04-第四节 存款合约

实验目的

- 掌握多个合约间的调用；
- 掌握 modifier/constant/immutable 的用法；
- 掌握如何实现一个活期存款合约；

实验环境

- Chrome/Microsoft Edge等浏览器；
- Remix IDE：https://remix.ethereum.org/；

实验内容

需求

1. 充值时的币种使用WETH，领取奖励和提取本金时的币种为ETH；
2. 每充值5个WETH，经过一个区块高度后，可以领取1个ETH的利息；
3. 在构造函数中设置活动起止区块高度，只有在起止区块高度内的充值才可以计算利息。起止区块高度之间的区块也称为**活动期**。活动期开始前的区块以及活动期结束后的区块，不计算利息。
4. 编写modifier，使得充值操作只能在活动期进行；
5. 在活动期内，用户可以随时充值、提取利息、提取本金；

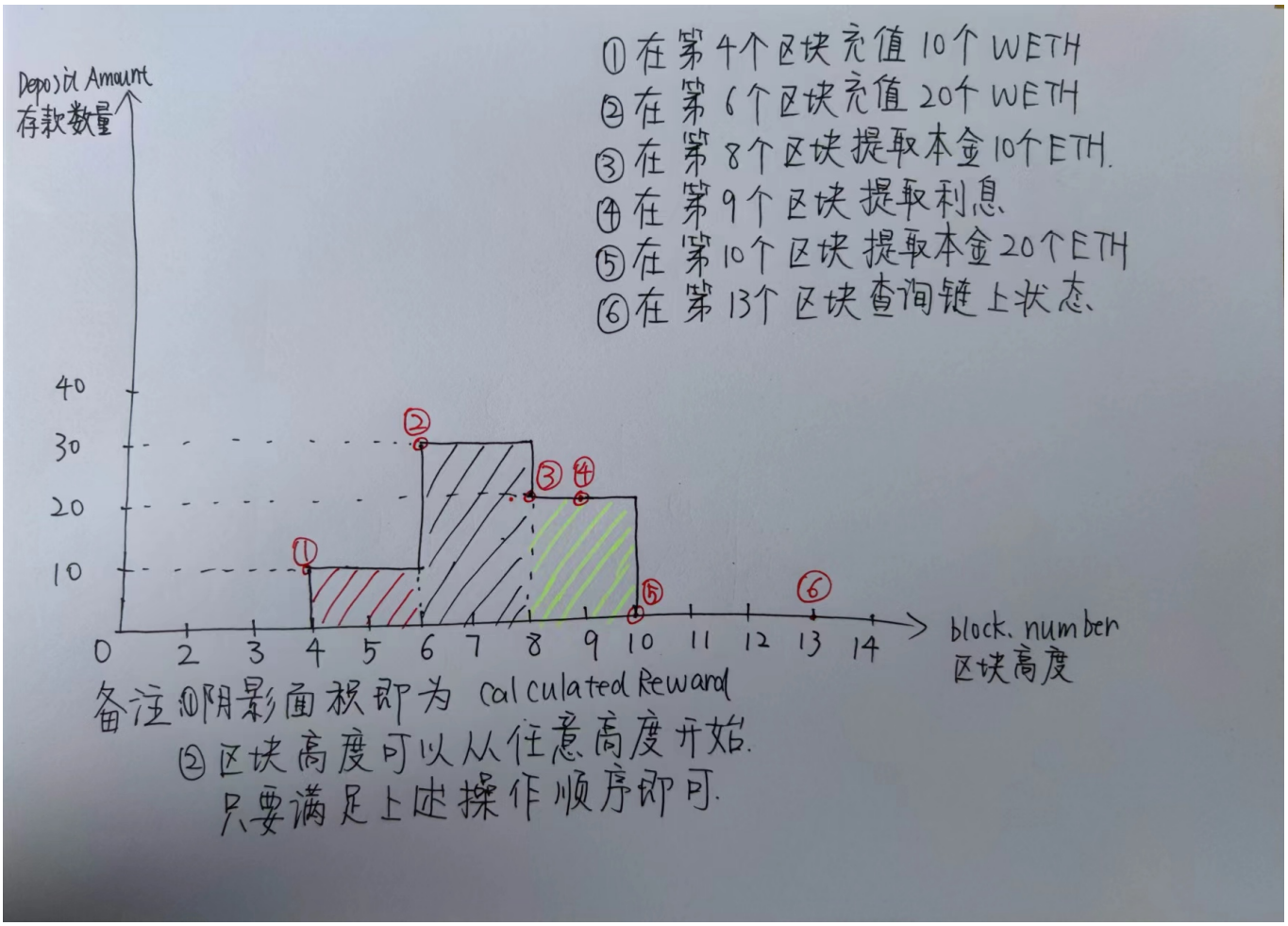
6. 利息不参与复投；

参考算法

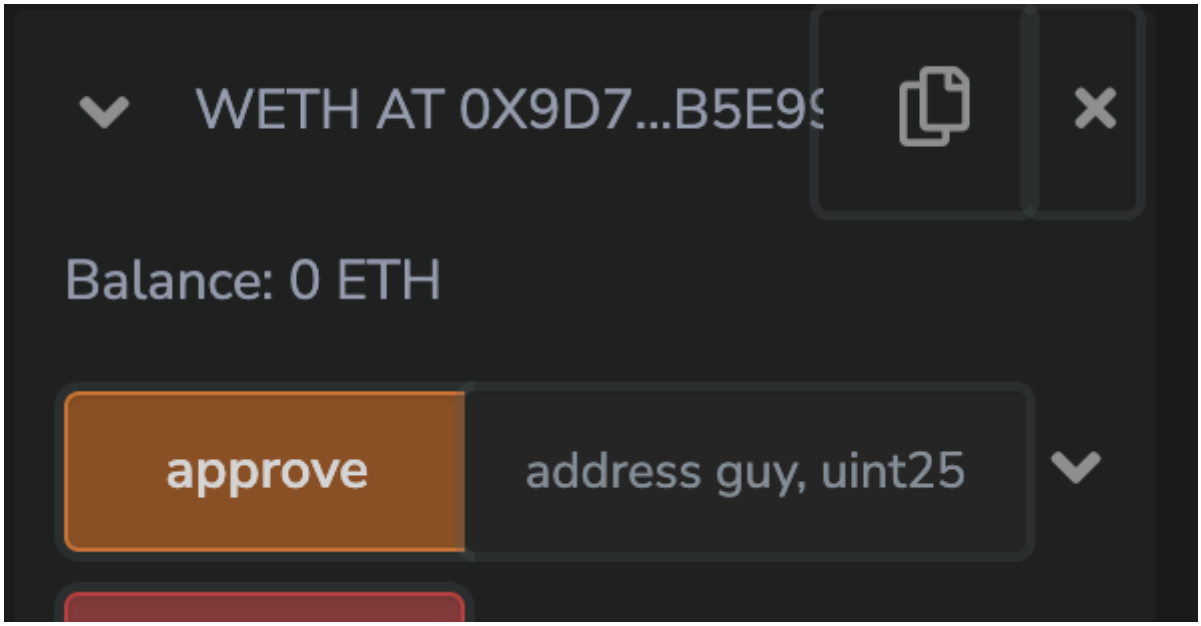
1. 使用depositAmount记录用户在活动期内的存款数量；
2. 对每个用户记录一个checkPoint，在每次用户存款或者提取本金时更新；checkPoint = depositAmount * currentBlockNumber;
