# Аудит смарт-контракта SmartMMM

## Описание

Контракт наследуется от контракта Ownable из репозитория Open-Zeppelin.

## Переменные

1. *DepositItem*
   структура депозита, где:
   - time - время создания депозита или реинвеста
   - sum - сумма депозита
   - withdrawalTime - время последнего вывода
   - restartIndex - номер последнего рестарта, который был сделан для этого депозита
   - invested - инвестировано в контракт
   - payments - получено выплат
   - referralPayments - получено реферальных выплат
   - cashback - получено кэшбэка
   - referalsLevelOneCount - количество рефералов первого уровня
   - referalsLevelTwoCount - количество рефералов второго уровня
   - referrerLevelOne - реферер первого уровня
   - referrerLevelTwo - реферер второго уровня
2. *techSupport*
   адрес технической поддержки (устанавливается 0x799358af628240603A1ce05b7D9ea211b9D64304)
3. *adsSupport*
   адрес маркетингового фонда (устанавливается 0x8Fa6E56c844be9B96C30B72cC2a8ccF6465a99F9)
4. *deposits*
   соответствие адреса пользователя структуре депозита
5. *referrers*
   соответствие адреса пользователя статусу реферера, выставляется создателем контракта
6. *waitingReferrers*
   соответствие адреса пользователя времени оплаты им статуса реферера
7. *referrerPrice*
   сумма оплаты статуса реферера (устанавливается 70700000000000000, что равняется 0.0707 eth)
8. *referrerBeforeEndTime*
   время окончания акции по получению статуса реферера бесплатно (устанавливается 0)
9. *maxBalance*
   максимальный баланс контракта (устанавливается 0)
10. *invested*
    инвестировано в контракт
11. *payments*
    выведено с контракта
12. *referralPayments*
    реферальные выплаты

13. *investorsCount*
    количество инвесторов
14. *historyOfRestarts*
    массив, содержащий времена историй рестарта

## События

1. *Deposit*
   создается при совершении депозита или реинвеста
   Параметры:
   - from - адрес инвестора
   - value - сумма в wei
2. *Withdraw*
   создается при выводе накопившихся средств
   Параметры:
   - to - адрес инвестора
   - value - сумма в wei
3. *PayBonus*
   создается при выплате бонуса (реферальный, кэшбэк)
   Параметры:
   - to - адрес получателя бонуса
   - value - сумма в wei

## Функционал

1. **Конструктор** контракта
   1.1. Функционал
      1.1.1. Запишет время создания контракта в массив historyOfRestarts.
2. Функция **bytesToAddress**
   2.1. Функция доступна только для данного контракта.
   2.2. Принимает на вход:
      2.2.1. source - любые байты
   2.3. Функционал:
      2.3.1. Создается переменная parsedAddress и ей задается значение адреса из байт source.
   2.4. Возвращается:
      2.4.1. Адрес рефрера parsedAddress.
3. Функция **getPercents**
   3.1. Функция доступна для всех и не требует газа для выполнения.
   3.2. Принимает на вход:
      3.2.1. balance - текущий баланс контракта.
   3.3. Возвращается:
      3.3.1. В зависимости от баланса:
         3.3.1.1. Минутный процент депозита.
         3.3.1.2. Процент реферера первого уровня.
         3.3.1.3. Процент реферера второго уровня.
         3.3.1.4. Процент кэшбэка.
         3.3.1.5. Процент тех поддержки.
         3.3.1.6. Процент маркетингового фонда.
   3.4. Примечания:

3.4.1. Точные цифры процентов можно найти в коде контракта в строках 186-197.

