

```

1 // SPDX-License-Identifier: MIT
2 // 1. Pragma
3 pragma solidity ^0.8.0;
4 // 2. Imports
5 import "@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol";
6 import "../PriceConverter.sol";
7
8 // 3. Interfaces, Libraries, Contracts
9 error FundMe__NotOwner();
10
11 /**@title A sample Funding Contract
12  * @author Patrick Collins
13  * @notice This contract is for creating a sample funding contract
14  * @dev This implements price feeds as our library
15  */
16 contract FundMe {
17     // Type Declarations
18     using PriceConverter for uint256;
19
20     // State variables
21     uint256 public constant MINIMUM_USD = 50 * 10**18;
22     address private immutable i_owner;
23     address[] private s_funders;
24     mapping(address -> uint256) private s_addressToAmountFunded;
25     AggregatorV3Interface private s_priceFeed;
26
27     // Events (we have none!)
28
29     // Modifiers
30     modifier onlyOwner() {
31         // require(msg.sender == i_owner);
32         if (msg.sender != i_owner) revert FundMe__NotOwner();
33         _;
34     }
35
36     // Functions Order:
37     //// constructor

```