3. 对每个用户记录一个calculatedReward，在每次用户存款/提取本金时更新；
4. 对每个用户记录一个claimedReward，在每次用户提取利息时记录；
5. 用户的待领取利息 = calculatedReward + 从上次操作到当前区块高度新产生的利息- claimedReward；
6. 用户在提取本金时，先返还之前的所有利息；
7. 所有对uint256的算数运算，使用SafeMath；

案例

准备好账户A：
0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB

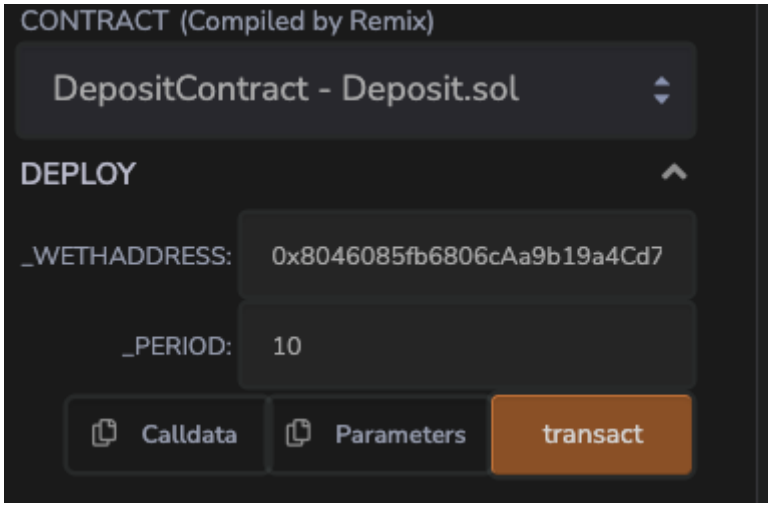

①在第4个区块充值10个WETH
②在第6个区块充值20个WETH
③在第8个区块提取本金10个ETH.
④在第9个区块提取利息.
⑤在第10个区块提取本金20个ETH
⑥在第13个区块查询链上状态.
Deposit Amount
存款数量
40
30
20
10
0 2 3 4 5 6 7 8 9 10 11 12 13 14
block. number
区块高度
备注①阴影面积即为 calculated Reward
②区块高度可以从任意高度开始.
只要满足上述操作顺序即可.

