**<https://github.com/07Rinat07>**

**Контрольные вопросы**

**1. Что такое отладка?  
Отладка – это процесс поиска, анализа и исправления ошибок в программном коде.**

**2. Какие инструменты отладки вам известны?**

* **Точки останова (обычные и условные).**
* **Окна Autos, Locals, Watch для отслеживания переменных.**
* **Отладка исключений.**
* **Пошаговое выполнение кода (Step Into, Step Over, Step Out).**
* **Журнал отладчика и трассировка (Debug Output).**

**3. Методы отладки.**

* **Использование точек останова и просмотра значений переменных.**
* **Логирование (вывод информации о ходе работы программы).**
* **Использование отладочных конфигураций (Debug).**
* **Анализ исключений и ошибок времени выполнения.**
* **Пошаговое выполнение кода с анализом состояния программы.**

**Ход работы**

**1. Создание программы.  
Для примера разработана простая программа на языке C#, вычисляющая факториал числа с использованием цикла for.  
Код программы:**

**using System;**

**namespace DebugExample**

**{**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**Console.Write("Введите число: ");**

**int n = int.Parse(Console.ReadLine());**

**long factorial = 1;**

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

**{**

**factorial \*= i;**

**Console.WriteLine($"Итерация {i}: факториал = {factorial}");**

**}**

**Console.WriteLine($"Факториал числа {n} равен {factorial}");**

**Console.ReadKey();**

**}**

**}**

**}**

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

* **В Visual Studio установлены точки останова на строках с вычислениями (factorial \*= i;) и выводом результата.**
* **При достижении точки останова выполнение программы приостанавливается, и управление передается отладчику.**

**3. Использование окна просмотра (Autos, Locals, Watch).**

* **В окне Autos отображаются значения переменных, к которым был недавний доступ.**
* **В окне Locals показаны все локальные переменные метода Main.**
* **В окно Watch вручную добавлены переменные n, i, factorial для наблюдения за изменением их значений.**

**4. Работа с контрольными точками (Breakpoints).**

* **Добавлены несколько точек останова внутри цикла для анализа промежуточных значений.**
* **Для отладки цикла настроена условная точка останова: программа останавливается только при i = 5.**

**5. Просмотр и изменение переменных во время отладки.**

* **В процессе выполнения программы через окно Watch изменено значение переменной n.**
* **Это позволило проверить, как программа ведет себя при изменении исходных данных без перезапуска.**

**6. Отладка с использованием исключений.**

* **В меню Debug → Windows → Exception Settings добавлен контроль исключений.**
* **При вводе некорректного значения (например, буквы вместо числа) отладчик перехватывает исключение FormatException.**

**7. Дополнительные возможности.**

* **Выполнено пошаговое выполнение программы (Step Into, Step Over, Step Out).**
* **Проверена работа приложения в конфигурациях Debug и Release.**