Содержание

[ОБЩАЯ ИНФОРМАЦИЯ 2](#_heading=h.gjdgxs)

[Тема занятия](#_heading=h.30j0zll) 2

[Цели и задачи 2](#_heading=h.1fob9te)

[Ожидаемый результат 2](#_heading=h.3znysh7)

[Структура занятия 3](#_heading=h.tyjcwt)

[ОПИСАНИЕ ЗАНЯТИЯ 4](#_heading=h.3dy6vkm)

# ОБЩАЯ ИНФОРМАЦИЯ

### Тема занятия

### Подготовка к проекту. Tic-Tac-Toe-2.

**Цели и задачи:**

* Научиться использовать полученные знания синтаксиса и основ программирования для создания проекта-игры

**По результатам занятия слушатель будет знать:**

* Как лучше подступить к реализации собственных проектов
* Что такое константная ссылка на объект-вектор

**По результатам занятия слушатель будет уметь:**

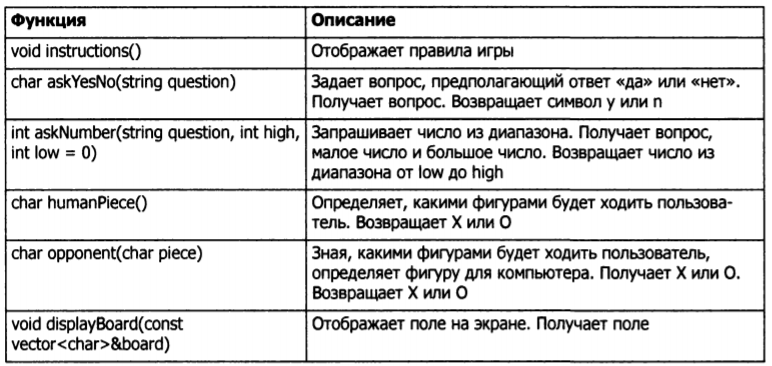
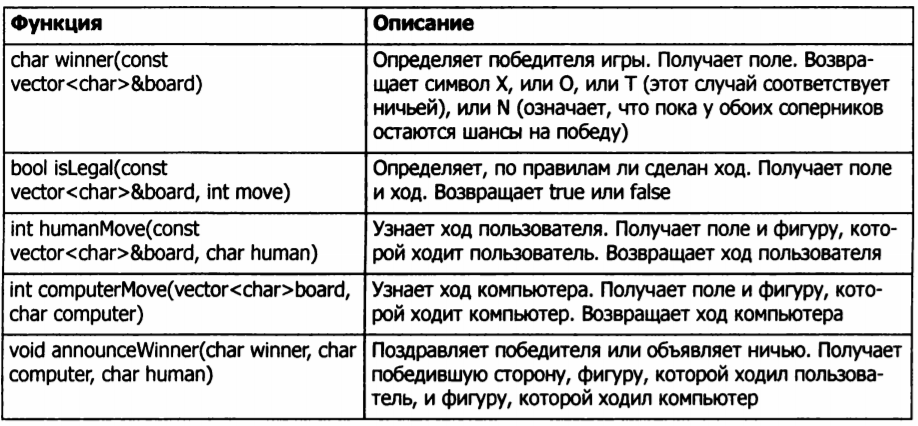
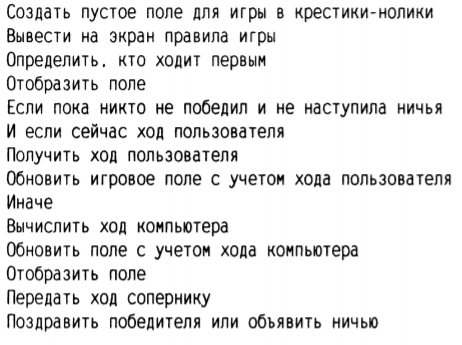
* Работать с константными ссылками на объект-вектор
* Понимать и разбираться в выборе структур данных для проектов

### Структура занятия

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тайминг занятия** | |  |  |  | Таблица 1 |
| № | Этапы | Что делает преподаватель | Что делает Слушатель | Время | Общее время |
| 1 | Приветственное слово преподавателя |  |  | 5 мин. | 5 мин. |
| 2 | Повторение пройденного |  |  | 5 мин. | 40 мин. |
| 3 | Теоретическая часть |  |  | 30 мин. |
| 4 | Вопросы по теоретической части |  |  | 5 мин. |
|  | *Перерыв* |  |  | 15 мин. | 15 мин. |
| 5 | Практическая часть |  |  | 40 мин. | 45 мин. |
| 6 | Рефлексия и вопросы |  |  | 5 мин. |