1. 部署WETH合约；


WETH AT 0X9D7...B5E9
Balance: 0 ETH
approve
address guy, uint25

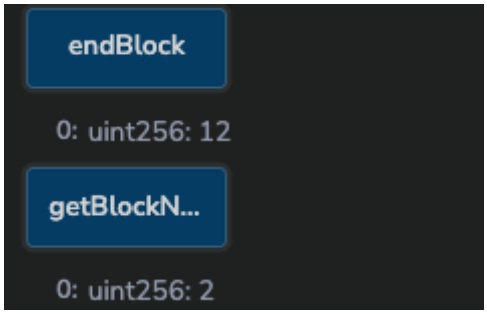
WETH 合约地址为：
0x8046085fb6806cAa9b19a4Cd7b3cd96374dD9573

2. 修改DepositContract合约里的_weth地址为真实地址，部署DepositContract合约，设置活动期为10个区块；



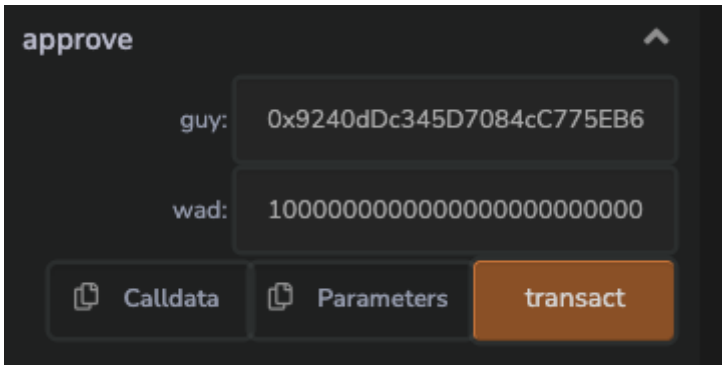
DepositContract 合约地址为：
0x9240dDc345D7084cC775EB65F91f7194DbBb48d8

