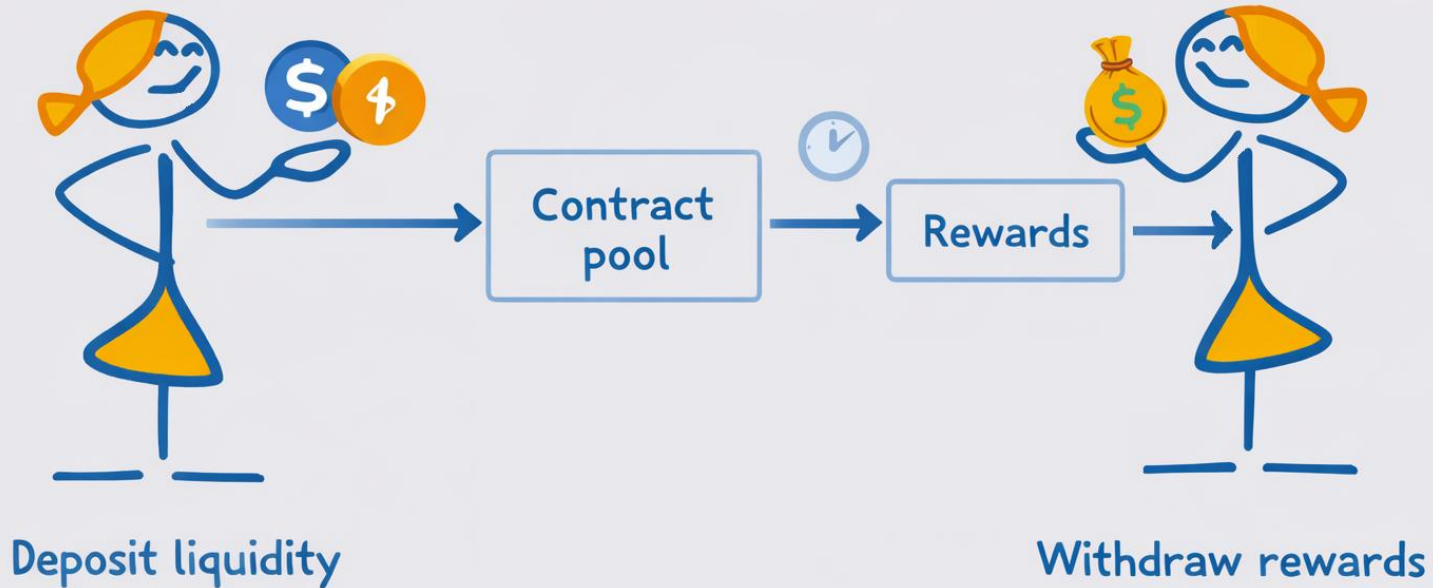


# FarmSwap: DEX with Yield Farming



Выполнили: Окунев Данила Игоревич, Лавицкая Александра Андреевна

## DEX with Yield Farming



# Функционал и реализация

```
// ===== EXTERNAL FUNCTIONS =====
```

```
/// @notice Adds Liquidity to the pool and mints LP tokens.
```

```
function addLiquidity(uint256 amountA, uint256 amountB) ...  
{ ...  
}
```

```
/// @notice Removes Liquidity from the pool and burns LP tokens.
```

```
function removeLiquidity(uint256 liquidity) ...  
{ ...  
}
```

```
/// @notice Swaps one token for another using the AMM formula.
```

```
function swap(address tokenIn, uint256 amountIn, uint256 minAmountOut) ...  
{ ...  
}
```

```
/// @notice Claims accumulated rewards for the caller.
```

```
function claimRewards() external nonReentrant updateReward(msg.sender) whenNotPaused {  
}
```

```
// ===== REWARD MANAGEMENT =====
```

```
/// @notice Funds the reward pool with additional tokens.
```

```
function fundRewards(uint256 amount) external onlyDistributor { ...  
}
```

```
/// @notice Sets a reward distributor status.
```

```
function setRewardDistributor(address distributor, bool status) external onlyOwner { ...  
}
```

```
/// @notice Sets a new reward rate.
```

```
function setRewardRate(uint256 newRate) external onlyOwner validRate(newRate) { ...  
}
```

```
// ===== VIEW FUNCTIONS =====
```

```
/// @notice Returns the earned rewards for an account.
```

```
function earned(address account) public view returns (uint256) { ...  
}
```

```
/// @notice Returns the current reward per LP token.
```

```
function getRewardPerLPToken() public view returns (uint256) { ...  
}
```

```
/// @notice Returns the current reserves of TOKEN_A and TOKEN_B.
```

```
function getReserves() public view returns (uint256, uint256) { ...  
}
```

```
/// @notice Returns maximum reward rate
```

```
function getMaxRewardRate() external pure returns (uint256) { ...  
}
```

```
