README — KokoWinToken (KOKO)

Основная информация о токене

Имя токена: KokoWin
Символ: KOKO
Десятичное представление: 18
Начальное количество токенов: 1,683,117,110
KokoWinToken реализован в виде смарт-контракта, на Solidity 0.8.7. , стандарта TRC-20 на блокчейне TRON.

Первоначальное распределение.

Реализована функция (transferOwnership), которая дает возможность текущему владельцу безопасно передать права на управление смарт-контрактом новому адресу.
Контракт следует шаблону Ownable, предоставляя особые права владельцу.
Первоначально весь объем токенов (**1,683,117,110 KOKO**), зачисляется на баланс владельца, таким образом, после запуска контракта владелец контролирует все токены и отвечает за их начальное распределение среди пользователей, бирж или других целей.
Владелец контракта TF8Rb692aMHGkZVhfAdQRsCrQ1zxpct4Mm
После развертывания контракта часть токенов будет заморожена на кошельках:

Кошельки проекта	Разблокировка	Токены
TAbt4mwB5PJMmZxhjPfumJpdKQRRRELAPb	16.05.2025	16 000000
TVXwUtbqPed7s9T7Dhme5FkZ5ZTuonSrnV	03.11.2026	20 000000
TGeetw91cFn1rG7m8Sc9W6x6kdwCJcxvk4	31.10.2027	78 000000
TNvc82f2BXCwGDKRJDVLtQ85kQKiXa4E2f	03.01.2028	97 000000
TXGZfjZTqr1MDKH4YXgkz5qJPVMMSrAZZu	22.05.2029	10 700000
TQcksDk1HLtGwhCM3b9u2HYSEgWNkTEjdc	06.11.2030	97 000000
TCFW5AKfjGAS7CuNcUUwPd5MdKNzo6fpTG	09.02.2031	10 900000
TSndzL7C2QGDx5WTdKJNrFwtKMeArJeQHs	30.07.2032	11 100000
TQUkbVsUSdiaQCQYaz9SWgKAmTjCdwxeV9	22.08.2033	11 600000
TF6x3jQ4VD44bWxGe3VGqdEe8Hcg1yC7C6	25.08.2034	11 100000
Сумма	363 400000 (21.59%)	

Механизмы безопасности

В контракт KokoWinToken встроены несколько механизмов безопасности для защиты пользователей и средств:

- onlyOwner: Многие функции администрирования помечены модификатором onlyOwner. Это означает, что вызвать такие функции может только текущий владелец контракта. Данный механизм предотвращает несанкционированное использование важных функций сторонними адресами. Например, функции паузы, распределения токенов, восстановления токенов и др. ограничены для вызова только владельцем.
- nonReentrant: К некоторым критически важным функциям применён модификатор nonReentrant, защищающий от атак повторного входа (reentrancy). Этот механизм не позволяет функции быть вызванной повторно до её завершения при попытке вредоносного вмешательства. Таким образом предотвращается возможность злоумышленнику вызвать функцию повторно через внешние вызовы до завершения первоначального выполнения, что защищает баланс токенов и состояние контракта от непредусмотренных изменений.
- Пауза (pause)/Возобновление (unpause): Контракт поддерживает режим паузы, реализованный с помощью функций pause() и unpause() (модификаторы whenNotPaused/whenPaused). Владелец имеет право приостановить работу основных функций токена, например, переводов и иных операций, вызвав паузу. В состоянии паузы большинство операций (такие как передача токенов) будут запрещены. Это может быть полезно в экстренных случаях, например, при обнаружении уязвимости или подозрительной активности. После устранения проблемы владелец может возобновить нормальную работу контракта, вызвав функцию unpause().
- Функция остановки новых стейкеров (setStakingStatus) позволяет владельцу контракта временно запретить добавление новых депозитов в стейкинг. Это необходимо для контроля роста эмиссии и предотвращения чрезмерного начисления вознаграждений.

Эти механизмы безопасности повышают надежность контракта, защищая от несанкционированного доступа и распространённых атак, а также позволяя реагировать на внештатные ситуации.