4. Функция **fallback** (вызывается при отправке эфиров на контракт)
   Функция доступна всем и принимает эфир.
   Запоминается текущий баланс контракта. Запоминаются значения процентов из функции **getPercents**. Если количество посланного на контракт эфира равно 0, то вызывается функция **payWithdraw** и выполнение функции заканчивается. Если количество эфира равно переменной *referrerPrice* и отправитель еще не является рефером и время оплаты статуса реферера равно 0 и отправитель уже является вкладчиком, то время оплаты статуса рефера для отправителя меняется на текущее, иначе вызывается функция **addDeposit** с адресом отправителя, количеством пришедшего эфира, балансом контракта и всеми процентами из функции **getPercents**, в качестве аргументов.
5. Функция **isNeedRestart**
   5.1. Функция доступна только для данного контракта.
   5.2. Принимает на вход:
      5.2.1. balance - текущий баланс контракта.
   5.3. Функционал:
      5.3.1. Если текущий баланс контракта balance меньше 30% от максимального баланса *maxBalance*:
         5.3.1.1. Максимальному балансу *maxBalance* задается значение 0.
         5.3.1.2. Возвращается true.
      5.3.2. Иначе возвращается false.
6. Функция **calculateNewTime**
   6.1. Функция доступна всем.
   6.2. Принимает на вход:
      6.2.1. oldTime - старое время.
      6.2.2. oldSum - старую сумму.
      6.2.3. newSum - новую сумму.
      6.2.4. currentTime - текущее время.
   6.3. Возвращается:
      6.3.1. oldTime + newSum / (newSum + oldSum) * (currentTime - oldTime)
7. Функция **calculateNewDepositSum**
   7.1. Функция доступна всем.
   7.2. Принимает на вход:
      7.2.1. minutesBetweenRestart - время между рестартами.
      7.2.2. minutesWork - время работы депозита.
      7.2.3. depositSum - сумма депозита.
   7.3. Функционал:
      7.3.1. Если время работы депозита больше чем время между рестартами, то время работы приравнивается к времени между рестартами.
   7.4. Возвращается:
      depositSum * (100 - (minutesWork * 100 / minutesBetweenRestart) + 7) / 100
8. Функция **addDeposit**
   8.1. Функция доступна только для данного контракта.
   8.2. Принимает на вход:
      8.2.1. investorAddress - адрес вкладчика.
      8.2.2. weiAmount - сумма вклада в wei.
      8.2.3. balance - текущий баланс контракта.
      8.2.4. referrerLevelOnePercent - процент реферера первого уровня.
      8.2.5. referrerLevelTwoPercent - процент реферера второго уровня.

8.2.6. cashBackPercent - процент кэшбэка.
8.2.7. depositPercent - процент начислений в минуту.
8.2.8. techSupportPercent - процент тех поддержки.
8.2.9. adsSupportPercent - процент маркетингового фонда.

8.3. Функционал:

8.3.1. Вызывается функция **checkReferrer** с investorAddress, weiAmount, referrerLevelOnePercent, referrerLevelTwoPercent, cashBackPercent в качестве аргументов.
8.3.2. На время работы функции копируется структура депозита deposit по адресу investorAddress.
8.3.3. Если сумма deposit равна 0, то депозиту устанавливается текущее время и общее количество инвесторов увеличивается на 1, иначе к текущей сумме депозита прибавляется значение из функции **getWithdrawSum** с investorAddress и depositPercent в качестве аргументов и время депозита устанавливается вызовом функции **calculateNewTime** с deposit.time, deposit.sum, weiAmount и текущим временем в качестве аргументов.
8.3.4. Время последнего вывода устанавливается на текущее.
8.3.5. К сумме депозита прибавляется количество инвестированных wei.
8.3.6. Последним рестартом назначается последний рестарт на данный момент.
8.3.7. Общее количество инвестиций увеличивается на количество инвестированных wei.
8.3.8. Депозит в памяти смарт-контракта заменяется измененной копией deposit.
8.3.9. Создается событие **Deposit** с адресом инвестора и суммой в wei в качестве аргументов.
8.3.10. Вызывается функция **payToSupport** с weiAmount, techSupportPercent, adsSupportPercent в качестве аргументов.
8.3.11. Если *maxBalance* меньше чем balance, то *maxBalance* делается равным balance.
8.3.12. *invested* увеличивается на weiAmount.

9. Функция **payToSupport**

9.1. Функция доступна только для данного контракта.

9.2. Принимает на вход:

9.2.1. weiAmount - количество инвестированных wei.
9.2.2. techSupportPercent - процент тех поддержки.
9.2.3. adsSupportPercent - процент маркетингового фонда.

9.3. Функционал:

9.3.1. На адрес *techSupport* отправляется процент techSupportPercent от суммы weiAmount.
9.3.2. На адрес *adsSupport* отправляется процент adsSupportPercent от суммы weiAmount.

10. Функция **checkReferrer**

10.1. Функция доступна только для данного контракта.

10.2. Принимает на вход:

10.2.1. investorAddress - адрес инвестора.
10.2.2. weiAmount - количество инвестированных wei.
10.2.3. referrerLevelOnePercent - процент реферера первого уровня.
10.2.4. referrerLevelTwoPercent - процент реферера второго уровня.
10.2.5. cashBackPercent - процент кэшбэка.

10.3. Функционал:

10.3.1. Создается переменная referrerLevelOneAddress, ей задается адрес реферера первого уровня из информации о депозите по адресу investorAddress.

10.3.2. Создается переменная referrerLevelTwoAddress, ей задается адрес реферера второго уровня из информации о депозите по адресу investorAddress.

