// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.8.0 <0.9.0;

contract BasicOperations {

uint256 public number;

uint public age = 25;

int public temperature = -10;

bytes public byteArray = "Hi, Solidity!";

bytes1 public byte1Var;

bytes2 public byte2Var;

enum Color { Red, Green, Blue }

Color public favoriteColor;

// Struct type

struct Person {

string name;

uint age;

}

Person public person;

string public text;

bool public flag;

constructor() {

number = 1 days;

text = "Hello, Solidity!";

byte1Var = 0x42;

byte2Var = 0x1234;

favoriteColor = Color.Blue;

person.name = "John Doe";

person.age = 30;

flag = true;

}

function performOperations(uint256 a, uint256 b) public pure returns (uint256[5] memory) {

uint256[5] memory results;

// Addition

results[0] = a + b;

// Subtraction

results[1] = a - b;

// Multiplication

results[2] = a \* b;

// Division

require(b != 0, "Division by zero");

results[3] = a / b;

// Modulus

require(b != 0, "Modulus by zero");

results[4] = a % b;

return results;

}

function concatenateText(string memory a, string memory b) public pure returns (string memory) {

return string(abi.encodePacked(a, b));

}

function toggleFlag() public {

flag = !flag;

}

}