3. 查看当前区块高度和endBlock；



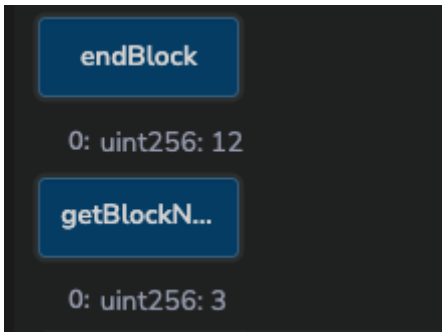
可以看到活动期为10个区块。

4. 在WETH合约里对DepositContract进行approve

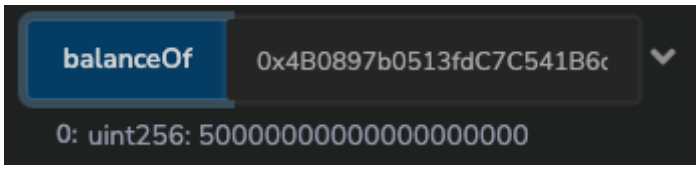


（想一想为什么要有这一步操作？）

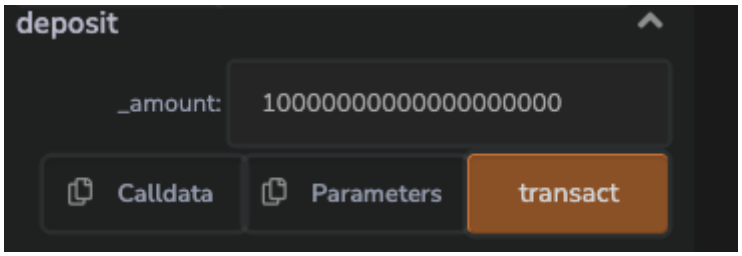
5. 查看此时区块高度



6. 充值50个WETH到WETH合约；



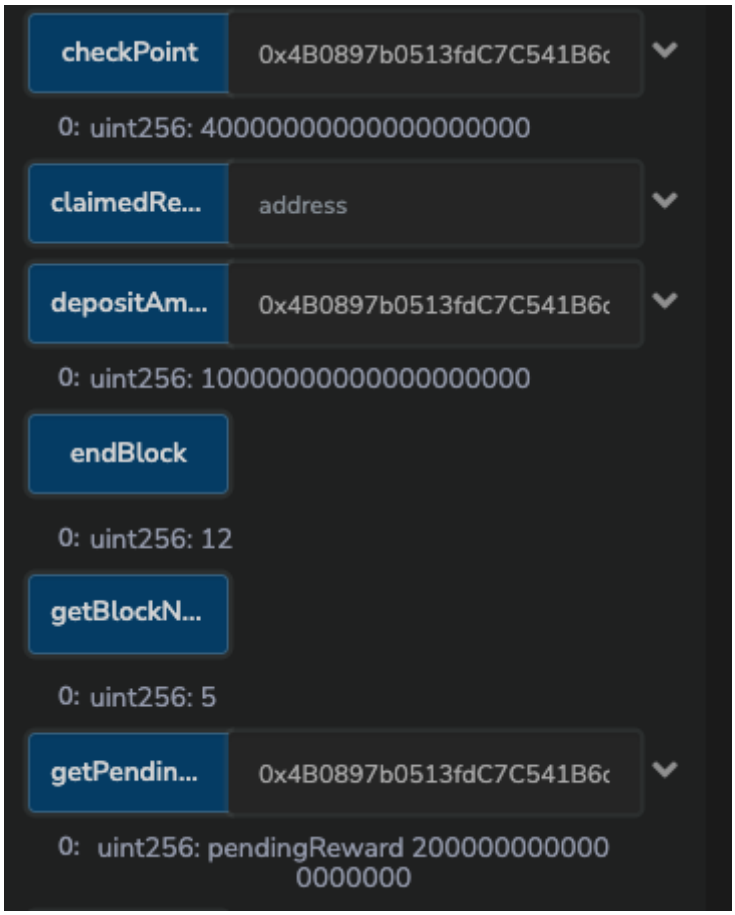
7. 充值10个WETH到DepositContract合约（对应红圈1）



8. 查询当前区块高度

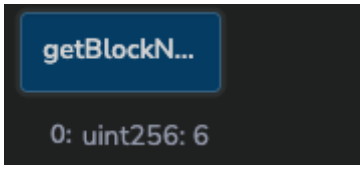
可以看到，此时区块链高度为5；

8. 调用DepositContract合约的相关方法，查看账户A的checkPoint、depositAmount和getPendingReward。

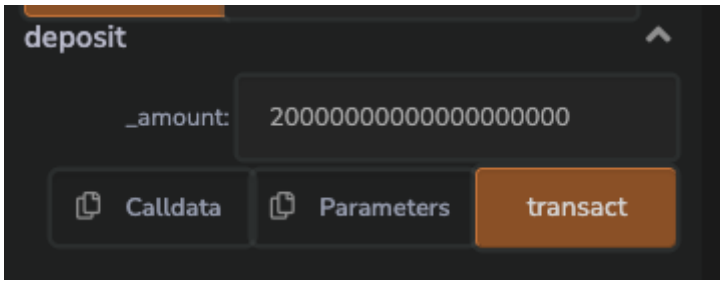


第7步的充值发生在第4个区块，当前区块高度为5，所以奖励 =（当前区块高度 - 充值区块高度）x 充值数量 / 奖励基准 = (5-4) x 10 / 5 = 2.跟图里的getPendingReward 返回结果一致；