Стейкинг

Контракт KokoWinToken предоставляет держателям возможность получать пассивный доход через механизм стейкинга. Реализовано три плана стейкинга, каждый с уникальными условиями:

- Каждый план предполагает определённый период удержания токенов и соответствующую ставку вознаграждения. Например, может быть короткий срок с меньшим процентом вознаграждения, средний срок с повышенным процентом и долгосрочный план с наибольшим вознаграждением. Это даёт пользователям гибкость выбирать между более быстрым вознаграждением или более высоким доходом при длительном удержании.
- Для участия в стейкинге пользователь вызывает функцию stake(amount, plan), указывая сумму токенов KOKO и выбранный план (из трёх доступных). Токены в указанном объёме переводятся на контракт и блокируются на условиях выбранного плана. Контракт сохраняет информацию о депозите: адрес стейкера, сумму, выбранный план и время начала стейкинга.
- Пользователь может забрать свои токены обратно, вызвав функцию unstake(). При выводе (анстейкинге) контракт рассчитывает полагающееся вознаграждение согласно условиям выбранного плана и длительности стейкинга. Пользователь получает обратно свою изначальную сумму токенов плюс начисленные награды (если условия плана выполнены, например, выдержан минимальный срок). Все выплаты вознаграждений происходят в токенах KOKO.
- Максимальный срок начисления вознаграждений – 180 дней. Это означает, что независимо от того, как долго токены находятся в стейкинге, после достижения порога в 180 дней новые награды перестанут накапливаться. Данный лимит введён для предотвращения бесконечного роста задолженности по вознаграждениям и стимулирования пользователей своевременно выходить из стейкинга или переразмещать токены.

Механизм стейкинга поощряет долгосрочное держание токенов и участие в экосистеме KokoWin. Пользователи, блокируя свои токены на заранее оговоренный срок, способствуют стабильности курса, а взамен получают вознаграждение в виде дополнительных токенов.

Заморозка токенов

Помимо стейкинга, контракт позволяет замораживать (блокировать) токены на определённое время с помощью функций lockTokens и claimLockedTokens:

- Функция lockTokens(address получатель, uint256 сумма, uint256 время Разблокировки) используется владельцем контракта для блокировки определённого количества токенов на адресе получателя до наступления указанного времени разблокировки. При вызове этой функции соответствующая сумма токенов списывается с баланса владельца и удерживается контрактом, помечаясь как "замороженная" для указанного получателя до оговоренной даты/времени.
- Заблокированные таким образом токены нельзя потратить или перевести, пока не наступит время их разблокировки. Это обеспечивает механизм вестинга или отложенного распределения токенов. Например, команда разработчиков может заморозить свои токены на год, или токены для партнёров/инвесторов могут быть заблокированы до определённой даты, чтобы гарантировать долгосрочные обязательства.
- Когда наступает время разблокировки, указанный получатель может получить свои токены, вызвав функцию claimLockedTokens(). Эта функция проверяет, что текущая дата/время на блокчейне достигла (или превысила) указанное время разблокировки. Если условие выполнено, контракт переводит ранее заблокированные токены со своего адреса на адрес получателя, фактически возвращая ему принадлежащие токены.
- Если claimLockedTokens вызывается до наступления времени разблокировки, контракт отклонит запрос (не позволит вывести токены преждевременно). Только после наступления заданного срока токены становятся доступными для получения.
- Для каждой заморозки контракт хранит информацию: кому предназначены токены, в каком количестве и до какого времени они заблокированы. После успешного claimLockedTokens соответствующая запись разблокировки удаляется, и токены более не числятся как заблокированные.

Механизм заморозки токенов позволяет реализовать отложенное распределение токенов, защищая их от использования до наступления определённых условий. Это важно для обеспечения доверия и стабильности: участники, чьи токены заблокированы, знают, что они смогут получить их только в установленное время.

