NBU.sol contract updates

Contract was completely reworked. Clean ERC20 based on OpenZeppelin guidelines, with Pausable interface and feature to vest funds on accounts.

LockRewardsFactory.sol

Based on already audited RewardsFactory an upgraded LockRewardsFactory was developed. This updated version includes a new feature to froze stake amount (for lock period).

```
uint256 public lockDuration;
mapping(address => mapping(uint256 => uint256)) public stakeLocks;
mapping(address => mapping(uint256 => uint256)) public
stakeAmounts;
mapping(address => uint256) public stakeNonces;
```

Added function:

```
function stakeFor(uint256 amount, address user) external
nonReentrant updateReward(user)
function withdrawAndGetReward(uint256 nonce) public nonReentrant
updateReward(msg.sender)
```

Remove `exit` function.

Updated `withdraw` and `stake` functions.

Factory.sol

Split base fee (0.30%) in two parts: one half is sending to Nimbus referral program, another is leaving on pair contract.

Main code injection was made in `swap` function:

```
{
address referralProgram =
INimbusFactory(factory).nimbusReferralProgram();
if (amount0In > 0) {
    address _token0 = token0;
    uint refFee = amount0In.mul(3)/ 1997;
    _safeTransfer(_token0, referralProgram, refFee);
    INimbusReferralProgram(referralProgram).recordFee(_token0, to,
refFee);
}
if (amount1In > 0) {
    uint refFee = amount1In.mul(3) / 1997;
    address _token1 = token1;
    _safeTransfer(_token1, referralProgram, refFee);
    INimbusReferralProgram(referralProgram).recordFee(_token1, to,
refFee);
}
```

Also fee amount was changed from 30 to 15 in the following lines:

```
uint balance0Adjusted =
balance0.mul(10000).sub(amount0In.mul(15));
uint balance1Adjusted =
balance1.mul(10000).sub(amount1In.mul(15));
```

Also the following functions and state variables were added to NimbusFactory:

```
address public nimbusReferralProgram;
function setNimbusReferralProgram(address _nimbusReferralProgram)
external{}
```

New contracts:

NbuTokenExchanger.sol
Contract for exchanging already deployed NBU for the new.

NimbusInitialSale.sol
Initial sale of NBU tokens

NimbusReferralProgram.sol
Referral program of the Nimbus platform