**Требования**

1. ~~округление результата до 4-х символов;~~
2. ~~интерактивное меню с выбором режима работы:~~
   1. ~~поочерёдный ввод чисел и операции с последующим вычислением:~~
      1. ~~ввод первого числа;~~
      2. ~~ввод второго числа;~~
      3. ~~вывод результата;~~
   2. ввод выражения целиком одной строкой с последующим вычислением;
3. генерация определённых чисел в виде расширений к типу int:
   1. положительные;
   2. отрицательные;
   3. чётные;
   4. нечётные;
4. ~~вычислительная логика распределена в классе Calculator. При инициализации данного класса требуется получить объект логгера, ведущего записи (может быть null). При наличии данного объекта в состоянии необходима выводить через него сообщение с названием метода, вызванного для выполнения операции калькулятора. Логгер по умолчанию просто должен иметь метод Write(string message), который будет выводить сообщение в консоль. Если в конструкторе класса передан логгер null, то записи сообщений не будет;~~
5. добавить в калькулятор возможность поиска максимального отрицательного и минимального положительного значения массива. По кнопке 1 происходит запуск калькулятора, по кнопке 2 запускается обработчик массивов.

**Пример работы приложения в режиме поочерёдного ввода двух чисел и операции**

Отслеживать выполняемые методы? (y/n)

y

Введите число…

58

Введите второе число

730

Введи операцию (*список возможных операций*)

-

Выполняется метод Subtract

Результат: -672

**Пример работы приложения в режиме работы с целым выражением**

Введите выражение...

(7 / 3) \* 100 – 71

Операция выполним. Идёт расчёт…

Результат: 929.

**Пример работы приложения с логгером**

Хотите ли отслеживать выполняемые методы? y/n

Y

Методы будут отслеживаться.

Введите число…

58

Введите ещё число…

730

Введите знак операции

Ч

Нет такой операции, введите допустимый знак операции…

-

Выполняется метод **Subtract** класса **Calculator**

Результат: -672