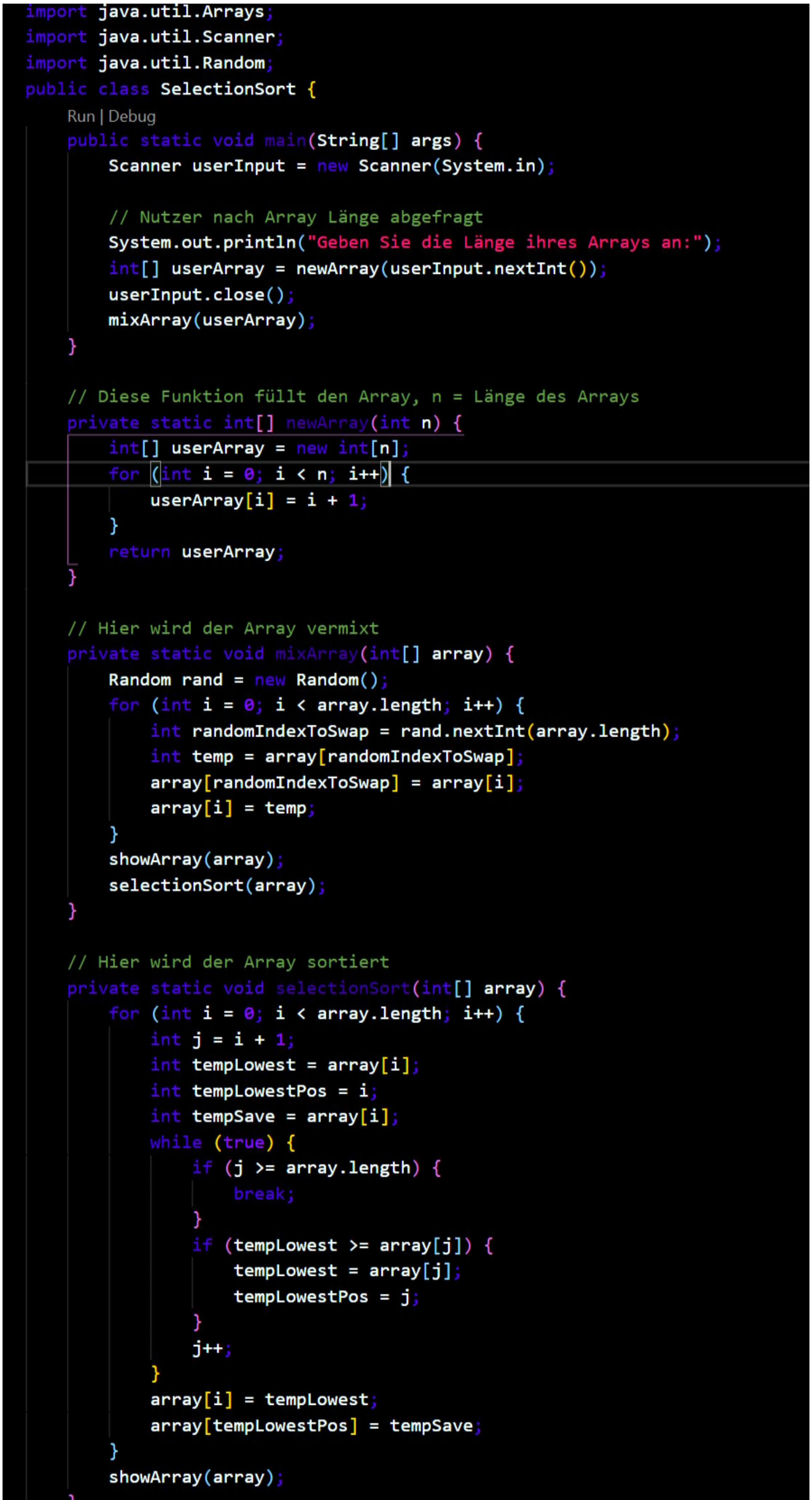
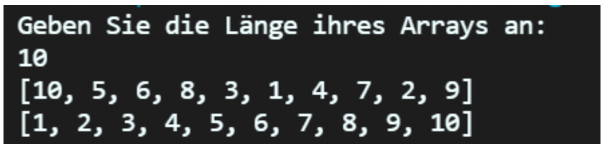
Uebung 6 Patryk Dajos

Aufgabe 1)

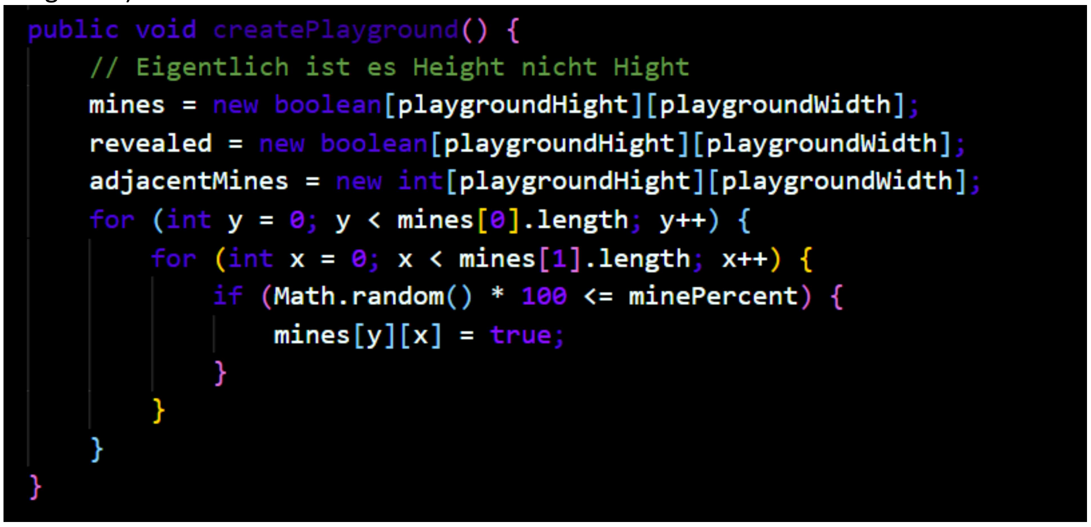


```java
import java.util.Arrays;
import java.util.Scanner;
import java.util.Random;
public class SelectionSort {
    Run | Debug
    public static void main(String[] args) {
        Scanner userInput = new Scanner(System.in);

        // Nutzer nach Array Länge abgefragt
        System.out.println("Geben Sie die Länge ihres Arrays an:");
        int[] userArray = newArray(userInput.nextInt());
        userInput.close();
        mixArray(userArray);
    }

    // Diese Funktion füllt den Array, n = Länge des Arrays
