**1 Отчёт о выполненной работе**

**1.1 Изучение отладчика интегрированной среды разработчика**

1. Создайте шаблон приложения, следуя инструкциям из п 1 дополнительных сведений.



Рисунок 1 - Создание пользовательского шаблона

2. Используя ранее созданный шаблон (п 2. дополнительных сведений), создайте приложение (консольное или WPF на выбор) со следующей функциональностью:

a. Метод принимает на вход произвольную строку

b. Метод на выход подаёт строку с измененным регистром каждого символа ( Hello - > hELLO)

c. Обязательно использовать циклы, методы для работы с символами и любые вспомогательные конструкции по желанию

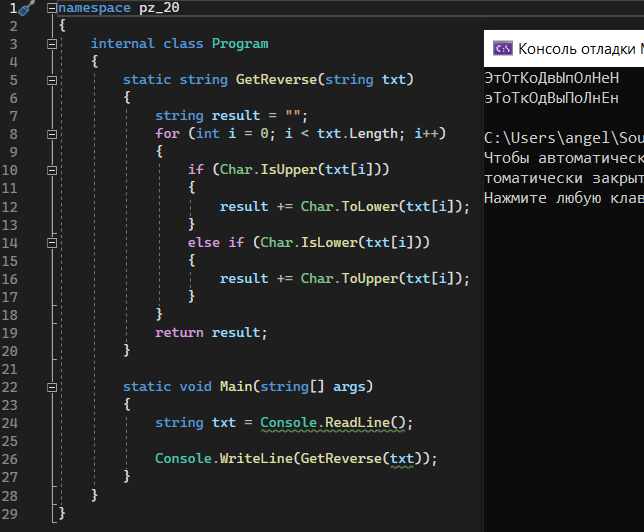


Рисунок 2 - Консольное приложение

1. Задайте значок приложения в свойствах проекта

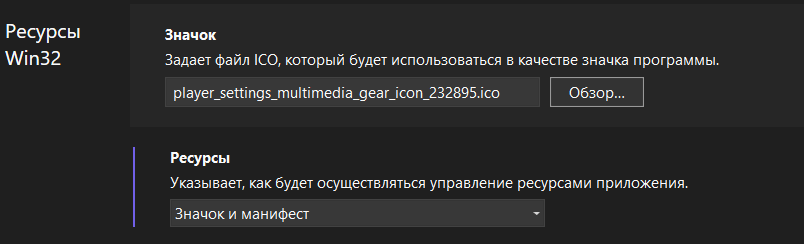


Рисунок 3 - Установка значка приложения

4. Произведите отладку кода с использованием точек останова:

a. Установите точку останова внутри метода на строке итерации цикла. Запустите стандартную отладку (F5). Проверьте содержимое счетчика цикла.

b. Установите точку останова на строке итерации цикла. Запустите отладку (Шаг с заходом), проверьте содержимое счетчика цикла.

c. Сделайте выводы по 2-м последним пунктам.

d. Используйте дополнительные свойства отладки (на выбор) для наблюдения за поведением кода

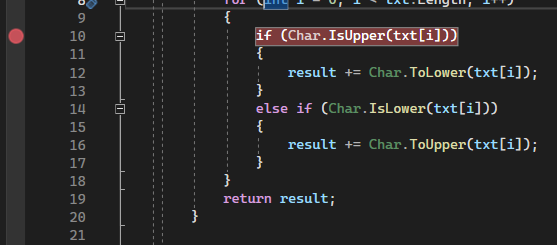


Рисунок 4 - Точка останова на строке итерации цикла

Вывод: При запуске стандартной отладки, с помощью которой мы можем контролировать ход действия программы, значения ее переменных и ход изменения данных в этих переменных, счетчик цикла меняет свое значение; Запуск отладки Шаг с заходом определяет работу отладки так что, команда в программе выполняет очередное действие, в отличии от стандартного запуска отладки, где мы перемещаемся по коду, не затрягивая все строчки а потом приостанавливает процесс, чтобы с помощью отладчика было можно проверить состояние программы на данном шаге.