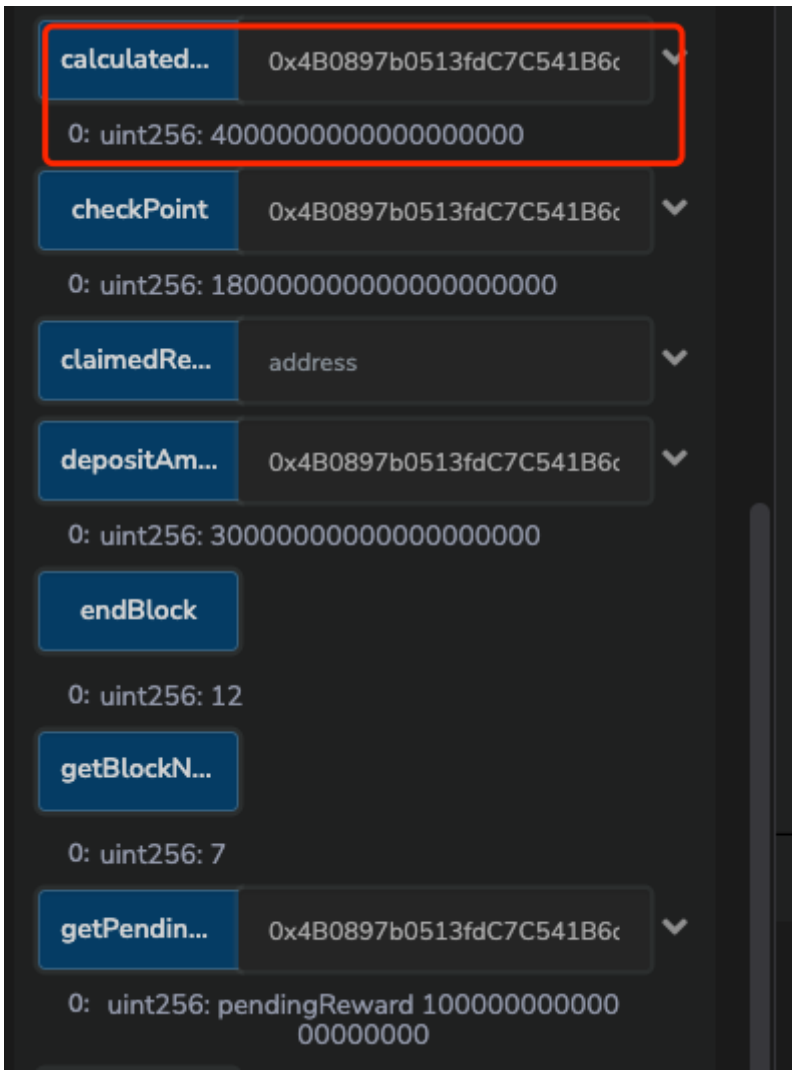
9. 调用DepositContract合约的addBlock方法，增加一个区块高度。



10. 充值20个WETH到DepositContract.（对应红圈2）

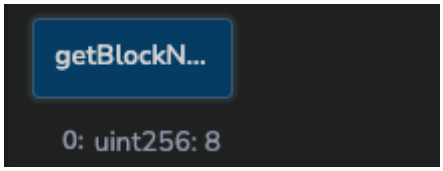


11. 调用DepositContract合约的相关方法，查看账户A的calculatedReward、checkPoint、depositAmount和getPendingReward。

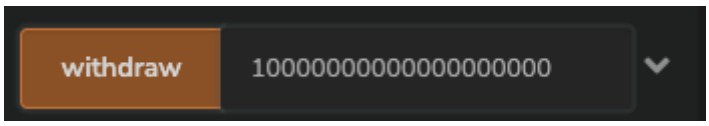


可以看到calculatedReward为4个，即为上图红色阴影区域；计算方法为(6 - 4) x 10 / 5 = 4;
其余20个WETH充值发生在区块高度6，当前区块高度为7，因此pengdingReward = 4 + (7 - 6) x 30 / 5 = 10个。

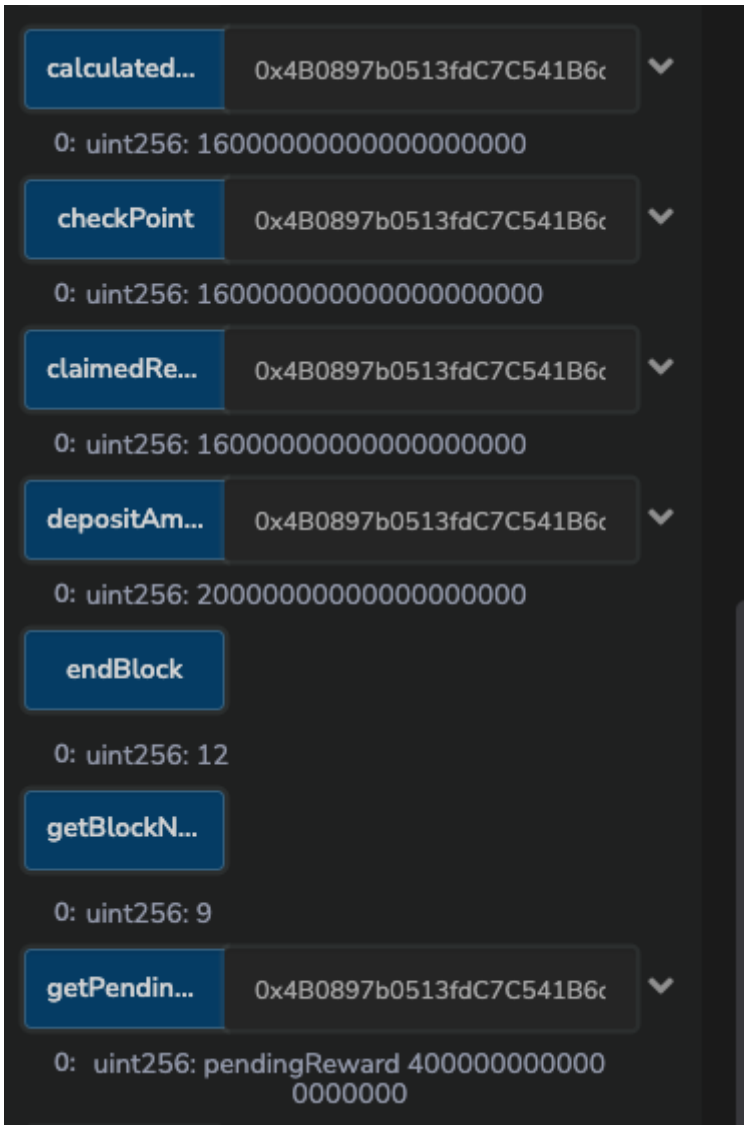
12. 调用DepositContract合约的addBlock方法，增加一个区块高度。