    private static int[] newArray(int n) {
        int[] userArray = new int[n];
        for (int i = 0; i < n; i++) {
            userArray[i] = i + 1;
        }
        return userArray;
    }

    // Hier wird der Array vermixt
    private static void mixArray(int[] array) {
        Random rand = new Random();
        for (int i = 0; i < array.length; i++) {
            int randomIndexToSwap = rand.nextInt(array.length);
            int temp = array[randomIndexToSwap];
            array[randomIndexToSwap] = array[i];
            array[i] = temp;
        }
        showArray(array);
        selectionSort(array);
    }

    // Hier wird der Array sortiert
    private static void selectionSort(int[] array) {
        for (int i = 0; i < array.length; i++) {
            int j = i + 1;
            int tempLowest = array[i];
            int tempLowestPos = i;
            int tempSave = array[i];
            while (true) {
                if (j >= array.length) {
                    break;
                }
                if (tempLowest >= array[j]) {
                    tempLowest = array[j];
                    tempLowestPos = j;
                }
                j++;
            }
            array[i] = tempLowest;
            array[tempLowestPos] = tempSave;
        }
        showArray(array);
```

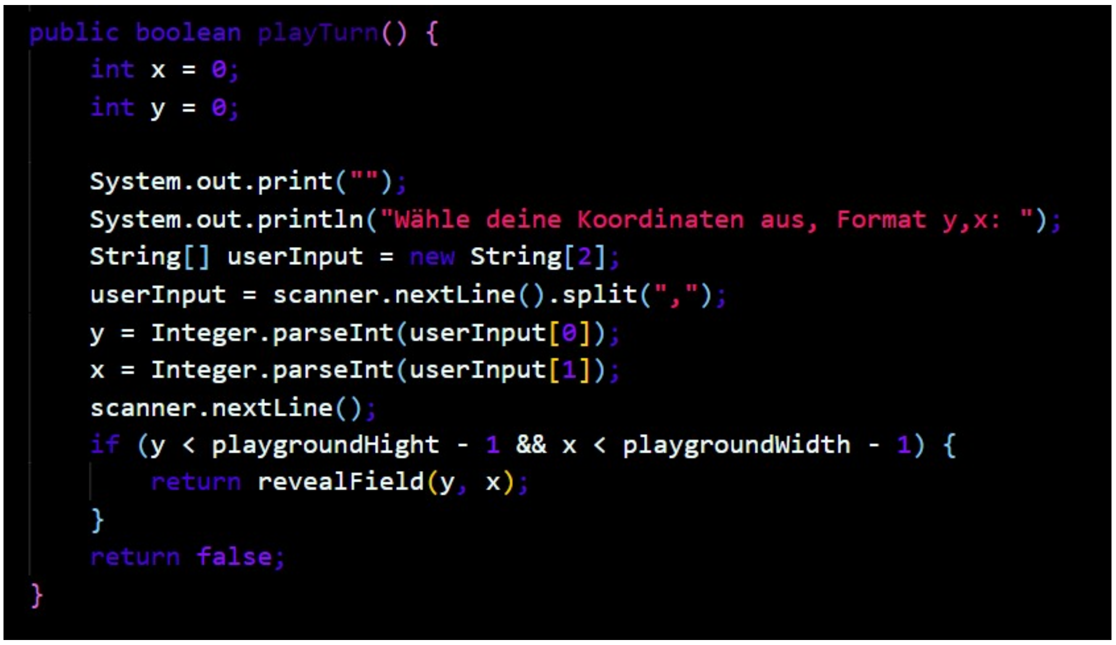


```
Geben Sie die Länge ihres Arrays an:
10
[10, 5, 6, 8, 3, 1, 4, 7, 2, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

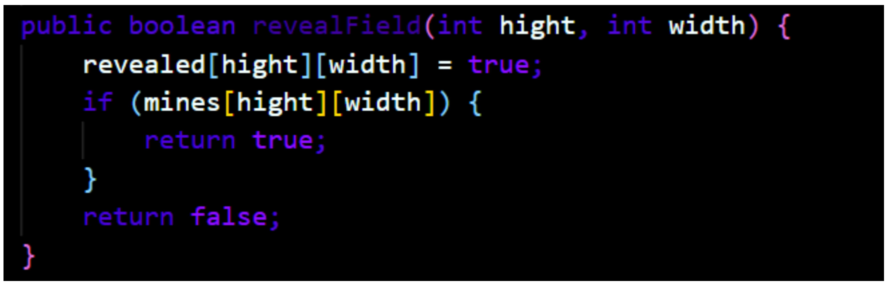
Aufgabe 3)



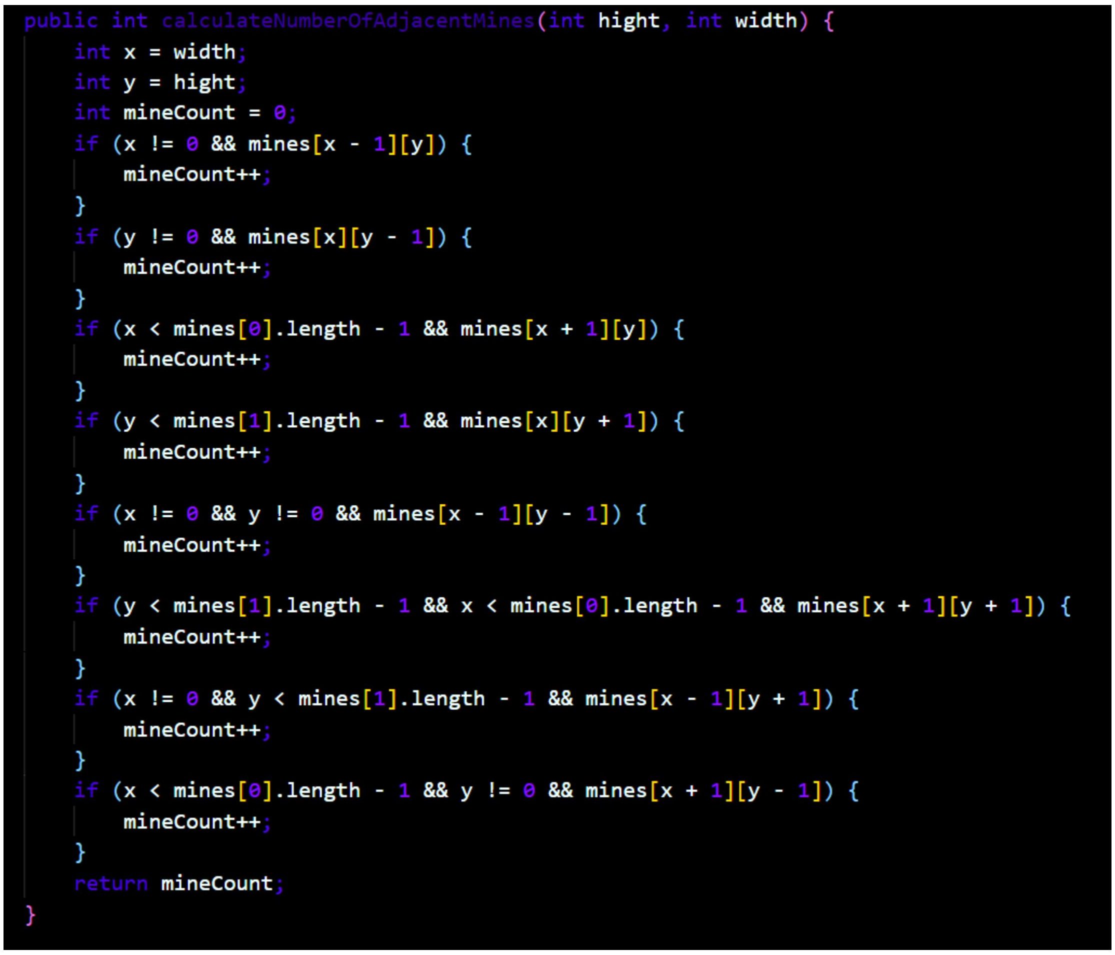
```java
public void createPlayground() {
    // Eigentlich ist es Height nicht Hight
    mines = new boolean[playgroundHight][playgroundWidth];
    revealed = new boolean[playgroundHight][playgroundWidth];
    adjacentMines = new int[playgroundHight][playgroundWidth];
    for (int y = 0; y < mines[0].length; y++) {
        for (int x = 0; x < mines[1].length; x++) {