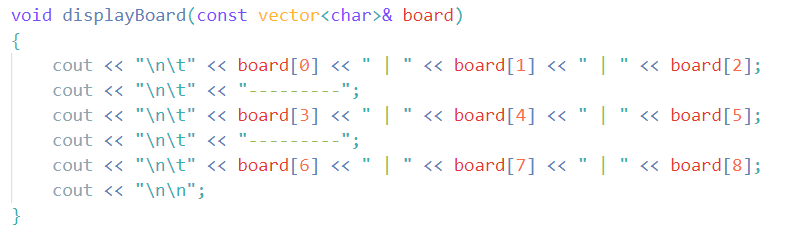
# ОПИСАНИЕ ЗАНЯТИЯ

**Практическая часть**



**Функция displayBoard()**

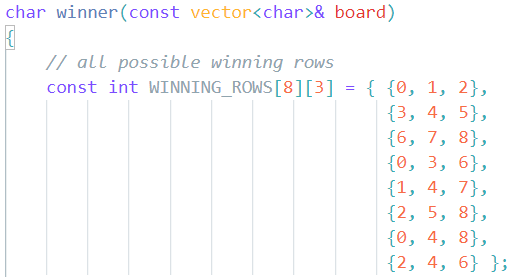
Эта функция отображает переданное ей игровое поле. Поскольку в игре присутствуют элементы всего трех видов: Х, О или пробел, функция может отображать каждый из этих элементов. Чтобы нарисовать красивое поле для игры в “Крестики нолики”, я использую и некоторые символы с клавиатуры:



Обратите внимание: вектор, представляющий игровое поле, передается с помощью константной ссылки. Таким образом, передача вектора организована эффективно, он не копируется. Это также означает, что вектор надежно защищен от любых изменений. Поскольку в этой функции я собираюсь просто отобразить игровое поле, но ничего не планирую с ним делать, такой вариант идеален.

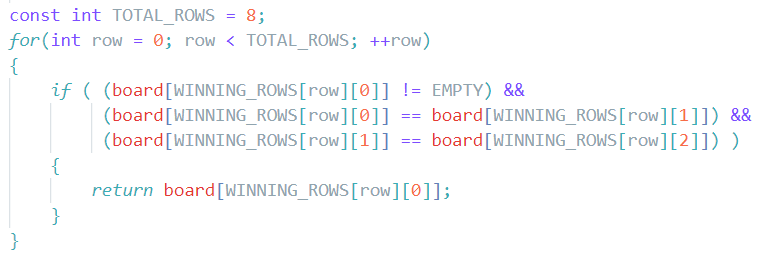
**Функция winner()**

Данная функция получает игровое поле и возвращает победителя. Она может иметь один из четырех вариантов значения. Если один из игроков победил, то функция вернет Х или О. Если игра окончилась вничью, то функция вернет ТIЕ. Наконец, если пока никто не победил, а на поле остается еще хотя бы одна свободная клетка, эта функция вернет NO\_ONE:



В начальной части функции я определяю константу - двухмерный массив целых чисел (int); эта константа будет называться WINNING\_ROWS. В ней представлены все восемь способов заполнить своими фигурами одну из линий и выиграть. Каждый выигрышный ряд представлен группой из трех чисел (тремя выигрышными позициями). Например, группа {О . 1. 2} соответствует верхней строке - позициям О, 1 и 2. Следующая группа { 3. 4. 5} соответствует средней строке - позициям 3, 4 и 5 и т. д.

Затем я проверяю, не победил ли уже один из игроков:



Я перебираю все возможные способы, которыми может победить игрок, проверяя, нет ли в любом из выигрышных рядов трех одинаковых символов. Инструкция i f проверяет, содержатся ли во всех клетках того или иного выигрышного ряда три одинаковых значения, и, если так, не равны ли эти значения ЕМРТУ. При выполнении обоих условий можно констатировать, что хотя бы в одном из рядов есть три крестика или три нолика, то есть один из игроков победил. Затем функция возвращает символ, стоящий в первой позиции выигравшего ряда.

Если ни один из соперников не победил, я проверяю, не наступила ли ничья:

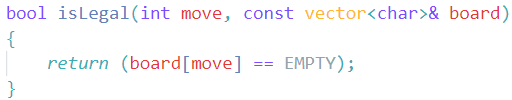


Если на поле не осталось пустых клеток, а победитель не определился, то игра закончилась вничью. Я использую алгоритм count ( ) из библиотеки STL, который подсчитывает, сколько раз заданное значение встречается в контейнере элементов, чтобы определить количество еще остающихся элементов, равных ЕМРТУ. Если это число равно О, то функция возвращает Т1 Е. Наконец, если ни один из игроков пока не победил, но и ничья в игре пока не наступила, то игра продолжается. Функция возвращает константу NO\_ONE:



**Функция isLegal()**

Эта функция получает игровое поле и сделанный ход. Она возвращает true, если ход сделан по правилам, и false - если не по правилам. Ход по правилам заключается во вставке символа Х или О в пустую клетку:



**Функция humanMove()**

Эта функция получает игровое поле и ту фигуру, которой ходит пользователь. Она возвращает номер той клетки, в которую пользователь хочет поставить свой символ. Функция запрашивает у пользователя номер клетки, в которую он хочет поставить символ, и ждет, пока пользователь не походит по правилам. Затем функция возвращает сделанный ход:

