273ffffffffffffffffffffffffffff
ffffffffffffffff1673ff16815260200191505060405180910390f35b
341561065a57600080fd5b610662611914565b604051808060200182810382528381815181526020019150805190602
0019080838360005b838110156106a2578082015181840152602081019050610687565b50505050905090810190601f
1680156106cf578082038051600183602003610100a031916815260200191505b509250505060405180910390f35b3
4156106e857600080fd5b61071d600480803573fff1690602001909190
80359060200190919050506119b2565b604051808215151515815260200191505060405180910390f35b34156107425
7600080fd5b6107ba600480803573fff16906020019091908035906020
01909190803590602001908201803590602001908080601f01602080910402602001604051908101604052809392919
0818152602001838380828437820191505050505091905050611bd1565b6040518082151515158152602001915050
60405180910390f35b34156107df57600080fd5b610814600480803573fff
fff16906020019091908035906020019091905050611d4b565b60405180821515151581526020019150506040518091
0390f35b341561083957600080fd5b610884600480803573fff1690602
001909190803573fff16906020019091905050611f47565b6040518082
815260200191505060405180910390f35b600080825184511415156108ad57600080fd5b600090505b8351811115156
10b2c5761097983828151811015156108cd57fe5b90602001906020020151600860003373fffffffffffffffffffff
ffffffffffffffff1673ff168152602001908152602001600020600
0878581518110151561092657fe5b9060200190602002015173fff1673
ff16815260200190815260200160002054611fce90919063fffffffff1
6565b600860003373fff1673fffffffffffffffffffffffffffffffffffff
ffffffff168152602001908152602001600020600086848151811015156109c857fe5b9060200190602002015173fffff
ffffffffffffffffffffffffffffffff1673fffffffffffffffffffffffffffffffffffff1681526020019081
52602001600020819055508381815181101515610a1e57fe5b9060200190602002015173fffffffffffffffffffff
ffffffff163373fffffffffffffffffffffffffffffffffffff167f8c5be1e5ebec7d5bd14f71427d1e
84f3dd0314c0f7b2291e5b200ac8c7c3b925600860003373fffffffffffffffffffff1673fff
fffffffffffffffffffff16815260200190815260200160002060008886815181101515610ac557
fe5b9060200190602002015173fffffffffffffffffffff1673fffffffffffffffffffff
ffffffff168152602001908152602001600020546040518082815260200191505060405180910390a3808060
01019150506108b2565b600191505092915050565b60038054600181600116156101000203166002900480601f01602
08091040260200160405190810160405280929190818152602001828054600181600116156101000203166002900480
15610bcd5780601f10610ba257610100808354040283529160200191610bcd565b820191906000526020600020905b8
15481529060010190602001808311610bb057829003601f168201915b5050505081565b600081600860003373ffff
fffffffffffff1673fffffffffffff168152602001908
15260200160002060008573fffffffffffff1673fffff
fffff168152602001908152602001600020819055508273fffff
63373fffff167f8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b229

