



Урок 3

Крестики-нолики. Генерация координат хода компьютера

Алгоритм ходов компьютера и его реализация.

[Генерация координат хода компьютера](#)

Генерация координат хода компьютера

«Глупый» компьютер будет ставить свои символы в ячейки случайным образом при помощи класса `java.util.Random`. Для этого потребуется как минимум экземпляр этого класса.

```
private static Random random = new Random();
```

В методе хода компьютера — по аналогии с методом хода игрока — будут нужны две координаты. Поскольку машина будет генерировать свой ход случайно, удобно задать этим координатам заведомо некорректные значения. Потом в цикле будем присваивать уже реальные значения, которые можно проверять на корректность.

Поэтому генерацию координат будем делать тоже через **do..while**.

Для прозрачности можно добавить и оповещения для пользователя, что компьютер выбрал ту или иную ячейку.

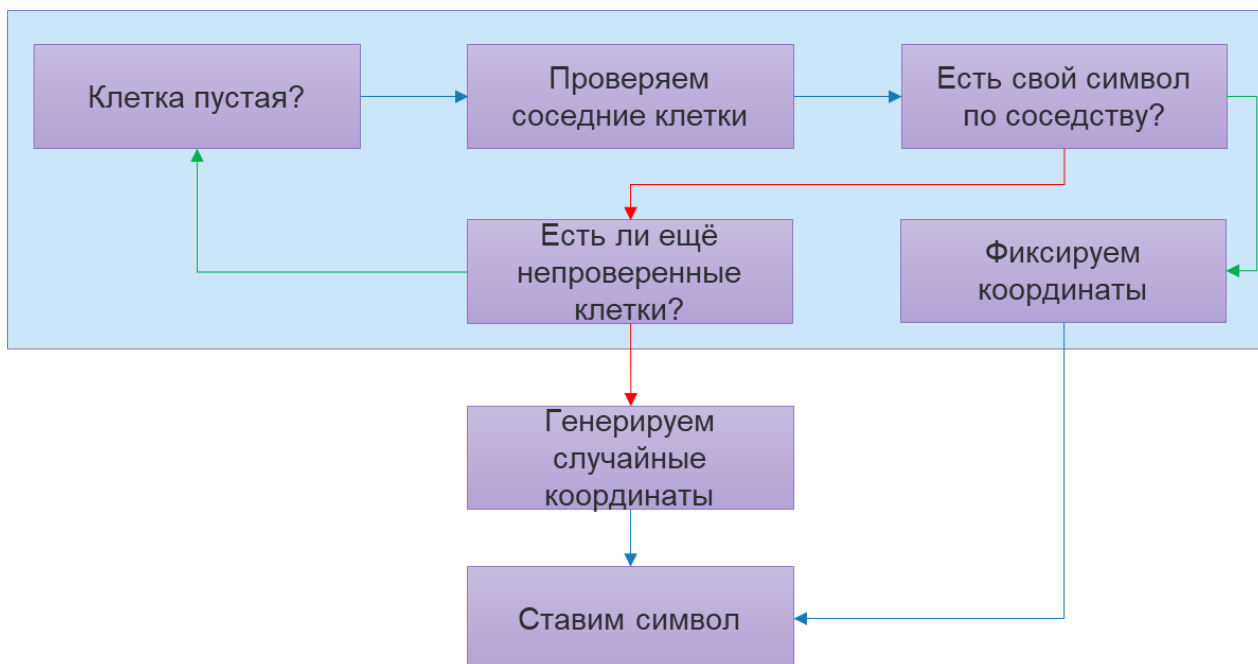
```
private static void computerTurn() {
    int x = -1;
    int y = -1;

    if (SILLY_MODE) {
        do {
            x = random.nextInt(SIZE);
            y = random.nextInt(SIZE);
        } while (!isCellValid(x, y));
    }
    else {
        for (int i = 0; i < SIZE; i++) {
            for (int j = 0; j < SIZE; j++) {
                // Проверяем клетки по направлениям
            }
        }
    }

    System.out.println("Компьютер выбрал ячейку " + (y + 1) + " " + (x + 1));
    map[y][x] = DOT_O;
}
```

В самом методе хода оставим «вилку» для усложненной логики, которую вам нужно будет реализовать в домашнем задании.

Алгоритмов очень много — например, стратегия Минимакса. Для реализации на нынешнем уровне знания языка применим алгоритм, диктующий компьютеру ставить свои символы в ту клетку, где по соседству уже есть его отметки. Опишем это схемой шагов:



Также можно просчитать «выгодность» клетки, ее рейтинг, выраженный в количестве соседних клеток, которые будут для компьютера «своими». Пройти поле можно двумя итерациями, что уже реализовано в приведенном выше листинге.

Подсказка для домашнего задания, где много работы с вложенными циклами: конструкция ***break*** прерывает только один цикл!