**Задача 1.** Как сделать, чтобы такой двойной цикл печатал таблицу из \* размером 6х6, и чтобы между звездочками стояли пробелы? Просто меняем условия работы циклов и корректируем print().

|  |
| --- |
| public class MainApp {     public static void main(String[] args) {         for (int i = 0; i < 6; i++) {             for (int j = 0; j < 6; j++) {                 System.*out*.print("\* ");             }             System.*out*.println();         }     }  }  **Результат:**  \* \* \* \* \* \*  \* \* \* \* \* \*  \* \* \* \* \* \*  \* \* \* \* \* \*  \* \* \* \* \* \*  \* \* \* \* \* \* |

**Задача 2.** А теперь давайте попробуем сделать так, чтобы правая половина этой таблицы состояла бы не из \*, а из 0. Добавим для этого условие.

|  |
| --- |
| public class MainApp {     public static void main(String[] args) {         for (int i = 0; i < 6; i++) {             for (int j = 0; j < 6; j++) {                 if (j < 3) {                     System.*out*.print("\* ");                 } else {                     System.*out*.print("0 ");                 }             }             System.*out*.println();         }     }  }  **Результат:**  **\* \* \* 0 0 0**  **\* \* \* 0 0 0**  **\* \* \* 0 0 0**  **\* \* \* 0 0 0**  **\* \* \* 0 0 0**  **\* \* \* 0 0 0** |

**Задача 3.** Сделать так, чтобы по краям квадрата стояли 0, а внутри он был бы заполнен \*.

Для этого необходимо проверять, если номер строки или столбца равен 0 или (размер таблицы - 1), тогда ставим 0, в противном случае \*.

|  |
| --- |
| public class MainApp {     public static void main(String[] args) {         for (int i = 0; i < 6; i++) {             for (int j = 0; j < 6; j++) {                 if (i == 0 || i == 5 || j == 0 || j == 5) {                     System.*out*.print("0 ");                 } else {                     System.*out*.print("\* ");                 }             }             System.*out*.println();         }     }  }  **Результат:**  **0 0 0 0 0 0**  **0 \* \* \* \* 0**  **0 \* \* \* \* 0**  **0 \* \* \* \* 0**  **0 \* \* \* \* 0**  **0 0 0 0 0 0** |

**Задача 4.** **Сложный пример.** Как заполнить таблицу в шахматном порядке? Попробуйте разобраться как работает следующий код. Для этого вам может пригодиться листик и ручка, чтобы нарисовать такую таблицу, и пронумеровать строки и столбцы. Напоминаем, что % это вычисление остатка от деления.

|  |
| --- |
| public class MainApp {     public static void main(String[] args) {         for (int i = 0; i < 6; i++) {             for (int j = 0; j < 6; j++) {                 if ((i + j) % 2 == 0) {                     System.*out*.print("\* ");                 } else {                     System.*out*.print("0 ");                 }             }             System.*out*.println();         }     }  }  **Результат:**  **\* 0 \* 0 \* 0**  **0 \* 0 \* 0 \***  **\* 0 \* 0 \* 0**  **0 \* 0 \* 0 \***  **\* 0 \* 0 \* 0**  **0 \* 0 \* 0 \*** |

**Задача 5.** **Сложный пример.** Заполнить одну половину таблицы 0, а вторую \*, при этом половину разделены диагональю.

|  |
| --- |
| public class MainApp {     public static void main(String[] args) {         for (int i = 0; i < 6; i++) {             for (int j = 0; j < 6; j++) {                 if (i > j) {                     System.*out*.print("0 ");                 } else if (i < j) {                     System.*out*.print("\* ");                 } else {                     System.*out*.print("  ");                 }             }             System.*out*.println();         }     }  }  **Результат:**  **\* \* \* \* \***  **0   \* \* \* \***  **0 0   \* \* \***  **0 0 0   \* \***  **0 0 0 0   \***  **0 0 0 0 0** |

**Разобрались? Тогда можете попробовать “зарисовать” вот такие картинки.** Это не домашнее задание, а просто дополнительная тренировка.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Рисунок 1** | **Рисунок 2** | **Рисунок 3** | **Рисунок 4** | **Рисунок 5** |
| **\* \* \* \* \* \***  **\*         \***  **\*         \***  **\*         \***  **\*         \***  **\* \* \* \* \* \*** | **\* \* \* \* \* \***  **\* \*       \***  **\*   \*     \***  **\*     \*   \***  **\*       \* \***  **\* \* \* \* \* \*** | **\***  **\* \***  **\*   \***  **\*     \***  **\*       \***  **\* \* \* \* \* \*** | **\* \* \* \* \* \***  **\* \* \* \* \* \***  **\* \*     \* \***  **\* \*     \* \***  **\* \* \* \* \* \***  **\* \* \* \* \* \*** | **\* \* \* \***  **\* \* \* \***  **\* \* \* \* \* \***  **\* \* \* \* \* \***  **\* \* \* \***  **\* \* \* \*** |

Для построения рисунков вам может пригодиться вот такая схема. Например, по рисунку на схеме хорошо видно особенность координат диагональных элементов. Для каждого рисунка можете составлять такую схему.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** |
| **0** | \* | \* | \* | \* | \* | \* |
| **1** |  | \* |  |  |  | \* |
| **2** |  |  | \* |  |  | \* |
| **3** |  |  |  | \* |  | \* |
| **4** |  |  |  |  | \* | \* |
| **5** |  |  |  |  |  | \* |

Когда пытаетесь решить любую задачу, пытайтесь найти зависимость между ячейками со звездочками и соответствующими индексами строк и столбцов.