5. Проверьте изменения значений переменных при отладке в окне Видимые

и Локальные (п.5), сделайте выводы

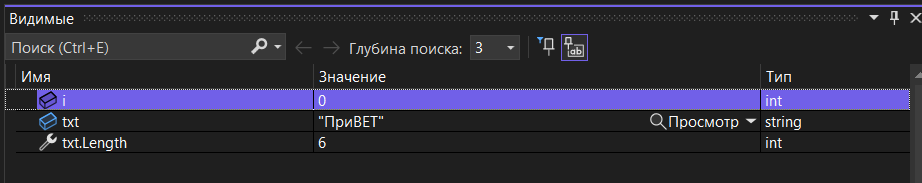


Рисунок 5 - Окно «Видимые»

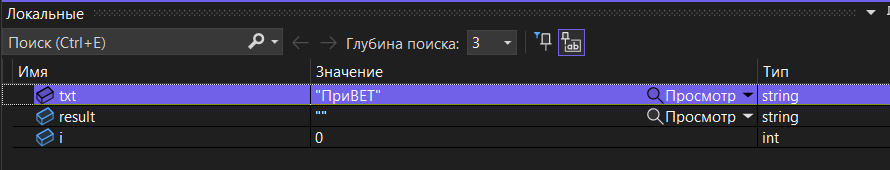


Рисунок 6 - Окно «Локальные»

Вывод: В окне «Видимые» представлены переменные, находящиеся непосредственно в том месте где стоит точка останова, например на рисунке 5 точка останова установлено на месте прохода по массиву через цикл for и в окне представлены значения внесенные в этот цикл. В окне «Локальные» же представленные все переменные, используемые локально в созданном методе.

1. Пронаблюдайте за состоянием стека вызовов при отладке и сделайте вывод.

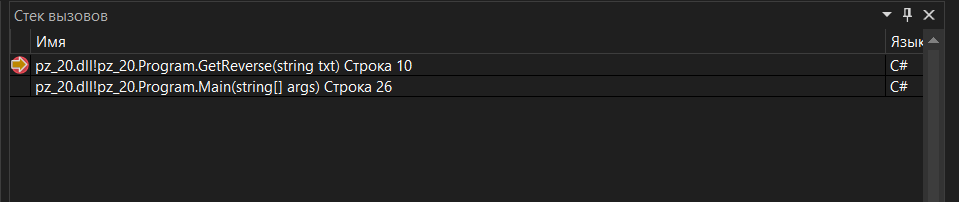


Рисунок 7 - Окно «Стек вызовов»

Вывод: «Стек вызовов» позволяет отслеживать место где находится точка останова и где сейчас идет выполнение кода, а также куда каждая из вызванных процедур должна вернуть управление после своего завершения.

1. Добавьте в проект код, вызывающий исключение (например выход за пределы строки) проанализируйте сообщение об исключении (п.4) – вывод оформить в отчете

Вывод: При возникновении исключения программа самостоятельно при режиме отладки выдаст всплывающее окно рядом с местом данной ошибки, где будет прописана что это за ошибка и информация о ней, из за чего она была вызвана и само описание этой ошибки.

8. Добавьте обработку исключения и приложите скриншот результата с выводом.

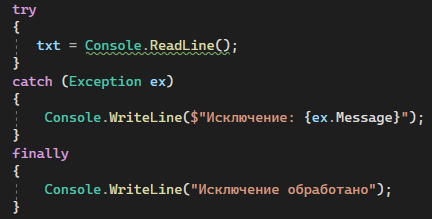


Рисунок 8 - Исключение

Было добавлено исключение класса Exception, чтобы получить информацию в случае возникновения исключения, а если все нормально, то просто вывод сообщения об обработки исключения.