# **Отчёт по лабораторной работе № 4. Модульное тестирование**

Лабораторная работа выполнялась командой в количестве двух студентов группы ИТ-41: Климов Валентин, Юмагишиев Аскерхан. Использовались язык программирования C# для разработки и фреймворк NUnit для тестирования. В ходе выполнения работы каждый из участников выполнял свои задачи.

Для первого класса ведомым являлся Юмагишиев

Для второго класса ведомым являлся Климов

Достоинства роли ведущего в том, что тестировщик в целом не тратит много ресурсов на проверку работоспособности и есть возможность идти по тест-шаблонам.

Недостатки же в том, что если найдется баг, то нужно тщательно изучать ТЗ и понять алгоритм, который выстроил ведомый.

И этот же недостаток является преимуществом ведомого.

Недостатки: Достаточно много времени для изучения документации о способах выполнения данной задачи.

Ролями не менялись, но Валентин разработал несколько методов из класса Аскерхана.

Программа состоит из двух классов, по условию. Для удобной работы с методами этих классов были одновременно описаны два оператора swith case. Программа в самом начале запрашивает у пользователя ввести 2 переменные строкового типа (числа a и b), с которыми в дальнейшем будут производиться вычисления. После этого в консоли выводится список методов 1 класса, в конце которого возможность перейти к оператору swith case второго класса.1 класс (Myint). Большинство методов данного класса подобны привычным операциям сложения, вычитания, умножения, деления в столбик (на листочке).

Метод Add (сложение):

В метод передаются введенные переменные a и b. Для удобства переменные реверсируются, программа меняет переменные местами таким образом, чтобы на первом место стояла та, в которой длина больше. Элементы (цифры) переменных поразрядно складываются при помощи стандартной операции сложения (+), после чего к результирующей переменной строкового типа прибавляется сумма текущих цифр разряда (если она больше 9, десяток переходит в следующий разряд, а прибавляется результат деления на 10 оператором (% остаток от деления). Таким образом, результат сложения двех строковых переменных записан задом наперед в результирующую переменную. Метод реверсирует ее и возвращает (return result;).

Метод вычитания (Substract):

Данный метод аналогичен методу сложения, но в самом начале для определения бОльшей переменной используется метод сравнения (CompareTo). Также отличие в том, что происходит поразрядное вычитание элементов переменных стандартным методом вычитания (-), и в случае, если вычитаемое больше вы уменьшаемого, к уменьшаему предворительно прибавляется 10, а из следующего разряда вычитается единица в следующем шаге цикла при помощи "запоминающей" булевой переменной.

Метод умножения (Multiplay):

Метод аналогичен методу сложения, но здесь происходит умножение сначала первого разряда второй переменной на каждый разряд первой, и так далее при помощи двумерного цикла, и вместо применения операции сложения применяется операция умножения (\*), после чего больший разряд присваивается одной переменной, а меньший прибавляется к результату result. Также после окончания перемножения одного разряда второй переменной на каждый разряд первого, к результату применяется метод сложения результирующей переменной result с переменной, в которую записывался результат перемножения (учитывается, что при умножении, например, второго разряда второй переменной к результату перемножения в конец прибавляется "0".

Метод деления (Divide):

Метод подразумевает из себя деление нацело. В дальнейшем он используется для метода нахождения наибольшего общего делителя (Gcd). Метод также аналогичен методу деления в столбик. В уменьшаемом выбирается определённое количество элементов, начиная с начала, равное вычитаемому (либо больше на 1 элемент, если при использовании метода CompareTo вычитаемое оказалось больше). Из выбранного количества элементов методом Substract поочередно вычитается уменьшаемое до тех пор, пока оно не станет равным 0 или меньше вычитаемого, при этом фиксируется количество вычитаний. Из уменьшаемого удаляеются первые элементы, с которыми мы работали, а остаток прибавляется к следующему элементу методом сложения. Далее в результат прибавляется это значение (количество вычитаний) и цикл повторяется (выбирается определённое количество элементов, начиная с начала, равное вычитаемому..).

Метод нахождения максимума (Max):

Вызывается метод сравнение (CompareTo), в результат записывается большее.

Метод нахождения минимума (Min):

Вызывается метод сравнение (CompareTo), в результат записывается меньшее.

Метод нахождения абсолютного значения (Abs):

Если в переменной в начале минус “-”, он удаляется и полученное значение возвращается.

Метод сравнения (CompareTo):

Сравнение происходит в несколько этапов. Сначала переменные проверяются на наличие минуса, и в зависимости от этого результат уже может быть известен. Следующим этапом числа проверяются на количество элементов в них, и если предыдущий этап не дал ответ, бОльшим становится то, у которого больше элементов (если у обоих не блыо знака минус, и наоборот в противном случае). Следующий этап наступает, если оба числа с одинаковым знаком и у обоих равное количество элементов. Элементы поразрядно сравниваются, и как только один элемент больше другого, программа возвращает 2 значения (первое – это знак равенства, второе - один из 4х выриантов (0 - все числа положительные, 1 - число a отрицательное, 2 - число b отрицательное, 3 - оба числа отрицательные))

Метод нахождения наибольшего общего делителя (Gcd):

Сначала первое число делится на минимально возможное целое число, чтобы не было остатка при помощи метода деления. То число, на которое поделилась переменная, записывается в список 1. Дальше этой переменной присваивается результат деления, и операция выполняется до тех пор, пока переменная не станет равной 1. Тоже самое происходит и со второй переменной. Далее одинаковые элементы Списков 1 и 2 умножатся (например, в списках есть одинаковый множитель 7, тогда результат умножается на 7). Таким образом находится наибольший общий делитель.

Матод преобразования в строку (ToString):

Здесь используется стандартный метод string result = new string(str);

Метод преобразования в число типа long (LongValue):

Реализовано с помощью оператора try catch. В блоке try результату присваивается переменная, а в блоке catch удаляется последний символ строки, затем цикл while повторяется, и это происходит до тех пор, пока блок try не выполнится.

2 класс (DEQueue)

Метод добавления элемента в конец (PushBack):

К исходной переменной строкового типа прибавляю элемент, который вводится с клавиатуры.

Метод добавления элемента в начало (PushFront):

К элементу, который вводится с клавиатуры, исходной прибавляется исходная строковая переменная.

Метод извлечения элемента с конца (PopBack):

return str[str.Length - 1];

Метод извлечения элемента с начала (PopFront):

Аналогично предыдущему.

Метод просмотра последнего элемента (Back):

Аналогично предыдущему.

Метод просмотра первого элемента (Front):

Аналогично предыдущему.

Метод вычисления количества элементов (Size):

Стандартный метод return str.Length;

Метод очистки очереди (Clear):

Присвоение переменной пустого значения “”.

Метод преобразование в массив (ToArray):

Стандартный метод return str.ToCharArray();