// ===== INTERNAL FUNCTIONS =====
```

```
function _updateReward(address account) internal { ...  
}
```

```
function _updateRewardInternal() internal { ...  
}
```

```
function _onlyDistributor() internal view { ...  
}
```

```
function _updateUserReward(address account) internal { ...  
}
```

```
function _getReserves(address tokenIn) internal view returns (uint256 reserveIn, uint256 reserveOut)  
}
```

```
function _safeTransfer(IERC20 token, address to, uint256 amount) internal { ...  
}
```

```
function _safeTransferFrom(IERC20 token, address from, address to, uint256 amount) internal { ...  
}
```

```
function _sqrt(uint256 x) internal pure returns (uint256) { ...  
}
```

```
// ===== ADMIN FUNCTIONS =====
```

```
/// @notice Pauses the contract, preventing user interactions.
```

```
function pause() external onlyOwner { ...  
}
```

```
/// @notice Unpauses the contract, allowing user interactions.
```

```
function unpause() external onlyOwner { ...  
}
```

```
/// @notice Withdraws accumulated protocol fees.
```

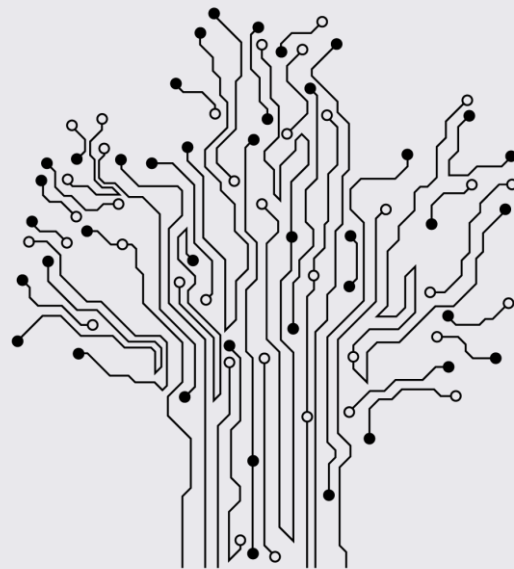
```
function withdrawProtocolFees() external onlyOwner { ...  
}
```

```
/// @notice Withdraws excess reward tokens (beyond allocated rewards).
```

```
function withdrawExcessRewards(uint256 amount) external onlyOwner { ...  
}
```

# Возможность развития

- Реальные LP токены вместо хранения ликвидности
- Оракулы для предсказания цены и защиты от атак
- Upgradeability (прокси контракт) + DAO (децентрализованное управление)



# Основные проблемы

- **Потеря точности:** При делении чисел без больших множителей ( $1e18$ ) сильно терялась точность
- **Неправильное начисление наград:** при вызове `claimRewards()` после внесения ликвидности когда контракт считал что пользователь вносил ликвидность с самого начала.
- **Утечка газа:** при исправлении предыдущей проблемы возникла ситуация, когда вызов почти каждого (даже `лёгкого`) метода вызывал повышенное потребление газа

