# ClassCalculator

Сделала Буланина А.И. группа 195 дата: 09.09.2021

1. Цель: ЗАЧЕМ вы это делали? Написать модульный тест для функции проверки корректности калькулятора и добавить функцию Main, в которой вы производите вызов функции.
2. Задачи: ЧТО вы должны были сделать? Тестирование калькулятора

(classCalculator).

Доп. задание: Калькулятор выполняет минимум 3 дополнительных функции помимо сложения, вычитания, деления и умножения. Пример (квадратный корень, число в степени, факториал).

Доп. задание: добавить функцию Main, в которой вы производите вызов функции

1. Теория: что надо ЗНАТЬ, чтобы выполнить задачи? Название шаблона и как

его найти в visual studio; как создать тестовый проект для нужного метода; команды и их значение.

предусловие (блок Аrrange) — устанавливает начальные условия для выполнения теста

действие (блок Act) — выполняет сам тест

постусловие (блок Assert) — верифицирует результат теста, и, в данном случае, оформление — повышает читаемость текста и облегчает его использование в качестве документации к тестируемой функциональности

Из чего состоит метод :

переменная для установки значения переменных из тестовых данных (sum1, sum2)

переменная для установки математических знаков из тестовых данных (operation)

переменная для установки точного значения из тестовых данных (result)

переменная, которая вернет актуальный результат при выполнении метода CheckPassword. В нашем случае ValidatePassword

сравниваем два значения: ожидаемое и реальное, метод AreEquel, и в качестве аргумента — наши данные (result, actual)

Описание: КАК вы решили задачи?

1 Составить таблицу (она преставлена ниже)

|  |  |  |  |
| --- | --- | --- | --- |
| Действие | Числа | Результат | True/False |
| Сумма(+) | 4+6  4+6 | 10  6 | True  False |
| Разность(-) | 6-2  6-2 | 4  3 | True  False |
| Произведение(\*) | 2\*2  2\*2 | 4  3 | True  False |
| Частное(/) | 4/2  4/2 | 2  1 + исключение | True  False |
| Квадратный корень | 144  144 | 12  13 | True  False |
| Число в степени | 2  2 | 4  5 | True  False |
| Факториал числа | 2  2 | 2  5 | True  False |

2 Открываем Visual Studio, создаём новый проект. В поиске шаблонов вбиваем «Библиотека классов»

3 Выбрав шаблон, называем проект «ClassCalculator». Нажимаем создать.

4 Как только создали сразу переименовываем «Class1.cs» в «CalcChecker.cs»

5 Так же переименовываем namespase («ClassLibrary1» в «ClassCalculator») и public class (вроде должен переименоваться сам)

6 Затем создаём статичный метод double " ValidateCalculator " а в теле метода пока просто возвращаем true

public static bool ValidateCalculator(double sum1, double sum2, string operation) {return true;}

7 Создаём тестовый проект для этого метода (правой кнопкой мыши наводим на " CalcChecker " и выбираем "создание модульных тестов". После этого переходим в "CalcCheckerTests.cs", там в public class CalcCheckerTests создаём метод " public void Check\_Summ\_ReturnsTrue() ".

Перед методом пишем "[TestMethod()]", внутри метода пишем:

double sum1 = 6;

double sum2 = 4;

string operation = "+";

double result = 10;

double actual = CalcChecker.ValidateCalculator(sum1, sum2, operation);

Assert.AreEqual(result, actual);

8 Таким образом создаём все строчки таблицы. В итоге получается 7 методов

9 Теперь переходим обратно в CalcChecker.cs и в "public static bool ValidateCalculator(double sum1, double sum2, string operation)" создаём условия.

В итоге " public static double ValidateCalculator(double sum1, double sum2, string operation)" выглядит так

public static double ValidateCalculator(double nums, double num1, string operation)

{

if (operation == "+" || operation == "сложение")

{

return (nums + num1);

}

if (operation == "-" || operation == "вычитание")

{

return (nums - num1);

}

if (operation == "\*" || operation == "умножение")

{

return (nums \* num1);

}

if (operation == "/" || operation == "деление")

{

if (num1 == 0)

{

Console.WriteLine("Ошибка!На 0 делить нельзя!");

}

else

{

return (nums / num1);

}

}

if (operation == "sqrt" || operation == "корень")

{

if (nums > 0)

{

double sqrt = Math.Sqrt(nums);

return (sqrt);

}

else

{

Console.WriteLine("Ошибка! Нельзя искать квадратный корень отрицательного числа");

}

}

if (operation == "pow" || operation == "степень")

{

double pow = Math.Pow(nums, num1);

return (pow);

}

if (operation == "fact" || operation == "факториал")

{

for (int i = 1; nums >= i; ++i)

{

num1 = num1 \* i;

}

return (num1);

}

return 0;

}

10 В классе «CalcChecker» создаём метод main и пишем:

static void Main()

{

Console.WriteLine("Какую функцию калькулятора вы хотите сделать? Можно выбрать: сложение, вычитание, умножение, деление, квадратный корень из числа (корень), возведение в степень (степень), факториал");

string operation = Console.ReadLine();

Console.WriteLine("Введите 1 число");

double a = double.Parse(Console.ReadLine());

Console.WriteLine("Введите 2 число");

double b = double.Parse(Console.ReadLine());

double result = ValidateCalculator(a, b, operation);

Console.WriteLine("Результат " + result);

}