

```

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.18;


/// @title Simple Bank Account
/// @author
/// @notice Minimal contract that accepts deposits and allows the owner to withdraw Ether.
/// @dev Uses address(this).balance instead of a manual balance state variable.

contract BankAccount {

    /// @notice Owner of the contract (deployer)
    address public owner;


    /// @notice Emitted when a deposit is made
    /// @param account The address that sent Ether
    /// @param amount Amount of Wei deposited
    event Deposit(address indexed account, uint256 amount);


    /// @notice Emitted when a withdrawal is made
    /// @param account The address that received Ether (owner)
    /// @param amount Amount of Wei withdrawn
    event Withdraw(address indexed account, uint256 amount);


    /// @dev Restricts function access to the owner
    modifier onlyOwner() {
        require(msg.sender == owner, "Only owner");
        _;
    }


    /// @notice Set the deployer as the owner

```

```

constructor() {
    owner = msg.sender;
}

/// @notice Receive function to accept plain Ether transfers
/// @dev Emits a Deposit event
receive() external payable {
    emit Deposit(msg.sender, msg.value);
}

/// @notice Fallback in case of non-empty calldata sent with Ether
fallback() external payable {
    if (msg.value > 0) {
        emit Deposit(msg.sender, msg.value);
    }
}

/// @notice Deposit Ether into the contract
/// @dev `msg.value` is added automatically to the contract balance
function deposit() external payable {
    require(msg.value > 0, "Send some Ether");
    emit Deposit(msg.sender, msg.value);
}

/// @notice Owner-only withdrawal
/// @param _amount Amount in Wei to withdraw
/// @dev Checks the contract balance then uses call to send Ether
function withdraw(uint256 _amount) external onlyOwner {

```

```

require(_amount > 0, "Amount must be > 0");
require(_amount <= address(this).balance, "Insufficient contract balance");


// Use call to forward all gas and handle increased gas costs on recipient
(bool success, ) = payable(msg.sender).call{value: _amount}("");
require(success, "Ether transfer failed");


emit Withdraw(msg.sender, _amount);
}


/// @notice Returns the contract's Ether balance (in Wei)
/// @return The contract balance in Wei
function getBalance() external view returns (uint256) {
    return address(this).balance;
}
}

```