// SPDX-License-Identifier: MIT pragma solidity ^0.6.2; interface IUniswapV2Pair { event Approval(address indexed owner, address indexed spender, uint value); event Transfer(address indexed from, address indexed to, uint value); function name() external pure returns (string memory); function symbol() external pure returns (string memory); function decimals() external pure returns (uint8); function totalSupply() external view returns (uint); function balanceOf(address owner) external view returns (uint); function allowance(address owner, address spender) external view returns (uint); function approve(address spender, uint value) external returns (bool); function transfer(address to, uint value) external returns (bool); function transferFrom(address from, address to, uint value) external returns (bool); function DOMAIN\_SEPARATOR() external view returns (bytes32); function PERMIT\_TYPEHASH() external pure returns (bytes32); function nonces(address owner) external view returns (uint); function permit(address owner, address spender, uint value, uint deadline, uint8 v, bytes32 r, bytes32 s) external; event Mint(address indexed sender, uint amount0, uint amount1); event Burn(address indexed sender, uint amount0, uint amount1, address indexed to); event Swap( address indexed sender, uint amount0In, uint amount1In, uint amount0Out, uint amount1Out, address indexed to ); event Sync(uint112 reserve0, uint112 reserve1); function MINIMUM\_LIQUIDITY() external pure returns (uint); function factory() external view returns (address); function token0() external view returns (address); function token1() external view returns (address); function getReserves() external view returns (uint112 reserve0, uint112 reserve1, uint32 blockTimestampLast); function price0CumulativeLast() external view returns (uint); function price1CumulativeLast() external view returns (uint); function kLast() external view returns (uint); function mint(address to) external returns (uint liquidity); function burn(address to) external returns (uint amount0, uint amount1); function swap(uint amount0Out, uint amount1Out, address to, bytes calldata data) external; function skim(address to) external; function sync() external; function initialize(address, address) external; }