# GEBToken.sol

常规代币合约需要注意以下几点：

1. 状态变量voteEnabled，用于标记现在DAICO的投票功能是否生效。
2. transfer和transferFrom函数，当voteEnabled为true的时候，代币转移完成会去调用fund和refund合约的修改用户投票权重的函数。
3. addTotal函数，用于增加代币总量。

# PublicRaising.sol(众筹合约)

该合约主要用于发起公募和存取公募资金。

## 状态变量：

round: 公募轮次记录。

price: 公募价格，即x geb = 1 eth。

startTime: 当前公募轮次的起始时间。

endTime: 当前公募轮次的结束时间。

saleTokens: 本轮公募的代币剩余量。

leftTokensUpperLimit: 本轮公募软顶。

结构体raiseParticipant和raiseParticipants数组：用于记录本次公募参与者的地址，购买金额和获得的代币数量。

inRasing: 用于标记是否处于众筹中，避免正在众筹时发起新众筹、用户在非众筹期间打钱等错误状态。

## 函数:

构造函数：

初始化代币合约地址。

newRaisingRound函数:

入参分别是公募价格、开始时间、结束时间、销售代币数量、软顶。功能是发起新一轮公募。

Fallback函数：

用于接受募集款项，并记录下用户可以获取的GEB数量。

finishRaise函数：

结束本轮公募，倘若达到软顶则发放GEB至投资者账户，反之，则退回募集款项。

sendFund函数：

用于发送募集资金，主要针对退款合约和资金合约分别发送退款和tap。

# Vote.sol

工具合约，主要用于对投票结果进行记录和反馈，任何合约均可以调用。

## 状态变量

totalVotesByAddress：用于记录发起投票的智能合约地址下面的每一次投票记录。struct singleVote用于真的每一个投票者进行记录，struct totalVotes用于记录下发起投票的智能合约的当前投票状态，minValidWeight代表本轮投票的最小有限投票数，倘若投票数小于该数值，则投票失败。maxSingleWeight代表每个投票者的最大权重，若代币拥有者代币数量，则投票权重为这个值。voteByAddress用于记录每一个投票者的权重和投票结果。

## 函数

voteStart函数：

开启新一轮投票，入参分别是投票开始时间、结束时间、最小有效票数、单人最大权重，通过传入的参数修改投票自己地址下的投票参数，并开启投票状态。

vote函数：

参与者投票函数，入参分别是投票者地址、投票结果、权重。通过其他合约记录下参与者的投票结果，并传入投票合约，调用该函数完成针对参与者的投票记录和统计工作。

revokeVote函数：

撤销投票函数，入参为撤销投票的参与者地址。通过其他合约的撤销投票函数记录下投票地址，其他合约将地址传入投票合约，调用该函数撤销参与者的投票，并修改统计结果，删除参与者投票记录。要求参与者必须曾经参与过本次投票并且合约处于可以投票的时间和状态。

voteStop函数：

到达投票结束时间后，发起投票的合约可以结束自己的投票，并得到本次投票结果，如果反对者大于赞同者或者未带到有效投票数量则本次投票结果为失败。

whenTokenTransfer函数：

当GEBToken主合约下的代币拥有者转移代币时，调用此函数，修改转移者和接受者已经参与投票的权重。

# Fund.sol

## 状态变量

TAP：每月基础Tap费用。

applyTap：通过发起投票申请的额外tap，每月可以申请一次。

appliedTap：申请成功的额外tap数。

applyStartTime：记录最近一次申请tap的时间，用于判断下一次申请的合法性。

MIN\_VALID\_VOTES：该合约发起的投票的最小有效投票数。

withdrawalTime：最近一次提取Tap的时间。

isContinue：该合约是否继续状态，仅可有退款合约更改为false状态，则该合约将不再可用，仅用于向投资者发送退款。

## 函数

timeValidation：

入参为需要检测的时间，判断传入的时间的公历格式是否在月数上小于现在的一个月或以上。

addTap：

申请增加tap，入参为申请的资金数。要求上一次申请的tap已经被取走并且距离上一次申请间隔在月数上相差1以上，并调用投票合约，发起申请投票。

finishApplyment：

结束申请tap的投票，若成功，则将applyTap的值赋予appliedTap，完成本次申请。

getTap：

获取当月tap，入参为获取本次tap的地址。

programStop：

由refund合约调用，如果退票投票成功，则本合约终止。

# Refund.sol

退款合约，每半年发起一次，倘若第一个退款投票成功，则下个月进行第二次，若两次投票均成功，则合约失效，投资者可退回相应款项。

## 状态变量

FIRST\_REFUND\_VOTE\_TIME：四轮退款投票发起第一次投票的时间。

SECOND\_REFUND\_VOTE\_TIME：四轮退款投票发起第二次投票的时间，如果第一投票失败，则不会发起第二次。

REFUND\_ROUND：当前退款投票轮次，0-3。

REFUND\_STAGE：当轮投票的阶段，1和2。

firstStageStatus：当轮投票的第一次投票结果。

isValid：退款合约有效性标志，当项目上交易所或退款投票成功时置为false。

MIN\_VALID\_VOTES：该合约发起的投票的最小有效投票数。

REFUND\_PRICE：退款价格，x geb = 1 ether。

## 函数

refundVoteStart：

根据当前投票的轮次和阶段发起向投票合约发起开始投票的请求，持续七天。

voteStatistics：

投票结果统计，如果第一次投票成功则进入第二次，失败则进入下一轮；如果第二次投票失败则进入下一轮，成功则合约失效，可发起退款。如果是最后一轮失败，则合约失效。

setRefundPrice：

当退款成功时设置退款价格。

refund：

当退款成功时，投资者通过该函数获取退款。

programSuccess：

当GEB项目上线交易所之后，则该合约失效。