**Отладка проекта в Visual Studio 2019**

**Цель работы**

Закрепление практических навыков работы с системой Visual Studio 2019, изучение инструментов для проведения отладки приложений.

**Ход работы**

**Шаг 1. Разработка проекта**

Создадим простую программу на C#, которая использует цикл и включает вычисление сложной функции. Для примера возьмем задачу вычисления факториала.

csharp

Копировать код

using System;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите число для вычисления факториала:");

int number = int.Parse(Console.ReadLine());

int result = CalculateFactorial(number);

Console.WriteLine($"Факториал числа {number} равен {result}");

}

static int CalculateFactorial(int n)

{

if (n < 0)

throw new ArgumentException("Число должно быть неотрицательным.");

int factorial = 1;

for (int i = 1; i <= n; i++)

{

factorial \*= i;

}

return factorial;

}

}

**Шаг 2. Запуск программы в режиме отладки**

1. Нажмите клавишу **F5** или кнопку **Start** в Visual Studio, чтобы запустить программу.
2. Вводите положительное число и проверяйте результат.

**Шаг 3. Установка точек останова**

1. Установите точки останова (breakpoints) в следующих строках:
   * В строке вызова метода CalculateFactorial.
   * Внутри цикла for.
2. Для установки точки останова щелкните слева от строки кода или нажмите **F9**.

**Шаг 4. Изучение значений переменных**

1. После остановки выполнения программы на точке останова наведите курсор на переменные, такие как i и factorial, чтобы просмотреть их значения.
2. Также используйте окно **Locals** для просмотра всех локальных переменных.

**Шаг 5. Выполнение программы пошагово**

1. Нажмите клавишу **F10** для выполнения программы по шагам (шаг без входа в метод).
2. Нажмите **F11**, чтобы перейти внутрь метода CalculateFactorial.

**Шаг 6. Условные точки останова**

1. Щелкните правой кнопкой на установленной точке останова и выберите **Conditions**.
2. Установите условие, например, i == 5, чтобы выполнение останавливалось только при достижении значения переменной i = 5.

**Шаг 7. Изменение значения переменной**

1. Приостановите выполнение программы.
2. В окне **Watch** добавьте переменную factorial.
3. Измените значение переменной вручную, например, установив factorial = 100.

**Шаг 8. Работа с исключениями**

1. Запустите программу и введите отрицательное число, например -5.
2. Программа сгенерирует исключение ArgumentException.
3. Перейдите в меню **Debug > Exceptions**, настройте остановку выполнения программы при возникновении исключения.

**Вывод**

В ходе лабораторной работы была создана программа для вычисления факториала, а также отработаны навыки использования инструментов отладки в Visual Studio:

* Установлены точки останова.
* Изучены значения переменных.
* Использованы условные точки останова.
* Исследовано поведение программы при возникновении исключений.

**Контрольные вопросы**

1. **Что такое отладка?** Отладка — процесс поиска и устранения ошибок в программном коде. Она включает использование инструментов для наблюдения за выполнением программы и анализом её состояния.
2. **Какие инструменты отладки вам известны?**
   * Точки останова (Breakpoints).
   * Окна наблюдения за переменными: Autos, Locals, Watch.
   * Изменение значений переменных во время выполнения.
   * Инструменты для анализа исключений.
3. **Методы отладки**
   * Ручная отладка (с использованием точек останова и пошагового выполнения).
   * Логирование (запись выполнения программы в лог-файлы).
   * Использование специальных инструментов, таких как профилировщики и анализаторы памяти.