

## Практическая работа № 9. Отладка кода.

Предоставлять выполненную работу необходимо в виде отчета, **даже если вы делали задание на паре.**

Дана программа, выполняющая некоторые функции, но имеющая недочеты, информация представлена ниже:

1. Калькулятор: выполняет различные математические операции, но имеет ошибки при работе с делением на ноль, возведением в степень отрицательных чисел и операциями с дробными числами.
2. Поиск элементов в массиве: поиск минимального, максимального значений и среднего по массиву, но в коде содержатся логические ошибки и недочёты.
3. Функция поиска элемента по значению в коллекции `List<int>`: предназначена для поиска, но результаты получаются неверными.
4. Проверка числа на простоту: код содержит ошибку в логике проверки, приводящую к неверному определению простоты числа.
5. Рекурсивная функция для расчета факториала: приводит к переполнению стека на больших значениях ввода из-за отсутствия ограничения на глубину рекурсии.
6. Генерация списка случайных чисел: функция формирует список случайных чисел, но допускает повторения и неверный диапазон значений.

**На основе представленных данных необходимо сделать следующее:**

### 1. Отладка калькулятора:

- Функция калькулятора выполняет основные операции: сложение, вычитание, умножение, деление и возведение в степень. В коде возникают ошибки при делении на ноль, обработке дробных чисел и возведении отрицательных чисел в степень.
- **Требуется:** найти все ошибки, установить точки останова перед каждой операцией и выполнить отладку с заходом (Step Into) и обходом (Step Over), анализируя промежуточные значения переменных.
- **Подсказка:** проверьте, какие результаты выводятся на консоль при выполнении каждой операции, включая сложение и вычитание дробных чисел.

### 2. Работа с массивом:

- Реализован поиск минимального и максимального значений массива, а также расчёт среднего. В коде есть ошибки в логике вычислений, которые приводят к неверному результату при пустом массиве или массиве с одним элементом.
- **Требуется:** отладить каждый этап вычисления, установить точки останова для отладки логики работы с массивом. Протестируйте функции на массивах с разным количеством элементов и значениями.
- **Подсказка:** особое внимание уделите обработке массива с одним элементом и пустого массива, проверяя промежуточные значения через точки останова.

### 3. Поиск элемента в коллекции:

- Программа должна находить определенное число в коллекции `List<int>`, но из-за ошибки в логике поиска неверно выводит результаты.
- **Требуется:** установить точки останова на ключевых операциях поиска и следить за порядком выполнения. Используйте шаг с заходом для проверки значений элементов на каждом этапе поиска.
- **Подсказка:** следите за значением переменной цикла и проверяйте, как меняется индекс при поиске.

### 4. Проверка, является ли число простым:

- Функция проверки числа на простоту содержит логические ошибки, которые приводят к неверному результату.
- **Требуется:** используя пошаговую отладку, найти участок, где происходит ошибка, и выявить некорректные значения переменных.
- **Подсказка:** установите точки останова на проверке делимости числа, шаг за шагом анализируя выполнение условий.

### 5. Рекурсивная функция факториала:

- Функция факториала реализована рекурсивно и приводит к переполнению стека при больших значениях ввода.
- **Требуется:** проследить шаги рекурсивного вызова функции, установить точку останова перед рекурсивным вызовом, используя шаг с заходом, чтобы увидеть значения переменной на каждом уровне рекурсии.
- **Ошибка:** при больших значениях  $n$  метод вызывает переполнение стека из-за глубокой рекурсии (возможное исправление: можно изменить рекурсивную реализацию на итеративную или добавить ограничение на входное значение  $n$ , чтобы избежать переполнения).

- **Подсказка:** обратите внимание на значение переменной, которая передается в рекурсию, чтобы понять, когда происходит переполнение.

#### 6. Генерация случайных чисел:

- Функция генерирует случайные числа и добавляет их в список `List<int>`, но допускает повторения и значения вне заданного диапазона.
- **Требуется:** установить точки останова на генерации чисел и проверять диапазон каждого сгенерированного числа. Пошагово следите за добавлением значений в список и наблюдайте за ошибками в диапазоне.
- **Подсказка:** проверьте, почему в список добавляются повторяющиеся значения, и проследите за значением диапазона для каждого нового числа.

Что необходимо сделать по итогу:

- **Отчет об отладке:** детальное описание выявленных ошибок и шагов по их устранению.
- **Отредактированный код:** исправленный код с добавлением необходимых исправлений и улучшений.
- **Скриншоты отладки:** скриншоты с точками останова, демонстрирующие значения переменных и шаги выполнения кода.