Airdrop

Контракт включает функции для массового распределения токенов и восстановления ошибочно отправленных средств:

- **Airdrop:** Владелец может проводить раздачу токенов на множество адресов с помощью специальной функции airdrop. Эта функция позволяет владельцу отправлять определённое количество токенов на список адресов одним действием. Благодаря этому, распределение токенов (например, в рамках маркетинговой кампании или программы лояльности сообщества) становится более эффективным и не требует выполнения множества отдельных транзакций. Все токены для airdrop списываются с баланса владельца.

Эти функции обеспечивают гибкость управления токенами: airdrop упрощает раздачу.

Сжигание токенов

KokoWinToken поддерживает прямое сжигание токенов через функцию burn. Сжигание приводит к безвозвратному уничтожению определённого количества токенов и сокращению общего предложения:

- Любой держатель токенов KOKO может вызвать функцию burn(uint256 amount), чтобы уничтожить часть своих токенов. При вызове burn указанная сумма списывается с баланса вызывающего и вычитается из общего количества выпущенных токенов (Total Supply).
- Сожжённые токены навсегда удаляются из обращения и не могут быть восстановлены. Это добровольный акт со стороны держателя (или иной уполномоченной сущности) для сокращения предложения.
- Функция burn может использоваться, например, владельцем для уничтожения нераспроданных токенов после проведения продажи, либо любым участником сообщества, желающим сократить количество обращающихся токенов. Сокращение общего предложения через сжигание теоретически увеличивает долю каждого оставшегося токена в общей массе, что может положительно влиять на ценность токена.

Каждый факт сжигания токенов фиксируется в блокчейне событием (см. событие Burn ниже), что обеспечивает прозрачность этого процесса.

Работа с нативными токенами (TRX)

Контракт KokoWinToken способен взаимодействовать с нативной криптовалютой блокчейна TRON – монетами TRX:

- Контракт настроен на приём TRX: он содержит механизм (payable-функцию), позволяющий ему получать на свой адрес монеты TRX. Например, если кто-то случайно или намеренно отправит TRX на адрес контракта, средства поступят на баланс контракта (в отличие от некоторых контрактов, которые отклоняют такие переводы).
- Владелец имеет право выводить накопленные на контракте TRX. Для этого реализована функция (доступная только владельцу), которая переводит весь (или часть) баланс TRX, хранящийся на контракте, на адрес владельца. Эта функция позволяет безопасно извлекать любые поступившие на контракт нативные монеты, чтобы они не остались заблокированными навсегда.
- Также в контракте есть функция getTRXBalance(), предоставляющая возможность в любой момент узнать текущий баланс TRX на контракте. Это «read-only» функция (не изменяющая состояние), которую могут вызывать внешние участники для проверки, сколько TRX удерживается контрактом. Она возвращает количество монет TRX, находящееся на балансе адреса контракта.

Поддержка работы с TRX делает контракт более гибким: он может аккумулировать нативные токены для оплаты комиссий сети или других целей и безопасно передавать их владельцу по необходимости.

События

Смарт-контракт генерирует ряд событий, записываемых в логах блокчейна, что повышает прозрачность и позволяет отслеживать действия:

- **Transfer:** стандартное событие токена TRC-20, возникающее при каждом переносе токенов с одного адреса на другой. Содержит информацию об адресе отправителя, адресе получателя и количестве токенов. Включает в себя также случаи сжигания (когда получателем указан адрес 0x0) и внутреннего распределения (airdrop).
- **Approval:** стандартное событие подтверждения разрешения, генерируемое при вызове функции approve. Событие фиксирует, что владелец токенов

предоставил определённому адресу-спендеру право потратить указанное количество его токенов.