13. 调用DepositContract的withdraw方法，提取10个ETH本金。（对应红圈3）

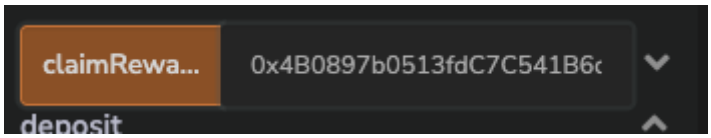


14. 查看账户A的calculatedReward/checkPoint/depositAmount/getPendingReward/claimedReward



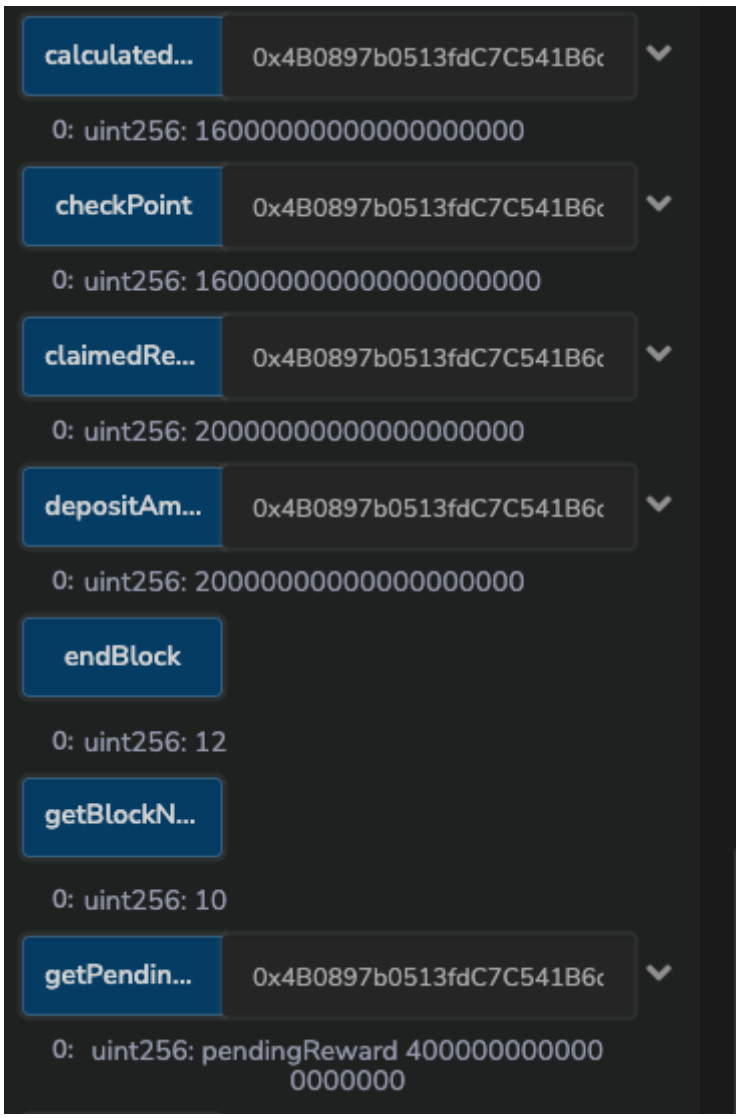
calculatedReward 为图中黑色阴影+红色阴影之和，计算方法为(6 - 4) x 10 / 5 + (8 - 6) x 30 / 5 = 4 + 12 = 16;
checkPoint = 区块高度 x 存款数量 = 8 x 20 = 160 x 1e18;
在提取本金时会把之前的奖励提出，提取了16个,因此claimedReward为16个;
depositAmount变成了20个WETH;
getPendingReward = (9 - 8) x 20 / 5 = 4;