10.3.3. Если сумма депозита по адресу invesotrAddress равна нулю и если отправленная на контракт data имеет длину 20 байт (проверка на наличие адреса в data):

10.3.3.1. referrerLevelOneAddress присваивается значение из функции **bytesToAddress** с отправленной на контракт data в качестве аргумента.

10.3.3.2. Если адрес referrerLevelOneAddress не является адресом самого инвестора и не является 0x0:

10.3.3.2.1. Если адрес referrerLevelOneAddress получил статус реферера от админа или он оплатил статус реферера более 7 дней назад или текущее время менее заданной даты *referrerBeforeEndTime*:

10.3.3.2.1.1. В структуре депозита по адресу investorAddress полю *referrerLevelOne* присваивается referrerLevelOneAddress.

10.3.3.2.1.2. В структуре депозита по адресу referrerLevelOneAddress поле *referalsLevelOneCount* увеличивается на 1.

10.3.3.2.1.3. Переменной referrerLevelTwoAddress присваивается адрес реферера первого уровня для referrerLevelOneAddress, т.е. значение поля referrerLevelOne из структуры депозита по адресу referrerLevelOneAddress.

10.3.3.2.1.4. Если адрес referrerLevelTwoAddress не является адресом самого инвестора и не является 0x0:

10.3.3.2.1.4.1. В структуре депозита по адресу investorAddress полю *referrerLevelTwo* присваивается referrerLevelTwoAddress.

10.3.3.2.1.4.2. В структуре депозита по адресу referrerLevelTwoAddress поле *referalsLevelTwoCount* увеличивается на 1.

10.3.4. Если адрес referrerLevelOneAddress не является 0x0:

10.3.4.1. Создается переменная cashBackBonus и ей присваивается процент cashBackPercent от суммы weiAmount в качестве значения.

10.3.4.2. Создается переменная referrerLevelOneBonus и ей присваивается процент referrerLevelOnePercent от суммы weiAmount в качестве значения.

10.3.4.3. Создается событие **PayBonus** с адресом инвестора investorAddress и суммой в wei cashBackBonus в качестве аргументов.

10.3.4.4. Создается событие **PayBonus** с адресом реферера первого уровня referrerLevelOneAddress и суммой в wei referrerLevelOneBonus в качестве аргументов.

10.3.4.5. К общему количеству реферальных выплат *referralPayments* прибавляется значение referrerLevelOneBonus.

10.3.4.6. В структуре депозита по адресу referrerLevelOneAddress количество реферальных выплат, т.е. поле *referralPayments* увеличивается на значение referrerLevelOneBonus.

10.3.4.7. На адрес referrerLevelOneAddress отправляется количество wei referrerLevelOneBonus.
10.3.4.8. В структуре депозита по адресу investorAddress кэшбэк, т.е. поле cashback увеличивается на значение cashBackBonus.
10.3.4.9. На адрес investorAddress отправляется количество wei cashBackBonus.
10.3.4.10. Если адрес referrerLevelTwoAddress не является 0x0:
10.3.4.10.1. Создается переменная referrerLevelTwoBonus и ей присваивается процент referrerLevelTwoPercent от суммы weiAmount в качестве значения.
10.3.4.10.2. Создается событие **PayBonus** с адресом реферера второго уровня referrerLevelTwoAddress и суммой в wei referrerLevelTwoBonus в качестве аргументов.
10.3.4.10.3. К общему количеству реферальных выплат *referralPayments* прибавляется значение referrerLevelTwoBonus.
10.3.4.10.4. В структуре депозита по адресу referrerLevelTwoAddress количество реферальных выплат, т.е. поле *referralPayments* увеличивается на значение referrerLevelTwoBonus.
10.3.4.10.5. На адрес referrerLevelTwoAddress отправляется количество wei referrerLevelTwoBonus.