- **Staked:** специальное событие стейкинга, эмитируемое при успешном выполнении функции stake. Содержит данные: адрес стейкера, идентификатор или тип выбранного плана, сумму застейканных токенов и время начала стейка. Это позволяет отслеживать, кто и сколько токенов поместил в стейкинг.
- **Unstaked:** событие, генерируемое при выводе токенов из стейкинга (unstake). Содержит информацию об адресе стейкера и сумме возвращённых токенов (включая начисленное вознаграждение), а также, возможно, идентификатор плана или продолжительность стейка. Это позволяет видеть, когда пользователь завершил стейкинг и сколько токенов получил обратно.
- **TokensLocked:** событие, сигнализирующее о том, что определённое количество токенов было заморожено для указанного адреса. Эмитируется при вызове lockTokens. В событии фиксируются: адрес, для которого заблокированы токены, сумма и время разблокировки. Это даёт прозрачность в отношении запущенных программ вестинга или иных заморозок токенов.
- **LockedTokensClaimed:** событие, возникающее при успешном получении заблокированных ранее токенов (вызов claimLockedTokens). Содержит адрес получателя и количество разблокированных токенов. Позволяет проследить, что токены были успешно разблокированы и возвращены владельцу по истечении срока блокировки.
- **Airdrop:** событие, фиксирующее факт массового распределения токенов посредством airdrop. В зависимости от реализации, может генерироваться либо единичное событие с общей информацией (например, количество адресов и общая сумма), либо серия событий по каждому получателю. В любом случае, это позволяет в логах отследить проведение раздачи токенов из резервов владельца.
- **TokensRecovered:** событие восстановления токенов, эмитируемое при вызове recoverTokens. В логе фиксируется, какой токен (адрес контракта токена) и какое количество было восстановлено владельцем из баланса данного контракта. Это делает процесс возврата ошибочно отправленных средств прозрачным для наблюдателей.
- **Burn:** событие сжигания токенов. Генерируется при вызове функции В событии указывается, с какого адреса сожжены токены и в каком

количестве. Событие Burn явным образом фиксирует уменьшение общего предложения токена.

- **StakingStatusChanged:** событие, фиксирующее изменение доступности стейкинга. Эмитируется при вызове setStakingStatus и показывает, разрешены ли новые депозиты, что помогает контролировать эмиссию токена.
- **OwnershipTransferred:** событие передачи прав владельца контракта. Генерируется при вызове transferOwnership и фиксирует адреса предыдущего и нового владельца, обеспечивая прозрачность смены управления.

Отслеживая эти события через обозреватель блокчейна или логи, можно получить полную картину того, как токены перемещаются, распределяются, блокируются или уничтожаются в рамках контракта, что крайне важно для прозрачности и доверия к проекту.

Вывод

KokoWinToken – это многофункциональный смарт-контракт токена, сочетающий в себе сразу несколько возможностей:

- Стандартный токен TRC-20: обеспечивает базовый функционал (переводы, балансы, разрешения) на блокчейне TRON, что делает KOKO совместимым с широким спектром кошельков и бирж.
- Стейкинг: Встроенная возможность для держателей получать доход от своих токенов, стимулирующая долгосрочное удержание и участие в экосистеме.
- Заморозка токенов: Инструменты для временной блокировки токенов обеспечивают гибкость в распределении (вестинг, защита от мгновенной продажи крупного объёма и т.д.).
- Администрирование и безопасность: Возможность приостановки операций, защита от реентрантных атак, ограничение доступа к чувствительным функциям – всё это повышает общую устойчивость и безопасность контракта.
- Airdrop: Контракт готов к массовым раздачам токенов что упрощает его использование и администрирование.

- Прозрачность: Все ключевые действия журналируются через события, позволяя сообществу и аудиторам отслеживать деятельность контракта в реальном времени.

В целом, KokoWinToken предоставляет полноценную экосистему для управления токеном KOKO, объединяя экономические стимулы и меры безопасности. Такая комбинация функционала делает контракт гибким и надёжным инструментом для реализации целей сообщества и разработчиков.

Лицензия

Данный смарт-контракт распространяется на условиях лицензии **MIT License**. Это означает, что исходный код открыт и может свободно использоваться, копироваться, изменяться и распространяться, при условии сохранения уведомления об авторских правах и текста лицензии. Лицензия MIT предоставляет минимальные ограничения, позволяя максимально широко использовать контракт в любых целях.