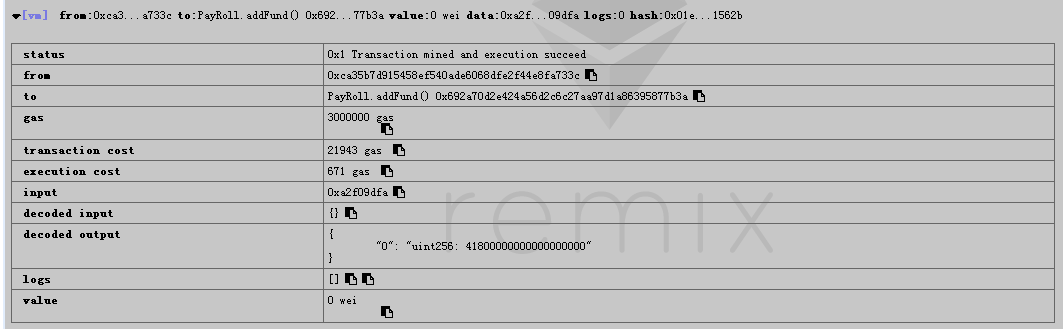
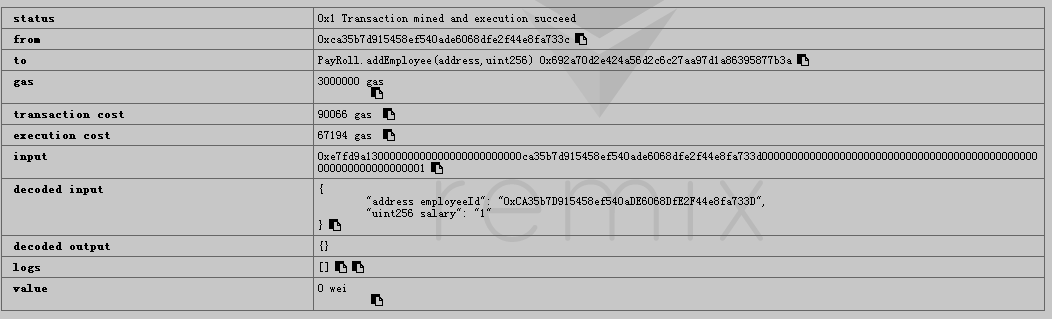
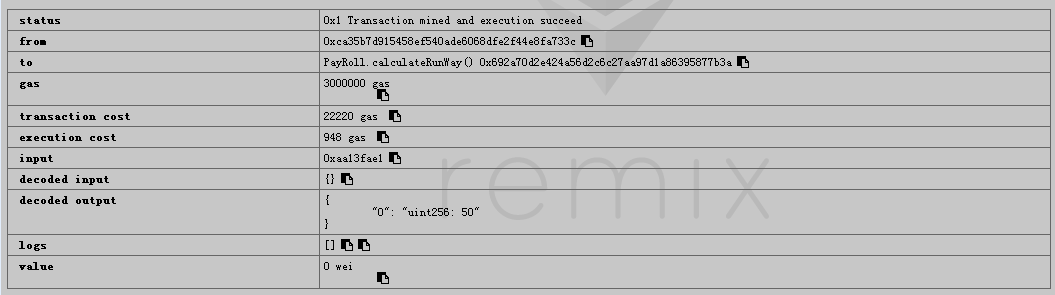
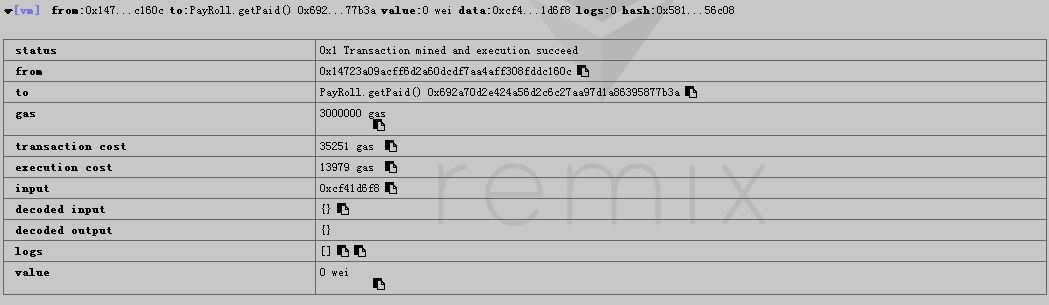
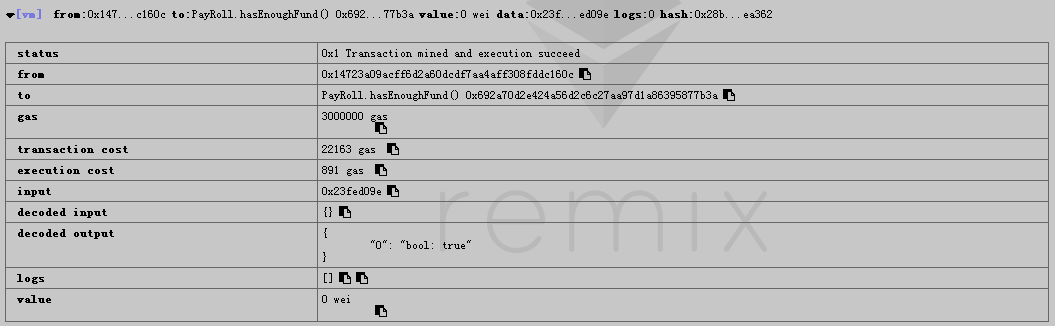
1. 函数调用截图如下：





