**Игра «Охота на лис»**

**Суть игры**

1. На поле случайным неизвестным для игрока образом расставляются «лисы», причём несколько из них могут находиться в одной и той же клетке.
2. Игрок задаёт своё положение, введя координаты. В ответ он получает количество «лис», которое пеленгуется из его нынешнего местоположения. Это число указывает, сколько лис расположено в одной вертикали и горизонтали с указанной клеткой.
3. Если местоположение игрока совпало с положением «лисы», она считается найденной.
4. Игра продолжается, пока не будут найдены все «лисы».

**План разработки игры**

Создать консольное приложение, моделирующее игру «Охота на лис».

1. Создать меню

* Свободный режим
* Правила игры
* Выход

1. Перемещение по таблице

Пользователь сначала вводит количество лис и размер поля (не больше 10х10), дальше перемещение только по таблице, но можно выйти из игры

1. Игра

Количество выстрелов и время неограниченно

1. Ведется подсчет затраченного времени и количество выстрелов
2. При каждом выстреле выводится справка:

* Нашли ли вы лис;
* Сколько их там было;
* Сколько лис по пеленгу;
* Сколько лис осталось еще найти.

1. Таблица:

* По умолчанию ячейки зеленые;
* Ячейка, в которую стреляют – красная;
* Ячейки по пеленгу – желтые;
* Ячейки, в которых лису уже нашли – высвечивается «V».
* Ячейки, в которых лису уже не нашли – высвечивается «X».

**Спецификация структуры ViewTable**

Описание структуры: в структуре реализована работа с видом таблицы.

Методы:ViewTable, SetTableCursorPosition, DrawField, Peleng, ConsoleMenu, SelectItemMenu, ConsoleYesNo, SelectYesNo.

**Описание методов структуры**

public ViewTable(int Size) – конструктор;

* Size – размер поля;

private void SetTableCursorPosition(int row, int column) - реализована формула которая переводит обычные координаты в экранные;

* row – координата выбранной ячейки по горизонтали;
* column – координата выбранной ячейки по вертикали;

public void DrawField(Table tb) – рисует таблицу;

* tb – программная таблица;

public void Peleng(int targetRow, int targetColumn, Table tb) – рисует пеленг точки (желтым цветом) и саму точку в которую попали (красным цветом);

* targetRow – координата выбранной ячейки по горизонтали;
* targetColumn – координата выбранной ячейки по вертикали;
* tb – программная таблица;

public void ConsoleMenu () – рисует пункты меню, фон серый;

public void SelectItemMenu(Menu mn) – рисует фон выбранного пункта меню зеленым;

* mn – выбранный пункт меню;

public void ConsoleYesNo(int x, int y) – рисует кнопки «» и «», фон серый;

* x – координата по горизонтали, где рисовать кнопки;
* y – координата по вертикали, где рисовать кнопки;

public void SelectYesNo(YesNo ch, int x, int y) – фон выбранной кнопки закрашивает зеленным;

* ch – кнопка, которую выбрал пользователь;
* x – координата по горизонтали, где рисовать кнопки;
* y – координата по вертикали, где рисовать кнопки;

**Спецификация структуры UserActions**

Описание структуры: в структуре реализована работа с действиями пользователя.

Методы:GetUserAction().

Реализовано перечисление Action (действия, которые может совершить пользователь).

**Описание методов структуры**

public static Action GetUserAction() – в зависимости от того какую кнопку нажал игрок, выполняется действие (движение влево, вправо, вверх, вниз, Enter и Esc);

**Спецификация структуры Table**

Описание структуры: в структуре реализованы вычисления и работа с таблицей

Методы:Table, MoveCursore, CountPeleng(), MyRandom(), CountKillFoxs()

Реализовано перечисление Menu (пункты меню), YesNo (кнопки «да», «нет»).

**Описание методов структуры**

public Table(int size, int countOfFoxes) **–** конструктор**;**

* size – размер поля;
* countOfFoxes – количество лис;

public void MoveGameCursore (int dx, int dy, ViewTable tb) – движение курсора по игровому полю;

* dx – сдвиг курсора по горизонтали;
* dy – сдвиг курсора по вертикали;
* tb – пользовательская таблица (видимое игровое поле);

public int EnterFoxes(ViewTable tb, ref int ch) – определяется нашел ли пользователь лису, сколько выстрелов уже было сделано, какую информацию надо выводить на экран, количество лис по полену определенной точки, количество уж найденных лис, количество лис, которых еще надо найти;

* tb – пользовательская таблица (видимое игровое поле);
* ch – если ch: 0 – не нашли лису, 1 – нашли лису, -1 – попали в клетку где находится игрок, 2 – игрок победил;

public void MoveMenu(ViewTable tb, int dy) – выбирается пункт меню;

* tb – пользовательская таблица (видимое игровое поле);
* dy – смещение по вертикали среди пунктов меню;

public void MoveYesNo(ViewTable tb, int dx) – если игрок победил, он выбирает либо начать новую игру, либо выйти в меню;

* tb – пользовательская таблица (видимое игровое поле);
* dх – смещение по горизонтали среди кнопок «Да» и «Нет»;

public void MoveYesNoExit(ViewTable tb, int dx) – если игрок в течении игры захотел выйти, он может выбрать: либо продолжить игру или выйти в меню;

* tb – пользовательская таблица (видимое игровое поле);
* dх – смещение по горизонтали среди кнопок «Да» и «Нет»;