15. 调用DepositContract的claimReward合约，提取利息到任意地址；（对应红圈4）

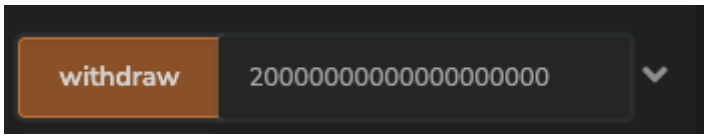


该比交易会被打包于高度为9的区块；

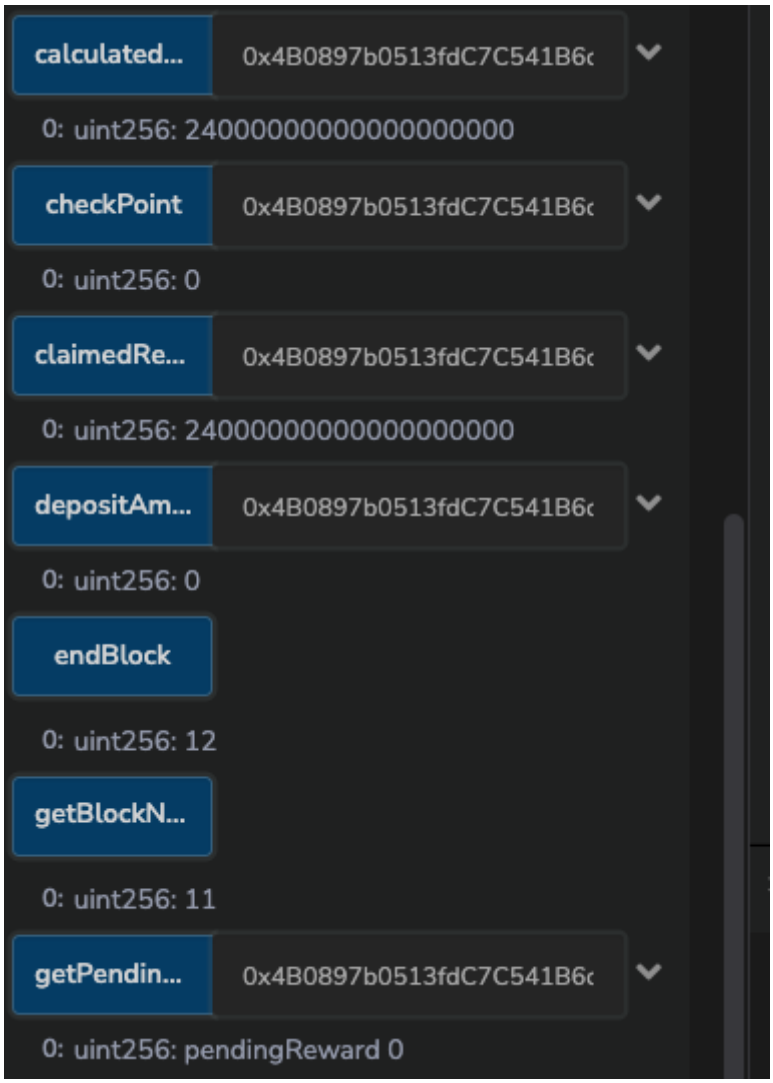
16. 查看当前链上状态



17. 调用DepositContract合约的withdraw方法，提取所有本金20个ETH；（对应红圈5）

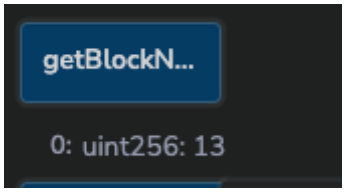


18. 查看此时的链上状态

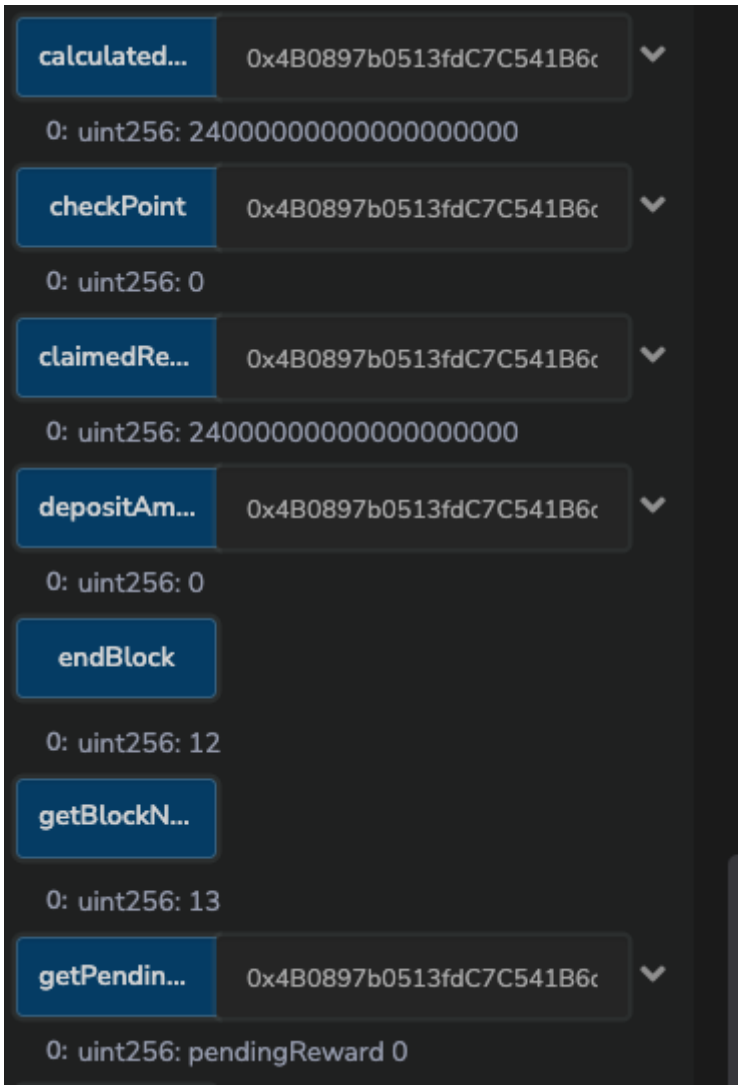


此时calculatedReward = claimedReward = 红色阴影面积 + 黑色阴影面积 + 黄色阴影面积 = (6 - 4) x 10 / 5 + (8 - 6) x 30 / 5 + (10 - 8) x 20 / 5 = 4 + 12 + 8 = 24；

