Программирование на языках высокого уровня

Лаб. работа 6. 2D-массивы

Общее задание

Требуется написать программу, визуализирующую 2D-массив, заполненный целыми числами от 0 до 9 в виде игрового поля. Число от 0 до 9 в ячейке поля соответствует его состоянию. Состояние ячейки отображать цветом. Исходный массив загружать из файла.

Чтение из текстового файла

Для начала нужно создать Scanner:

```
Scanner sc = new Scanner(new File("C:\\Temp\\input.txt"));
```

потом подключить библиотеки для его работы:

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
```

Eclipse подчеркнет строку с созданием Scanner, нужно навести на нее мышку и выбрать в подсказке **Surround with Try/Catch**
Eclipse добавит обработку исключения:

```
try {
    Scanner sc = new Scanner(new File("C:\\Temp\\input.txt"));
} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

Всю остальную работу с файлом делаем внутри try. Не забываем закрыть Scanner, вызвав `sc.close();`

Работа с текстовым файлом

Если вы хотите загружать данные из файла в панели, модифицированной для рисования, то не загружайте данные из файла в методе paint. Вместо этого в классе панели (которую мы могли назвать **MyPanel**) лучше всего создать **метод-конструктор**, выполняющийся автоматически при создании панели. **Конструктор**, загружающий двумерный массив из файла может выглядеть так:

```
public class MyPanel extends JPanel {

    private int arr[][]; //Массив
    private int w, h;    //Ширина и высота

    public MyPanel() {                          //Конструктор, он всегда public и без типа
        try {
            Scanner sc = new Scanner(new File("C:\\Temp\\input.txt"));
            w = sc.nextInt();                   //Загрузка ширины
            h = sc.nextInt();                   //Загрузка высоты
            arr = new int[w][h];                //Выделение памяти для массива
            for (int i = 0; i < h; i++) {       //Внешний цикл бежит по стокам
                for (int j = 0; j < w; j++) {   //Внутренний - по числам в строке
                    arr[j][i] = sc.nextInt();   //В первых скобках - координата X - номер столбца, а во вторых - Y
                }
            }
            sc.close();                         //Не забываем закрыть файл
        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
```

Обратите внимание на объявления переменных в начале класса. В конструкторе они заполняются данными, затем их можно использовать и в методе **paint**. Но мы ставим перед ними **private**, чтобы спрятать их “под капот” нашего класса – они не должны быть доступны снаружи

Домашнее задание

Добавить в программу обработку массива, соответствующую варианту и выполнить визуализацию массива после обработки. Если число оказалось выходящим за пределы [0..9] – закрашивать ячейку красным цветом

Домашнее задание

Варианты А

A1	Переставить строки, содержащие минимальный и максимальный элементы	4 3 0 2 3 1 3 4 2 -1 5 3 4 6	4 3 0 2 3 1 3 4 3 4 6 2 -1 5
A2	Все нулевые элементы заменить на минимальный элемент столбца	4 3 0 2 0 1 3 0 -3 -5 5 3 4 -6	4 3 -3 2 -6 1 3 -6 -3 -5 5 3 4 -6
A3	Все нечетные элементы заменить на максимальный элемент столбца	4 3 0 2 0 1 3 0 -3 -5 5 3 4 -6	4 3 -3 2 -6 1 3 -6 -3 -5 5 3 4 -6
A4	Все четные элементы заменить на максимальный элемент строки	4 3 2 3 4 5 4 6 7 9 1 1 2 3	4 3 4 3 4 5 6 6 7 9 1 1 3 3
A5	Переставить столбцы, имеющие максимальную и минимальную сумму.	4 3 1 2 3 4 5 6 7 8 9 1 5 7	4 3 3 2 1 6 5 4 9 8 7 7 5 1
A6	Переставить строки, имеющие максимальную и минимальную сумму.	4 3 1 2 3 4 5 6 7 8 9 1 5 7	4 3 7 8 9 4 5 6 1 2 3 1 5 7

A7	Обнулить элементы в строках, где нет четных элементов	4 3 1 2 3 4 5 6 7 8 9 1 5 7	4 3 1 2 3 4 5 6 7 8 9 0 0 0
A8	Увеличить в 10 раз элементы в столбцах, где нет нечетных элементов	4 3 1 2 3 4 4 6 7 6 9 1 2 7	4 3 1 20 3 4 40 6 7 60 9 1 20 7
A9	Увеличить в 10 раз элементы, ниже которых находятся нечетные элементы	4 3 1 2 3 4 4 6 7 6 9 1 2 7	4 3 1 2 3 40 4 60 70 6 90 1 2 7
A10	Уменьшить в 2 раза элементы, правее которых находятся четные элементы	4 3 1 2 3 4 4 6 7 6 9 1 2 7	4 3 0 2 3 2 2 6 3 6 9 0 2 7
A11	Заменить четные элементы суммой всех нечетных элементов строки	4 3 1 2 3 4 4 5 7 6 9 1 2 7	4 3 1 4 3 5 5 5 7 16 9 1 8 7
A12	Заменить нечетные элементы суммой всех четных элементов столбца	4 3 1 2 3 4 4 5 7 6 8 1 2 7	4 3 4 2 8 4 4 8 4 6 8 4 2 8

Домашнее задание

Варианты В

В1	Удалить строку, содержащую минимальное число	4 3 0 2 3 1 3 4 2 -1 5 3 4 5	3 3 0 2 3 1 3 4 3 4 5
В2	Удалить столбец, содержащий максимальное число	4 3 0 2 3 1 3 4 2 -1 5 3 4 5	4 2 0 2 1 3 2 -1 3 4
В3	Удалить строку, содержащую максимальное количество нечетных чисел	4 3 0 2 3 1 3 5 2 1 5 3 4 5	3 3 0 2 3 2 1 5 3 4 5
В4	Удалить столбец, содержащий максимальное количество четных чисел	4 3 0 2 3 1 4 5 2 6 5 3 4 5	4 2 0 3 1 5 2 5 3 5
В5	Удалить все столбцы, сумма элементов которых четная	4 3 0 2 3 1 3 4 2 1 5 3 4 5	4 1 3 4 5 5
В6	Столбец, содержащий максимальный элемент, продублировать	4 3 0 2 3 1 3 4 2 1 5 9 4 8	4 4 0 0 2 3 1 1 3 4 2 2 1 5 9 9 4 8
В7	Удалить те столбцы, в которых есть элементы, сумма цифр которых равна 10	4 3 19 2 3 1 22 4 28 1 33 9 4 45	4 2 2 3 22 4 1 33 4 45
В8	Удалить те строки, в которых все элементы имеют сумму цифр меньше 5	4 3 11 12 24 13 10 21 22 31 12 99 11 24	2 3 11 12 24 99 11 24
В9	Столбцы, содержащие четные элементы, продублировать	4 3 0 2 3 1 3 7 2 1 5 9 4 8	4 5 0 0 2 2 3 1 1 3 3 7 2 2 1 1 5 9 9 4 4 8
В10	Строки, которые состоят только из простых чисел, продублировать	4 3 3 2 3 2 3 7 2 4 5 9 4 8	6 3 3 2 3 3 2 3 2 3 7 2 3 7 2 4 5 9 4 8
В11	Переставить строки в порядке увеличения суммы элементов строк	4 3 3 2 3 2 3 1 2 4 5 0 2 8	4 3 2 3 1 3 2 3 0 2 8 2 4 5
В12	Переставить столбцы в порядке уменьшения суммы элементов столбцов	4 3 3 2 3 2 3 1 2 4 5 0 2 8	4 3 3 2 3 1 2 3 5 4 2 8 2 0