课程代码如下：

pragma solidity ^0.4.14;

import './SafeMath.sol';

import './Ownable.sol';

contract PayRoll is Ownable{

using SafeMath for uint;

struct Employee {

address id;

uint salary;

uint lastPayDay;

}

address owner;

uint totalSalary = 0;

uint constant payDuration = 10 seconds;

mapping(address => Employee) employees;

function PayRoll() payable {

owner = msg.sender;

}

modifier employeeExist(address employeeId) {

var employee = employees[employeeId];

assert(employee.id != 0x0);

\_;

}

modifier employeeNotExist(address employeeId) {

var employee = employees[employeeId];

assert(employee.id == 0x0);

\_;

}

function addFund() payable returns (uint) {

return this.balance;

}

function \_patialPaid(Employee employee) private {

uint payment = employee.salary.mul((now - employee.lastPayDay)).div(payDuration);

employee.id.transfer(payment);

}

function addEmployee(address employeeId, uint salary) onlyOwner {

var employee = employees[employeeId];

assert(employee.id==0x0);

employees[employeeId] = Employee(employeeId,salary\* 1 ether,now);

totalSalary = totalSalary.add(employees[employeeId].salary);

}

function removeEmployee(address employeeId) onlyOwner employeeExist(employeeId) {

var employee = employees[employeeId];

\_patialPaid(employee);

totalSalary -= employees[employeeId].salary;

delete employees[employeeId];

}

function changePaymentAddress(address oldAddr, address newAddr) onlyOwner employeeExist(oldAddr) employeeNotExist(newAddr) {

var employee = employees[oldAddr];

\_patialPaid(employee);

totalSalary -= employees[oldAddr].salary;

employees[newAddr] = Employee(newAddr,employee.salary,now);

delete employees[oldAddr];

totalSalary = totalSalary.add(employees[newAddr].salary);

}

function updateEmployee(address employeeId, uint salary) onlyOwner employeeExist(employeeId) {

var employee = employees[employeeId];

\_patialPaid(employee);

totalSalary -= employees[employeeId].salary;

employees[employeeId].salary = salary \* 1 ether;

employees[employeeId].lastPayDay = now;

totalSalary += employees[employeeId].salary;

}

function calculateRunWay() returns (uint) {

return this.balance/totalSalary;

}

function hasEnoughFund() returns (bool) {

return calculateRunWay() > 0;

}

function getPaid() employeeExist(msg.sender) {

var employee = employees[msg.sender];

uint nextPayDay = employee.lastPayDay + payDuration;

assert(nextPayDay < now);

employee.lastPayDay = nextPayDay;

employee.id.transfer(employee.salary);

}

}

1. changePaymentAddress里增加一个modifier employeeNotExist

modifier employeeNotExist(address employeeId) {

var employee = employees[employeeId];

assert(employee.id == 0x0);

\_;

}  
 function changePaymentAddress(address oldAddr, address newAddr) onlyOwner employeeExist(oldAddr) employeeNotExist(newAddr) {

var employee = employees[oldAddr];

\_patialPaid(employee);

totalSalary -= employees[oldAddr].salary;

employees[newAddr] = Employee(newAddr,employee.salary,now);

delete employees[oldAddr];

totalSalary = totalSalary.add(employees[newAddr].salary);

}