# **Отчёт по лабораторной работе № 1. Парное программирование: Лабиринт**

Лабораторная работа выполнялась командой в количестве двух студентов группы ИТ-41: Климов Валентин, Юмагишиев Аскерхан. Использовались язык программирования C# для разработки и фреймворк NUnit для тестирования. В ходе выполнения работы каждый из участников выполнял свои задачи.

Ведомым программистом являлся Климов.

Ведущим программистом являлся Юмагишиев.

В процессе разработки мы менялись ролями. Достоинства роли ведущего в том, что тестировщик в целом не тратит много ресурсов на проверку работоспособности и есть возможность идти по тест-шаблонам.

Недостатки же в том, что если найдется баг, то нужно тщательно изучать ТЗ и понять алгоритм, который выстроил ведомый.

И этот же недостаток является преимуществом ведомого.

Недостатки: достаточно много времени для изучения документации о способах выполнения данной задачи.

По условию задачи необходимо разработать «идеальный» лабиринт. Он считается таковым если:

1. Он представляет собой прямоугольную сетку комнат, размером R рядов на C столбцов.
2. Лабиринт имеет две двери наружу: вход и выход. Вход всегда размещается на северной стене, выход может быть на любой стене.
3. Между любыми двумя комнатами в лабиринте существует ровно один маршрут.

Нам необходимо пройти алгоритм, используя алгоритм «всегда поворачивать налево», в котором при любой возможной развилке мы выбираем самое левое ответвление. Если упёрлись в тупик, то дважды поворачиваем направо (разворачиваетесь на 180 градусов) и продолжаем движение. Пройдя лабиринт, мы проходим его еще раз (по тому же алгоритму с поворотом налево), но в этот раз начав с выхода и закончив на входе.

Путь через лабиринт описывается тремя символами: ‘W’ означает движение вперед в следующую комнату, ‘L’ означает, что поворот налево на 90 градусов, и ‘R’ означает, поворот направо на 90 градусов. Начинаетя путь снаружи лабиринта, на клетке прямо перед входом, лицом ко входу. А заканчивается, когда выходим из лабиринта через выход.

Программа должна считывать данные из текстового файла. Первая строка файла содержит целое число N, описывающее количество заданий в файле. Далее следует N строк, описывающих прямое и обратное прохождения лабиринтов в формате: прямое\_прохождение обратное\_прохождение

Все пути состоят как минимум из двух символов, содержат только символы ‘W’, ‘L’, ‘R’, начинаются и заканчиваются символом ‘W’.

Программа должна выводить результаты в текстовый файл.

Для каждого задания (пары строк с прямым и обратным прохождением) выведите строку ‘Case #x’, где x – номер задания. Следующие R строк файла должны описывать лабиринт размером R на C. В каждой из строк должно быть C символов, описывающих, в каких направлениях можно двигаться из этой комнаты.

Программа выполняется в 1 классе Program.cs, ниже представлено начало класса Program.cs. Сначала определяется положение наблюдателя position, width. Затем направление 0 - North, 1 - South, 2 - West, 3 - East. Текущая позиция записывается в extremeWidth и extremeHeight.



Рисунок 1 – Листинг кода класса Program.cs

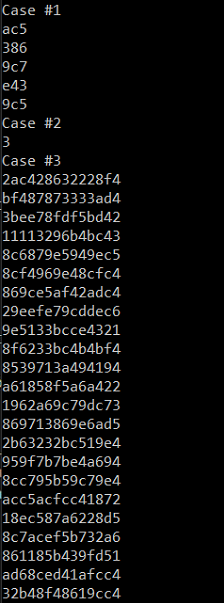


Рисунок 2 – Результат выполнения программой первых 3 кейсов



Рисунок 3 – Результат выполнения программой последнего кейса

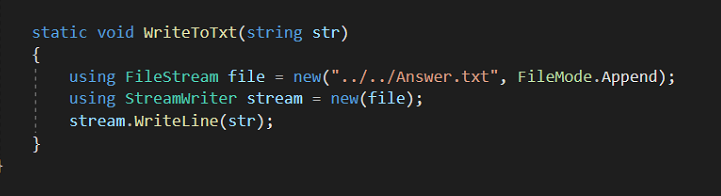


Рисунок 4 – Метод WriteToTxt

Метод WriteToTxt использовался для записи ответа в отдельный файл.

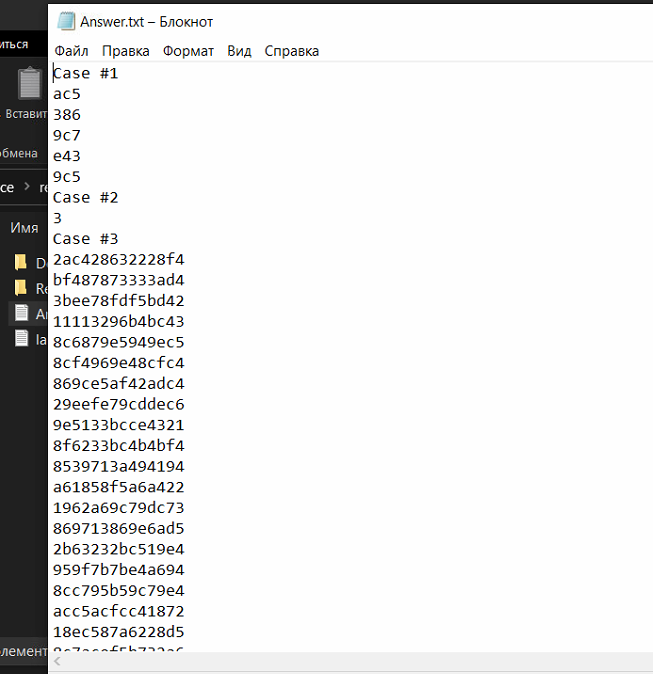


Рисунок 5 – Файл Answer.txt сформированный методом WriteToTxt

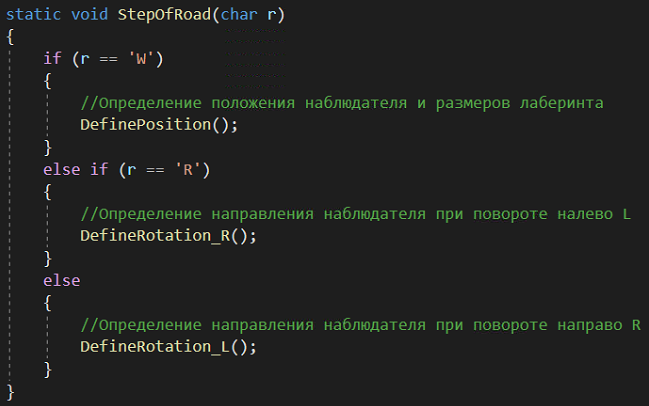


Рисунок 6 – Метод StepOfRoad