pragma solidity >=0.5.0 <0.6.0;

import "./zombiefeeding.sol";

contract ZombieHelper is ZombieFeeding {

// Modifier fonks: Bir java metodu, değişkeni ya da sınıf oluşturulurken bu öğelere kimlerin erişebileceğini belirtme olanağımız vardır. Bu eylemi gerçekleştirecek olan anahtar kelimelere Erişim Belirleyiciler(Access Modifiers) adını veririz.

modifier aboveLevel(uint \_level, uint \_zombieId) {

require(zombies[\_zombieId].level >= \_level);

\_;

}

function changeName(uint \_zombieId, string calldata \_newName) external aboveLevel(2 //isim değişmesi için lvl 2 olması lazım), \_zombieId) {

require(msg.sender == zombieToOwner[\_zombieId]);

zombies[\_zombieId].name = \_newName;

} // Not: çağrı verileri bir şekilde belleğe(Memory) benzer, ancak yalnızca harici(externall) işlevler için kullanılabilir.

+++son satırda dna değişmesi

function changeDna(uint \_zombieId, uint \_newDna) external aboveLevel(20, \_zombieId) {

require(msgsender. == zombieToOwner[\_zombieId]);

zombies[\_zombieId].dna =\_newDna;

}

// View işlemi dışardan çağrılırsa(external) herhangi bir işlem yapmadığı için ücretsiz olur.

function getZombiesByOwner (address \_owner) external view returns (uint [] memory) {uint[] memory result = new uint[](ownerZombieCount[\_owner]);

uint counter = 0;

for (uint i = 0; i < zombies.length; i++) {

if (zombieToOwner[i] == \_owner) {

result[counter] = i;

counter++;

}//zombilerimizi saymak için döngümüz

}

return result;

}

}