IMe NFT

**struct** ConnectedBuyer {  
 **bool** state;  
 **address** rewardType;  
 **address** rewardReceiver;  
 **address** rewardSender;  
}  
  
**struct** NFTData {  
 **bool** manualConfirm;  
 **address** author;  
 **string** name;  
 **string** botAvatar;  
 **string** telegramAddress;  
}  
  
**mapping**(**uint256** => NFTData) **public** NFTDatas;  
**mapping**(**uint256** => **mapping**(**uint256** => ConnectedBuyer))  
 **public** ConnectedBuyers;

Каждая NFT содержит данные в виде структуры NFTData, в которую входят:

name – Наименование бота  
author – Автор бота  
manualConfirm – Необходимость ручного подтверждения покупателей  
botAvatar – Аватар бота  
telegramAddress – Адрес телеграм  
  
Также каждая NFT содержит маппинг Подключенных покупателей по их ID. Каждый подключенный покупатель представлен в виде структуры ConnectedBuyer, в которую входят:

state – Статус  
rewardType – Тип награды  
rewardReceiver – Адрес получающего награду  
rewardSender – Адрес передающего награду

Функции:

1. **function** mint(  
    **address** \_to,  
    **uint256** \_tokenId,  
    **string** calldata \_uri,  
    **bool** manualConfirm,  
    **address** author,  
    **string** calldata name,  
    **string** calldata botAvatar,  
    **string** calldata telegramAddress  
   ) **external** onlyOwner

Для минта NFT необходимо передать информацию, содержащуюся в структуре NFTData. Подключенные покупатели не добавляются

1. **function** editConnectedBuyer(  
    **uint256** tokenID,  
    **uint256** buyerID,  
    **bool** state,  
    **address** rewardType,  
    **address** rewardReceiver,  
    **address** rewardSender  
   ) **external** onlyOwner

Данная функция служит для добавления/изменения Подключенного покупателя конкретной NFT. Необходимо передать ID NFT, ID Подключенного покупателя, а также данные из структуры ConnectedBuyer.

Создает ивент ConnectedBuyerEdited

1. **function** editNFTData(  
    **uint256** tokenID,  
    **bool** manualConfirm,  
    **address** author,  
    **string** calldata name,  
    **string** calldata botAvatar,  
    **string** calldata telegramAddress  
   ) **external** onlyOwner

Данная функция служит для изменения конкретной NFT. Необходимо передать ID NFT, а также данные из структуры NFTData.

Создает ивент NFTDataEdited

1. **function** editURI(**uint256** tokenID, **string** calldata newURI)  
    **external** onlyOwner

Данная функция служит для изменения URI конкретной NFT.

IMeMarketPlace

**struct** Disposal {  
 **bytes32**[] tags;  
 **address** offeror;  
 **string** category;  
 **string** description;  
 **string** author;  
 **uint256** OLCFPrice;  
 **uint256** deadline;  
}

Структура Disposal содержит:

tags – теги  
category – категория  
description - описание  
OLCFPrice – стоимость в OLCF  
author – автор бота  
deadline – время окончания действия размещения  
offeror – адрес, разместивший NFT

В контракте указаны адреса контрактов LIME, OLCF и IMeNFT, адрес, на который поступают комиссии за размещение в LIME (LIMEFeeAddress).

offeredNFTS – Множество ID NFT доступных к покупке в данный момент.  
disposalByID – Маппинг по структур данных размещения (Disposal) по ID NFT.  
addressToIDs – Маппинг множеств ID NFT размещенных конкретным адресом.  
periodToPrice – Маппинг стоимостей периодов размещения.

Функции:

1. **function** offerBot(  
    **bytes32**[] calldata tags,  
    **uint256** period,  
    **string** calldata category,  
    **string** calldata description,  
    **string** calldata author,  
    **uint256** OLCFPrice,  
    **uint256** tokenID  
   ) **external**

Данная функция служит для размещения NFT на маркетплейсе. Перед вызовом необходимо сделать approve NFT и LIME. Period указывается в днях. Также передается остальная информация из Disposal и ID NFT.

1. **function** buyBot(**uint256** tokenID) **external**

Данная функция служит для покупки NFT с определенным tokenID. Необходимо чтобы данная NFT была размещена в данный момент и её deadline был меньше текущего времени.  
Необходимо сделать approve OLCF.

1. **function** removeOffer(**uint256** tokenID) **external**

Данная функция служит для удаления размещения определенной NFT по tokenID. Необходимо чтобы вызывающий эту транзакцию владел данным размещением.