1e5b200ac8c7c3b925846040518082815260200191505060405180910390a36001905092915050565b60065481565b6
0008073ff168373ff16
14151515610d0a57600080fd5b6000808573ff1673ffffffffffffffffffff
ffffffffffffffffffffffffffffffff168152602001908152602001600020548211151515610d5757600080fd5b6008600085
73ffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffff168152602
00190815260200160002060003373ffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffff
ffffffffffffffff168152602001908152602001600020548211151515610de257600080fd5b610e3382600080877
3ffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffff1681526020
0190815260200160002054611fec90919063ffffffff16565b6000808673ffffffffffffffffffffffffffff
ffff1673ffffffffffffffffffffffffffffffff16815260200190815260200160002081905550610ec682
6000808673ffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffff1
6815260200190815260200160002054611fce90919063ffffffff16565b6000808573ffffffffffffffffffff
ffffffff1673ffffffffffffffffffffffffffff1681526020019081526020016000208190555
0610f9782600860008773ffffffffffffffffffff1673ffffffffffffffffffff
ffffffff16815260200190815260200160002060003373ffffffffffffffffffff1673fff
ffffffffffffffffffff16815260200190815260200160002054611fec90919063ffffffff1656
5b600860008673ffffffffffffffffffff1673ffffffffffffffffffff
fff16815260200190815260200160002060003373ffffffffffffffffffff1673ffff
ffffffffffff168152602001908152602001600020819055508273ffffffffffff
ffffffff168473ffffffffffff167fddf252ad1be2c89b69c2b068fc37
8daa952ba7f163c4a11628f55a4df523b3ef846040518082815260200191505060405180910390a3600190509392505
050565b600560009054906101000a900460ff1681565b600080825184511415156110ad57600080fd5b600090505b83
518111151561120b5782818151811015156110ca57fe5b90602001906020020151600860003373ffffffffffff
ffffffff1673ffffffffffff1681526020019081526020016000
206000868481518110151561112357fe5b9060200190602002015173ffffffffffff
f1673ffffffffffff16815260200190815260200160002081905550838181518110
151561117957fe5b9060200190602002015173ffffffffffff163373ffff
ffff167f8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b9
2585848151811015156111df57fe5b906020019060200201516040518082815260200191505060405180910390a3808
06001019150506110b2565b600191505092915050565b60028054600181600116156101000203166002900480601f01
60208091040260200160405190810160405280929190818152602001828054600181600116156101000203166002900
480156112ac5780601f10611281576101008083540402835291602001916112ac565b82019190600052602060002090
5b81548152906001019060200180831161128f57829003601f168201915b5050505081565b600080600860003373f
ffffffffffff1673ffffffffffff168152602001
90815260200160002060008573ffffffffffff1673ffffffffffff
ffffffff168152602001908152602001600020549050808311156113c5576000600860003373ffffffffffff
ffffffff1673ffffffffffff16815260200190815260200
160002060008673ffffffffffff1673ffffffffffff
ffff16815260200190815260200160002081905550611459565b6113d88382611fec90919063ffffffff16565b60086
0003373ffffffffffff1673ffffffffffff1681
5260200190815260200160002060008673ffffffffffff1673ffff
ffff168152602001908152602001600020819055505b8373ffff
ffff163373ffff167f8c5be1e5ebec7d5bd14f71427d1e84f3d

d0314c0f7b2291e5b200ac8c7c3b925600860003373fff1673fffffffff
ff16815260200190815260200160002060008873fffffffffffffffff
ffffffffffffffff1673ffffffffffffffffffffffffffffffff168152602001908152602001600020546040
518082815260200191505060405180910390a3600191505092915050565b60008060008373fffff
ffffffffffffffff1673ffffffffffffffffffffffff16815260200190815260200160002054
9050919050565b6000806000835185511415156115a257600080fd5b600091505b8451821115156118e257600860003
373ffffffffffffffff1673ffffffffffffffff16815260
20019081526020016000206000868481518110151561160057fe5b9060200190602002015173fffff
ffffffff1673ffffffffffffffff168152602001908152602001600020
54905080848381518110151561165657fe5b90602001906020020151115611704576000600860003373fffff
ffffffff1673ffffffffffffffff1681526020019081526020
01600020600087858151811015156116b757fe5b9060200190602002015173fffff
ffffffff1673ffffffffffffffff168152602001908152602001600020819055506117c6
565b61172e848381518110151561171557fe5b9060200190602002015182611fec90919063fffff
03373ffffffffffffffff1673ffffffffffffffff168152
6020019081526020016000206000878581518110151561177d57fe5b9060200190602002015173fffff
ffffffff1673ffffffffffffffff1681526020019081526020016000
20819055505b84828151811015156117d457fe5b9060200190602002015173fffff
ffffffff163373ffffffffffffffff167f8c5be1e5ebec7d5bd14f71427d1e84f3dd0314
c0f7b2291e5b200ac8c7c3b925600860003373fffff
ffffffff1673ffffffffffffffff1681526020019081526020016000206000898781518110151561187b57fe5b906020
0190602002015173fffff
ffffffff1673ffffffffffffffff168152602001908152602001600020546040518082815260200191505060405180910390a38180600101925050
6115a7565b60019250505092915050565b600760009054906101000a900473fffff
ffffffff1681565b60048054600181600116156101000203166002900480601f01602080910402602001604051908101
60405280929190818152602001828054600181600116156101000203166002900480156119aa5780601f1061197f576
101008083540402835291602001916119aa565b820191906000526020600020905b8154815290600101906020018083
1161198d57829003601f168201915b5050505081565b60008073fffff
68373ffffffffffffffff16141515156119ef57600080fd5b6000803373fffff
ffffffff1673ffffffffffffffff16815260200190815260200
1600020548211151515611a3c57600080fd5b611a8d826000803373fffff
1673ffffffffffffffff16815260200190815260200160002054611fec90919063fffff
fff16565b6000803373ffffffffffffffff1673ffffffffffffffff
ffffffff16815260200190815260200160002081905550611b20826000808673fffff
ffffffff1673ffffffffffffffff16815260200190815260200160002054611fce9091
9063fffff16565b6000808573ffffffffffffffff1673fffffffffffff
ffffffff168152602001908152602001600020819055508273fffff
fff163373ffffffffffffffff167fddf252ad1be2c89b69c2b068fc378daa952ba7f16
3c4a11628f55a4df523b3ef846040518082815260200191505060405180910390a36001905092915050565b60008084
9050611be18585610bd5565b15611d42578073fffff
0876040518563ffffffff167c010028152600401
808573ffffffffffffffff1673fffffffffffff
26020018481526020018373fffffffffffff1673fffff

