

This screenshot shows the VS Code editor with the `AggregatorV3Interface.sol` file open. The left sidebar displays the `FILE EXPLORERS` pane with a workspace named `default_workspace` containing files `AggregatorV3Interface.sol`, `IERC20.sol`, `ILendingPool.sol`, and `IWeth.sol`. The main editor area shows the following Solidity code:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.6.0;
3
4 interface AggregatorV3Interface {
5     function decimals() external view returns (uint8);
6
7     function description() external view returns (string memory);
8
9     function version() external view returns (uint256);
10
11     // getRoundData and latestRoundData should both raise "No data present"
12     // if they do not have data to report, instead of returning unset values
13     // which could be misinterpreted as actual reported values.
14     function getRoundData(uint80 _roundId)
15         external
16         view
17         returns (
18             uint80 roundId,
19             int256 answer,
20             uint256 startedAt,
21             uint256 updatedAt,
22             uint80 answeredInRound
23         );
24 }
```

This screenshot shows the VS Code editor with the `IERC20.sol` file open. The `FILE EXPLORERS` pane on the left shows the same workspace structure. The main editor area displays the `IERC20` interface code:

```
1 pragma solidity ^0.6.0;
2
3 interface IERC20 {
4     function allowance(address owner, address spender) external view returns (uint256 remaining);
5
6     function approve(address spender, uint256 value) external returns (bool success);
7
8     function balanceOf(address owner) external view returns (uint256 balance);
9
10    function decimals() external view returns (uint8 decimalPlaces);
11
12    function decreaseApproval(address spender, uint256 addedValue) external returns (bool success);
13
14    function increaseApproval(address spender, uint256 subtractedValue) external;
15
16    function name() external view returns (string memory tokenName);
17
18    function symbol() external view returns (string memory tokenSymbol);
19
20    function totalSupply() external view returns (uint256 totalTokensIssued);
21
22    function transfer(address to, uint256 value) external returns (bool success);
23
24    function transferFrom(
```

This screenshot shows the VS Code editor with the `ILendingPool.sol` file open. The `FILE EXPLORERS` pane on the left shows the workspace structure. The main editor area displays the `ILendingPool` interface code, including imports and a deposit event:

```
1 // SPDX-License-Identifier: agpl-3.0
2 pragma solidity 0.6.12;
3 pragma experimental ABIEncoderV2;
4
5 import {ILendingPoolAddressesProvider} from "@aave/protocol-v2/contracts/interfaces/ILendingPoolAddressesProvider.sol";
6 import {DataTypes} from "@aave/protocol-v2/contracts/protocol/libraries/types/DataTypes.sol";
7
8 interface ILendingPool {
9     /**
10      * @dev Emitted on deposit()
11      * @param reserve The address of the underlying asset of the reserve
12      * @param user The address initiating the deposit
13      * @param onBehalfOf The beneficiary of the deposit, receiving the aTokens
14      * @param amount The amount deposited
15      * @param referral the referral code used
16      */
17     event Deposit(
18         address indexed reserve,
19         address user,
20         address indexed onBehalfOf,
21         uint256 amount,
22         uint16 indexed referral
23     );
24 }
```