



## Урок 1

# Крестики-нолики В ООП-стиле. Определение классов

Рефакторинг игры. GUI. Классы в игре.

[Рефакторинг игры](#)

[GUI](#)

[Классы в игре](#)

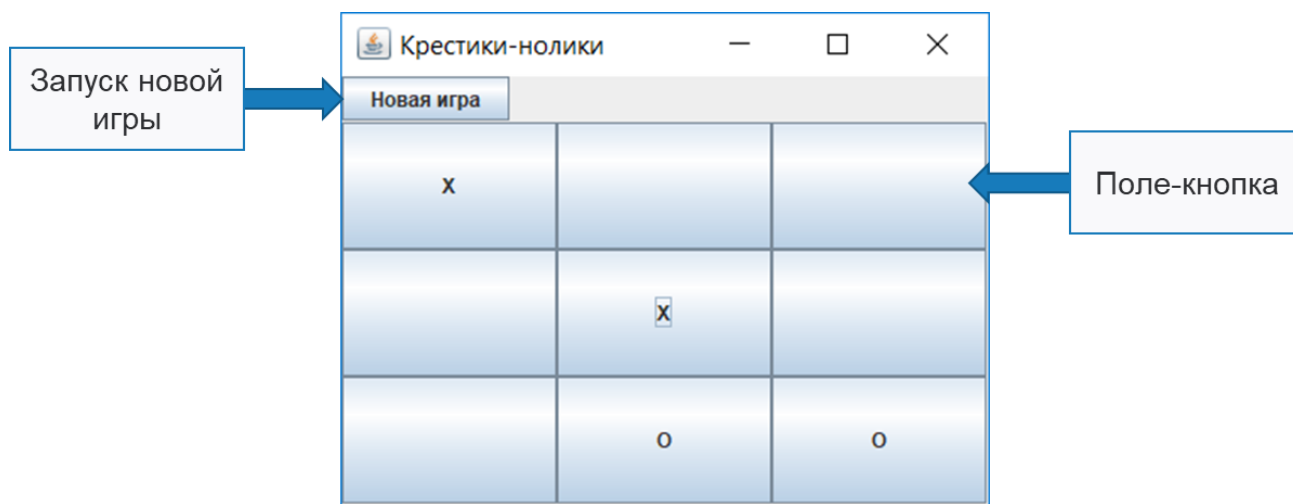
# Рефакторинг игры

В процедурной версии игры мы писали весь код в единственном классе, храня в нем логику, переменные состояния и методы. Теперь полученные знания подсказывают, что это не лучший подход для реализации поддерживаемой программы на Java, которая строится на парадигме ООП.

Перестроим игру в ООП-стиле, параллельно добавив ей GUI.

## GUI

Начнем с интерфейса пользователя.



Игра будет представлять системное окно, содержащее игровое поле и кнопку запуска. Каждая ячейка поля будет кнопкой, чтобы сразу поддерживать нажатия. При них будет производится проверка на доступность поля к ходу и проставляться символ.

Полезно показывать пользователю информационные окна о состоянии игры.

## Классы в игре

Перечислим сущности, которые будут содержаться в коде игры.

Это сама игра, которая будет хранить состояние, ссылки на участников, указатель текущего хода, а также перечень методов для инициации игры, передачи хода, получения текущего игрока и отправки информационных сообщений.

Игровое поле будет сущностью, которая отвечает за его отрисовку, а значит расширяет класс **JFrame**. Там же расположим размерность поля, размер клетки для графического отображения, ссылку на поле и на игру.

Методы будут реализовывать инициацию и очистку поля, получение ссылки на игру, проверку клетки на возможность хода, обновление состояния поля и проверку победы.

Будет небольшой класс игрока.

Специально для игры реализуем игровые кнопки и свой слушатель событий, который реагирует на нажатия по ним.

Работу с кнопками можно облегчить, если знать размерность поля. Тогда строку и столбец можно вычислить через номер кнопки.

	0	1	2
0	0	1	2
1	3	4	5
2	6	7	8

Строка = номер / размер;  
Столбец = номер % размер;

При таком подходе вложенные циклы можно упростить до одноуровневых.

Созданные сущности должны взаимодействовать между собой. В нашем коде Игра будет начинать работу программы, создавая поле и участников. Игровое поле будет инициировать кнопки ячеек. Далее программа перейдет в режим ожидания событий, которые будет прослушивать **GameActionListener**.