Encoded:

```
Arg [0] : 000000000000000000000000000000000000000000000000000000000000000000174876e800
Arg [1] : 000000000000000000000000000000000000000000000000000000000000000000080
Arg [2] : 00000000000000000000000000000000000000000000000000000000000000000002
Arg [3] : 000000000000000000000000000000000000000000000000000000000000000000c0
Arg [4] : 00000000000000000000000000000000000000000000000000000000000000000008
Arg [5] : 4361736843617473000000000000000000000000000000000000000000000000000000
Arg [6] : 00000000000000000000000000000000000000000000000000000000000000000005
Arg [7] : 24434154530000000000000000000000000000000000000000000000000000000000
```

ABI

```
[
  {
    "constant": false,
    "inputs": [
      {
        "name": "_spender",
        "type": "address[]"
      },
      {
        "name": "_addedValue",
        "type": "uint256[]"
      }
    ],
    "name": "multiIncreaseApproval",
    "outputs": [
      {
        "name": "",
        "type": "bool"
      }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "constant": true,
    "inputs": [],
    "name": "name",
    "outputs": [
      {
        "name": "",
        "type": "string"
      }
    ],
    "payable": false,
```

```

    "stateMutability": "view",
    "type": "function"
  },
  {
    "constant": false,
    "inputs": [
      {
        "name": "_spender",
        "type": "address"
      },
      {
        "name": "_value",
        "type": "uint256"
      }
    ],
    "name": "approve",
    "outputs": [
      {
        "name": "",
        "type": "bool"
      }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "constant": true,
    "inputs": [],
    "name": "totalSupply",
    "outputs": [
      {
        "name": "",
        "type": "uint256"
      }
    ],
    "payable": false,
    "stateMutability": "view",
    "type": "function"
  },
  {
    "constant": false,
    "inputs": [
      {
        "name": "_from",
        "type": "address"
      },
      {
        "name": "_to",
        "type": "address"
      },
      {
        "name": "_value",
        "type": "uint256"
      }
    ]
  }

```

```

    ],
    "name": "transferFrom",
    "outputs": [
        {
            "name": "",
            "type": "bool"
        }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
},
{
    "constant": true,
    "inputs": [],
    "name": "decimals",
    "outputs": [
        {
            "name": "",
            "type": "uint8"
        }
    ],
    "payable": false,
    "stateMutability": "view",
    "type": "function"
},
{
    "constant": false,
    "inputs": [
        {
            "name": "_spender",
            "type": "address[]"
        },
        {
            "name": "_value",
            "type": "uint256[]"
        }
    ],
    "name": "multiApprove",
    "outputs": [
        {
            "name": "",
            "type": "bool"
        }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
},
{
    "constant": true,
    "inputs": [],
    "name": "standard",
    "outputs": [
        {

```

```

        "name": "",
        "type": "string"
    }
],
"payable": false,
"stateMutability": "view",
"type": "function"
},
{
    "constant": false,
    "inputs": [
        {
            "name": "_spender",
            "type": "address"
        },
        {
            "name": "_subtractedValue",
            "type": "uint256"
        }
    ],
    "name": "decreaseApproval",
    "outputs": [
        {
            "name": "",
            "type": "bool"
        }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
},
{
    "constant": true,
    "inputs": [
        {
            "name": "_owner",
            "type": "address"
        }
    ],
    "name": "balanceOf",
    "outputs": [
        {
            "name": "balance",
            "type": "uint256"
        }
    ],
    "payable": false,
    "stateMutability": "view",
    "type": "function"
},
{
    "constant": false,
    "inputs": [
        {
            "name": "_spender",

```

```

        "type": "address[]"
    },
    {
        "name": "_subtractedValue",
        "type": "uint256[]"
    }
],
"name": "multiDecreaseApproval",
"outputs": [
    {
        "name": "",
        "type": "bool"
    }
],
"payable": false,
"stateMutability": "nonpayable",
"type": "function"
},
{
    "constant": true,
    "inputs": [],
    "name": "owner",
    "outputs": [
        {
            "name": "",
            "type": "address"
        }
    ],
    "payable": false,
    "stateMutability": "view",
    "type": "function"
},
{
    "constant": true,
    "inputs": [],
    "name": "symbol",
    "outputs": [
        {
            "name": "",
            "type": "string"
        }
    ],
    "payable": false,
    "stateMutability": "view",
    "type": "function"
},
{
    "constant": false,
    "inputs": [
        {
            "name": "_to",
            "type": "address"
        },
        {
            "name": "_value",

```



```

        "type": "uint256"
    }
],
"name": "transfer",
"outputs": [
    {
        "name": "",
        "type": "bool"
    }
],
"payable": false,
"stateMutability": "nonpayable",
"type": "function"
},
{
    "constant": false,
    "inputs": [
        {
            "name": "_spender",
            "type": "address"
        },
        {
            "name": "_value",
            "type": "uint256"
        },
        {
            "name": "_extraData",
            "type": "bytes"
        }
    ],
    "name": "approveAndCall",
    "outputs": [
        {
            "name": "success",
            "type": "bool"
        }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
},
{
    "constant": false,
    "inputs": [
        {
            "name": "_spender",
            "type": "address"
        },
        {
            "name": "_addedValue",
            "type": "uint256"
        }
    ],
    "name": "increaseApproval",
    "outputs": [

```

```

        {
            "name": "",
            "type": "bool"
        }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
},
{
    "constant": true,
    "inputs": [
        {
            "name": "_owner",
            "type": "address"
        },
        {
            "name": "_spender",
            "type": "address"
        }
    ],
    "name": "allowance",
    "outputs": [
        {
            "name": "",
            "type": "uint256"
        }
    ],
    "payable": false,
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "name": "initialSupply",
            "type": "uint256"
        },
        {
            "name": "tokenName",
            "type": "string"
        },
        {
            "name": "decimalUnits",
            "type": "uint8"
        },
        {
            "name": "tokenSymbol",
            "type": "string"
        }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "constructor"
},

```

```

{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "name": "owner",
      "type": "address"
    },
    {
      "indexed": true,
      "name": "spender",
      "type": "address"
    },
    {
      "indexed": false,
      "name": "value",
      "type": "uint256"
    }
  ],
  "name": "Approval",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "name": "from",
      "type": "address"
    },
    {
      "indexed": true,
      "name": "to",
      "type": "address"
    },
    {
      "indexed": false,
      "name": "value",
      "type": "uint256"
    }
  ],
  "name": "Transfer",
  "type": "event"
}
]

```

SourceMap

```

3869:6512:0:-;;;3960:32;;;;;:::i;::-;4212:569;;;;;
;;;;;4387:13;4364:8;:20;4373:10;4364:20;;;;;:36;;;4471:13

```

;4457:11;:27;;;4532:9;4525:4;:16;;;::i;::-;4605:11;4596:6;:20;;;::i;::-
;;4684:12;4673:8;:23;;;4764:10;4758:5;:16;;;4212:569;;;3869:65
12;;;::i;::-;::o;::-;::o;::-;