11. Функция **payWithdraw**
11.1. Функция доступна только для данного контракта.
11.2. Принимает на вход:
11.2.1. to - адрес инвестора, куда выводить средства.
11.2.2. balance - текущий баланс контракта.
11.2.3. percent - текущий минутный процент.
11.3. Функционал:
11.3.1. Если сумма депозита по адресу to меньше 0, то выполнение контракта приостанавливается.
11.3.2. Вызывается функция **isNeedRestart** с текущим балансом контракта balance в качестве аргумента.
11.3.3. Если **isNeedRestart** вернуло true:
11.3.3.1. В список историй рестартов *historyOfRestarts* добавляется текущие дата и время.
11.3.4. Создается переменная lastRestartIndex и ей задается значение номера последнего рестарта.
11.3.5. Если разница между номером последнего рестарта lastRestartIndex и номером рестарта в депозите по адресу to, т.е. поле *restartIndex* больше или равно 1:
11.3.5.1. Создается переменная minutesBetweenRestart и ей задается значение частного разницы между значением рестарта по номеру lastRestartIndex и значением рестарта по номеру рестарта для депозита по адресу to и 60 секунд.
11.3.5.2. Создается переменная minutesWork и ей задается значение частного разницы между значением рестарта по номеру lastRestartIndex и значением времени (в UNIX) создания депозита или реинвеста из структуры депозита по адресу to и 60 секунд.
11.3.5.3. Сумме, т.е. полю *sum*, в структуре депозита по адресу to задается значение из функции **calculateNewDepositSum** с минутами между рестартами minutesBetweenRestart и минутами работы депозита

minutesWork и текущей суммой депозита по адресу to в качестве аргументов.

11.3.5.4. Последнему номеру рестарта, т.е. полю *restartIndex*, в структуре депозита по адресу to задается значение текущего последнего номера рестарта lastRestartIndex.

11.3.5.5. Времени, т.е. полю *time*, в структуре депозита по адресу to задается значение текущего времени (UNIX).

11.3.6. Создается переменная текущей суммы вывода sum и ей задается значение функции **getWithdrawSum** с адресом инвестора to и процентом percent в качестве аргументов.

11.3.7. Если значение переменной sum меньше 0, то выполнение контракта останавливается и все изменения отменяются.

11.3.8. Времени вывода, т.е. поле *withdrawalTime*, в структуре депозита по адресу to задается значение текущего времени (UNIX).

11.3.9. Количество выплат, т.е. поле *payments*, в структуре депозита по адресу to увеличивается на значение переменной текущей суммы вывода sum.

11.3.10. Значение общих выплат с контракта *payments* увеличивается на значение переменной текущей суммы вывода sum.

11.3.11. Инвестору по адресу to отравляется количество wei из переменной sum.

11.3.12.Создается событие **Withdraw** с адресом инвестора to и количеством выведенных wei sum в качестве аргументов.

12. Функция **getWithdrawSum**
    12.1. Функция доступна только для данного контракта
    12.2. Принимает на вход:
        12.2.1. investorAddress - адрес инвестора.
        12.2.2. percent - текущий минутный процент.
    12.3. Функционал:
        12.3.1. Создается переменная разницы между текущим временем и временем вывода депозита по адресу investorAddress, в минутах, minutesCount.
        12.3.2. Создается переменная суммы вывода sum и ей задается значение: текущая сумма депозита по адресу investorAddress sum * percent / 10000000000000000 * minutesCount.
    12.4. Возвращается сумма вывода sum.
13. Функция **addReferrer**
    13.1. Функция доступна только для создателя контракта.
    13.2. Принимает на вход:
        13.2.1. referrerAddress - адрес реферера.
    13.3. Функционал:
        13.3.1. Статусу реферера в соответствии *referrers* по адресу referrerAddress задается значение true.
14. Функция **setReferrerPrice**
    14.1. Функция доступна только для создателя контракта.
    14.2. Принимает на вход:
        14.2.1. newPrice - новая цена за статус реферера.
    14.3. Функционал:
        14.3.1. Цене оплаты статуса реферера *referrerPrice* задается значение newPrice.
15. Функция **setReferrerBeforeEndTime**
    15.1. Функция доступна только для создателя контракта.

15.2. Принимает на вход:
15.2.1. newTime - новое время завершения акции по получению статуса реферера.
15.3. Функционал:
15.3.1. Текущему время завершения акции по оплате статуса реферера *referrerBeforeEndTime* задается значение newTime.

16. Функция **getDaysAfterStart**
16.1. Функция доступна всем и не требует газа для выполнения.
16.2. Функционал:
16.2.1. Создается переменная количества дней со старта работы контракта daysAfterStart и ей задается значение частного разницы между текущим временем и временем первого рестарта из списка *historyOfRestarts* и количестве секунд в 1 дне.
16.3. Возвращается:
16.3.1. Количество дней со старта работы контракта daysAfterStart.

17. Функция **getDaysAfterLastRestart**
17.1. Функция доступна всем и не требует газа для выполнения.
17.2. Функционал:
17.2.1. Создается переменная количества дней с последнего рестарта daysAfeterLastRestart и ей задается значение частного разницы между текущим временем и временем последнего рестарта из списка *historyOfRestarts* и количестве секунд в 1 дне.
17.3. Возвращается:
17.3.1. Количество дней с последнего растрата daysAfeterLastRestart.