Проект 2

Программа делится на 4 части: проверка корректности игрового поля, проверка попадания/промаха игрока и корректности введенных координат, «Убийство» корабля и расстановка промахов на соседних с ним клеток, Основа, объединяющая части, выводящая поле и определяющая победителя.

Для хранения данных о поле имеется матрица 10x10 из integer

Для хранения местоположения кораблей используется массив на 10 элементов из string(12 символов макс), в каждой строке идут координаты, принадлежащие кораблю.сокращенно: матрица кораблей.

Координа записываются как 'символ с кодом, равным номеру строки матрицы', 'символ с кодом, равным номеру столбца матрицы', 'пробел'.

1.Программа создает игровое поле

1a.*Ввод матрицы из файла: М-море, К-корабль, 4 однопалубных, 3 двупалубных, 2 трехпалбных, 1 четырпалубный.*

Подпрограмма должна записать поле в матрицу, где 0 – море, -1 – корабль. Если в файле меньше ячеек чем надо – заполнить оставшиеся нолями.

*На выходе: Матрица 10x10, где 0 – море, -1 – корабль.*

1б. Проверка корректности Correct (изменяет исходную матрицу, создает массив кораблей, возвращает переменную Boolean, что матрица корректна)

Сначала создаётся массив на 4 элемента из integer, хранящий количество кораблей. Первый элемент 4, второй 3 и т.д.

Подпрограмма проходится по каждому элементу матрицы или пока не обнаружится ошибка.

Пропускаем ячейки, пока не находим элемент меньший 0. Потом заменяем его значение на 1, добавляем координаты и пробел в текущую строку массива кораблей(в начале в первую строку) и проверяем соседей(8ячеек вокруг и 1 собственную) с помощью суммы координат оригинальной ячейки и двух переменных dx и dy, изменяющихся от -1 до 1. Необходимо учитывать границы массива и пропускать заграничные ячейки. Изначально координаты следующей ячейки = -1, -1.

Если мы нашли клетку с 1 или -1, то проверяем, что dx and dy = 0. Это значит, что клетка не касается углом исходной, иначе записываем ошибку, пропускаем остальные шаги(добавляя где необходимо в условие переменную correct) и завершаем подпрограмму. Если все корректно и ячейка равна -1, то записываем ее координаты в строку из 2 символов(если находим ещё раз - перезаписываем).

После прохода по всем соседям, если координаты не -1, -1, то увеличиваем длину корабля(изначально 1) на 1. Если длина больше 4, то записываем ошибку. Иначе добавляем координаты и пробел в строку корабля, ячейку, координаты которой мы добавили, меняем на 1, и аналогично предыдущим шагам проверяем соседей(можно в отдельную подпрог).

Если не нашли новые координаты, то отнимаем из переменной с индексом, равному длине корабля, в массиве количества кораблей единицу. Если значение уменьшенной переменной меньше 0, то значит кораблей этого типа больше чем нужно, записываем ошибку. Если все корректно, то увеличиваем счётчик кораблей(чтобы координаты следующего корабля записывать в новую строку в массиве кораблей) и продолжаем проходить по матрице.

В конце проверяем что все корабли найдены (все элементы массива количества кораблей равны 0) и если все корректно то подпрограмма завершается, иначе записываем ошибку.

На Выходе должна быть переменная Boolean(что игровое поле корректно), матрица заполненная 0(море) и 1(корабль), массив из 10 элементов с координатами ячеек каждого корабля.

Игровое поле представляе собой матрицу, в которой 0 - море(пустая клетка), 1- невзорванная часть корабля, 2- промах, 3 - раненая часть корабля, 4 - убитая.

2.программа занимается обработкой введенных пользователем координат, если они корректны, то проверка, что эта ячейка ещё не "обстреляна", и если все корректно, то происходят соответствующие изменения в таблицу и если произошло ранение, то идёт проверка на "убийство"(программа 3).

2.а обработка координат

Вход: строка введённая пользователем(примеры: А10, 5д, 6-Д, Г-3).

-------------------------------------------------------------------------

На выходе строка из 2 символов, код первого - номер строки(соотв. цифре), код второго - номер столбца - соответствует букве).

2.б проверка координат

На вход: матрица, строка-координаты "удара"(константы)

Если координатам в матрице соответствует число больше 1, то возвращает false, иначе true

На выход - boolean-подтверждение что координаты попадают в клетку, в которую не происходил удар до этого.

2.в Обновление матрицы'

На вход - матрица, в которую вносятся изменения, координаты удара, массив кораблей

Если в ячейке удара значение 0(море), то оно меняется на 2(промах), если 1, то на 3 и запускается проверка на убийство корабля(3 часть)

На выход - обновленная матрица.

3.программа ищет клетки корабля и проверяет, являются ли они раненными.

На вход : матрица, массив кораблей, строка координаты удара.

Программа проходится по массиву кораблей, пока в одной из строк позиция вхождения строки-координат удара не станет больше 0. Это значит, что в строку данного корабля входит координаты. Потом пока позиция следующего пробела не станет 0, проверяется ячейки матрицы, у которых координаты равны коду символа (позиция пробела -2) и (позиция пробела -1) из. Если хотя бы одна ячейка имеет значение 1, а не 3(ранен), то программа завершается, иначе корабль - убит, программа заново, пока позиция пробела не станет 0, заменяет ячейки матрицы, координаты которых определяются как написано выше, на 4(убит), а также строка, из которой берутся координаты обнуляется.

На выход: обновленная матрица.

4.интерфейс

Сначала проверяются обе матрицы из файлов, если хотя бы одна из них неверна - вывести сообщение.

Потом игра повторяется пока количество подбитых ячеек одного из игроков не равна 20.

Сверху по центру находится сообщение, чей ход сейчас происходит

Выводится матрица в виде таблицы (первый индекс=строки, второй = столбцы), 0 и 1 выводить как точку, 2 - как П, 3- как Р, 4 - как У.

Далее ход игрока повторяется, пока не будет произведен промах или кол-во подбитых ячеек равна 20

Пользователю предлагается ввести координату, пока введённая координата не станет правильной.

Введённая координата проверяется, если неверно, то сообщение об ошибке в координатах или о повторе ячейки(что в ячейку уже стреляли до этого), иначе - выполняется подпрограмма 2.в

После выводится сообшение, чей ход, и матрица заново.

После выводится результат удара: если ячейка, в которую произведен удар, 2, то выводится промах и т.д. если 3 или 4, то счётчик подбитых увеличивается на 1 и выводится соответствующее сообщение.

После этого проверяется, что ячейка, в которую попал удар является 2( промахом) или кол-во подбитых равно 20 , тогда если все же 20 подбитых, то ход следующего пропускается и переход к сообщению о победе. Если не 20, то ходит второй игрок.

После того как у одного из игроков стало 20, то он объявляется победителем. (Все как в быках и коровах).