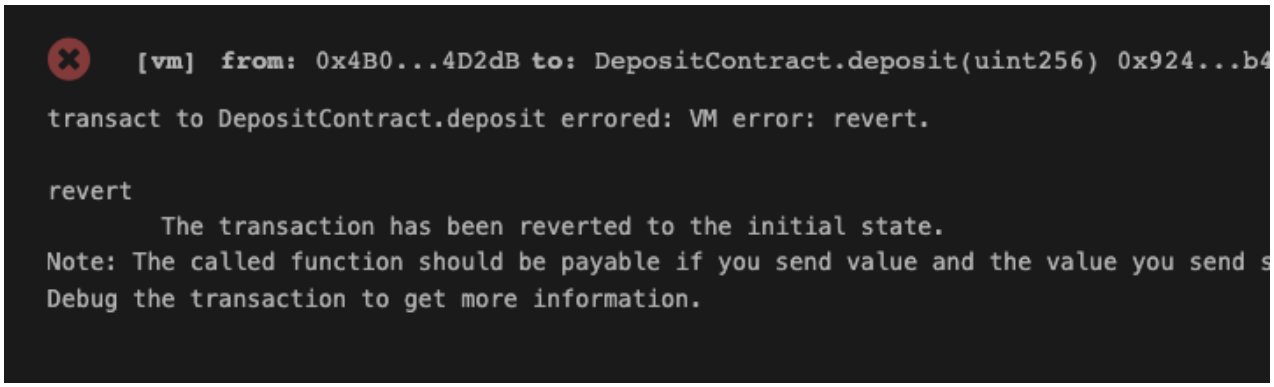
19. 调用DepositContract合约的addBlock方法，增加两个区块高度。



20. 查看此时链上状态（对应红圈6）



21. 调用DepositContract合约的deposit方法，会报错。因为已经超过有效时间。



实验报告内容

参考上面案例过程撰写实验报告。并回答以下问答题：

1. 阐述constant和immutable的联系与区别
2. 阐述modifier的用法；

在实验报告开头中写明：姓名/学号/班级，内容参考上面实验报告示例；

实验报告提交方式

实验报告完成后发送到邮箱 cbireport@163.com，标题为`学号-班级-姓名-第四次实验报告`

本次实验报告提交截止时间为第八次课结束之前。