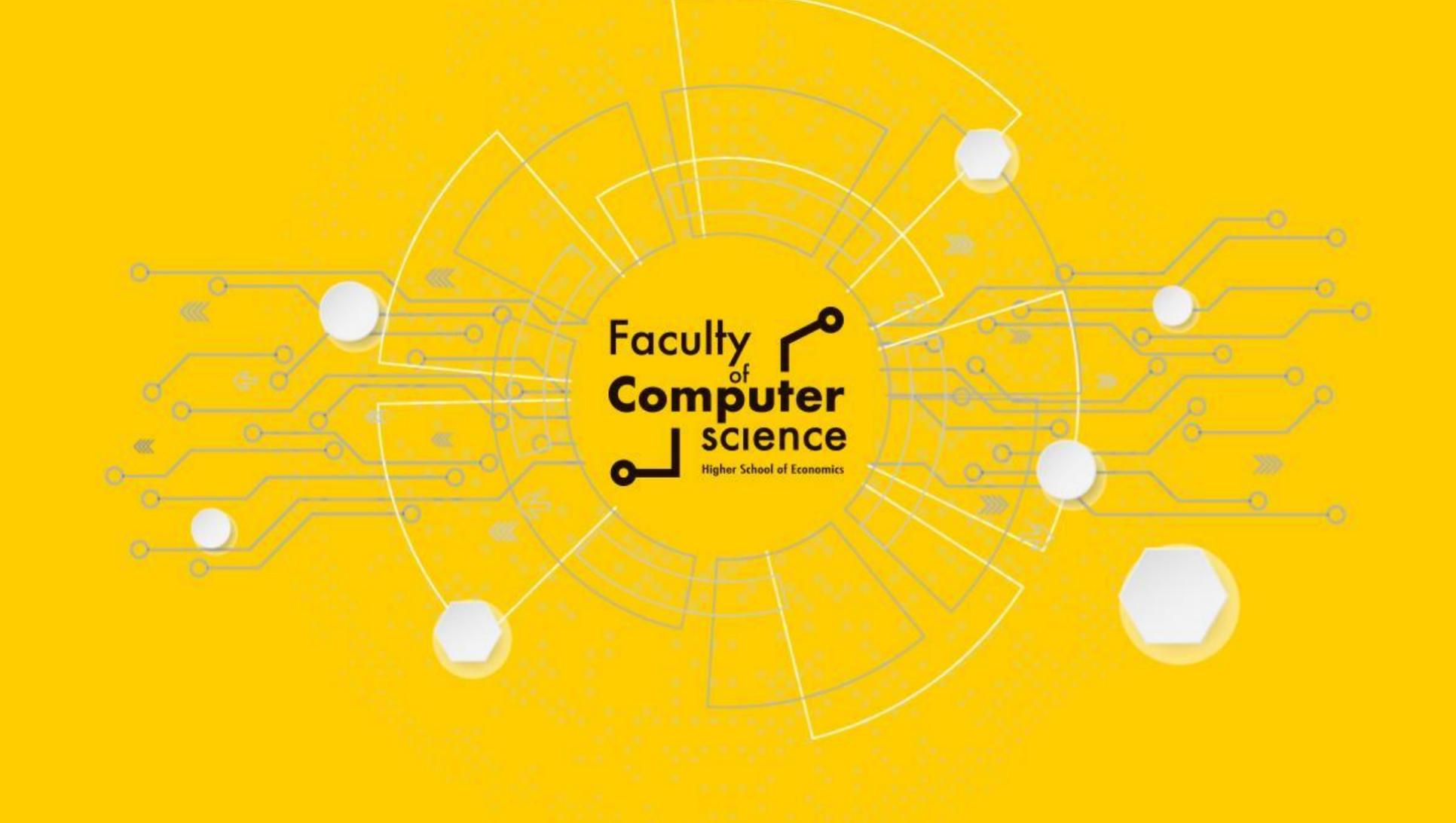
# Тестирование



Suite result: **ok**. 33 passed; 0 failed; 0 skipped; finished in 75.98ms (155.94ms CPU time)

Ran 1 test suite in 79.88ms (75.98ms CPU time): 33 tests passed, 0 failed, 0 skipped (33 total tests)

File	% Lines	% Statements	% Branches	% Funcs
contracts/FarmSwap.sol	93.02% (160/172)	92.90% (157/169)	67.69% (44/65)	100.00% (28/28)
contracts/test/ERC20Mock.sol	50.00% (2/4)	50.00% (1/2)	100.00% (0/0)	50.00% (1/2)
Total	92.05% (162/176)	92.40% (158/171)	67.69% (44/65)	96.67% (29/30)



Faculty  
of  
**Computer**  
science

Higher School of Economics