            if (Math.random() * 100 <= minePercent) {
                mines[y][x] = true;
            }
        }
    }
}
```



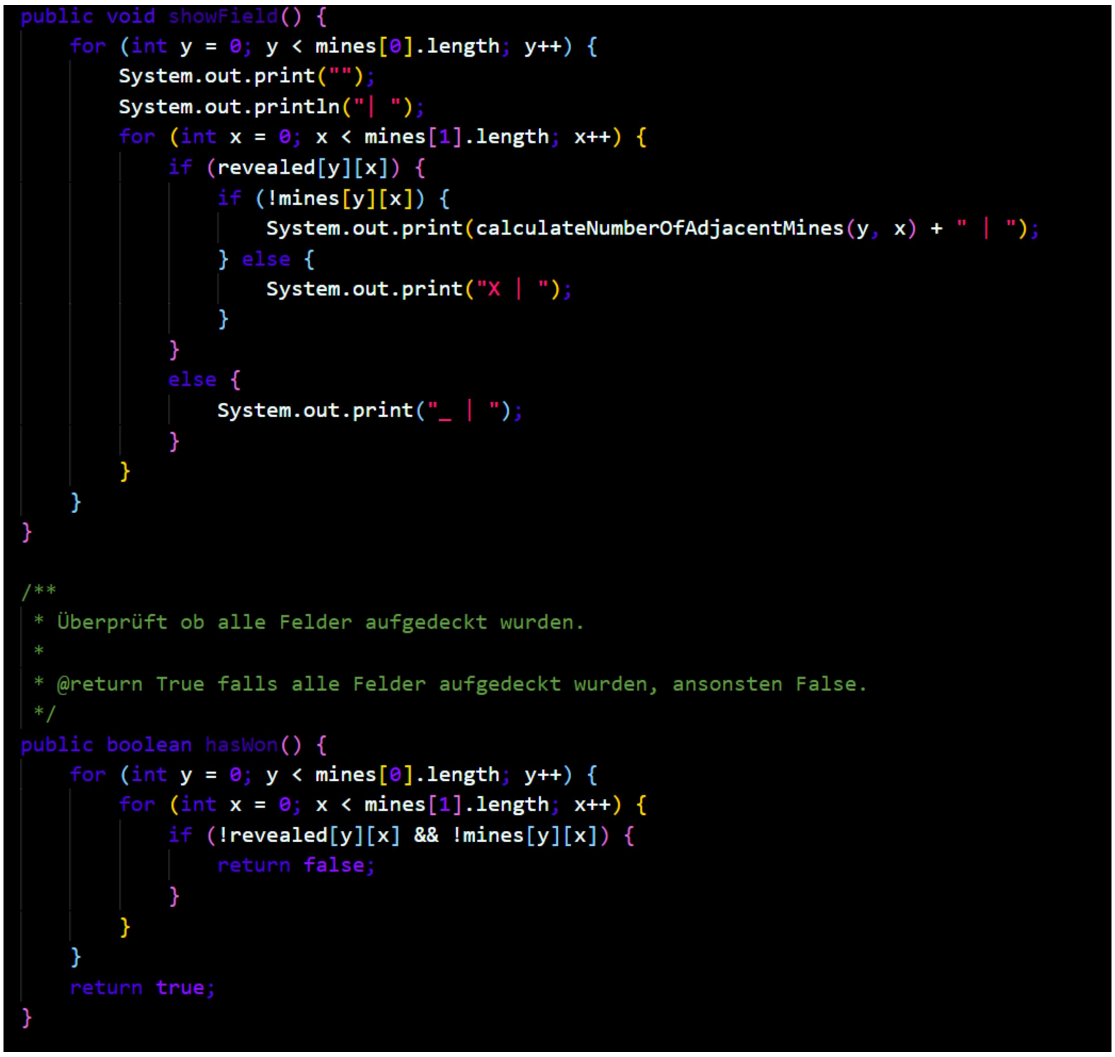
```java
public boolean playTurn() {
    int x = 0;
    int y = 0;

    System.out.print("");
    System.out.println("Wähle deine Koordinaten aus, Format y,x: ");
    String[] userInput = new String[2];
    userInput = scanner.nextLine().split(",");
    y = Integer.parseInt(userInput[0]);
    x = Integer.parseInt(userInput[1]);
    scanner.nextLine();
    if (y < playgroundHight - 1 && x < playgroundWidth - 1) {
        return revealField(y, x);
    }
    return false;
}
```



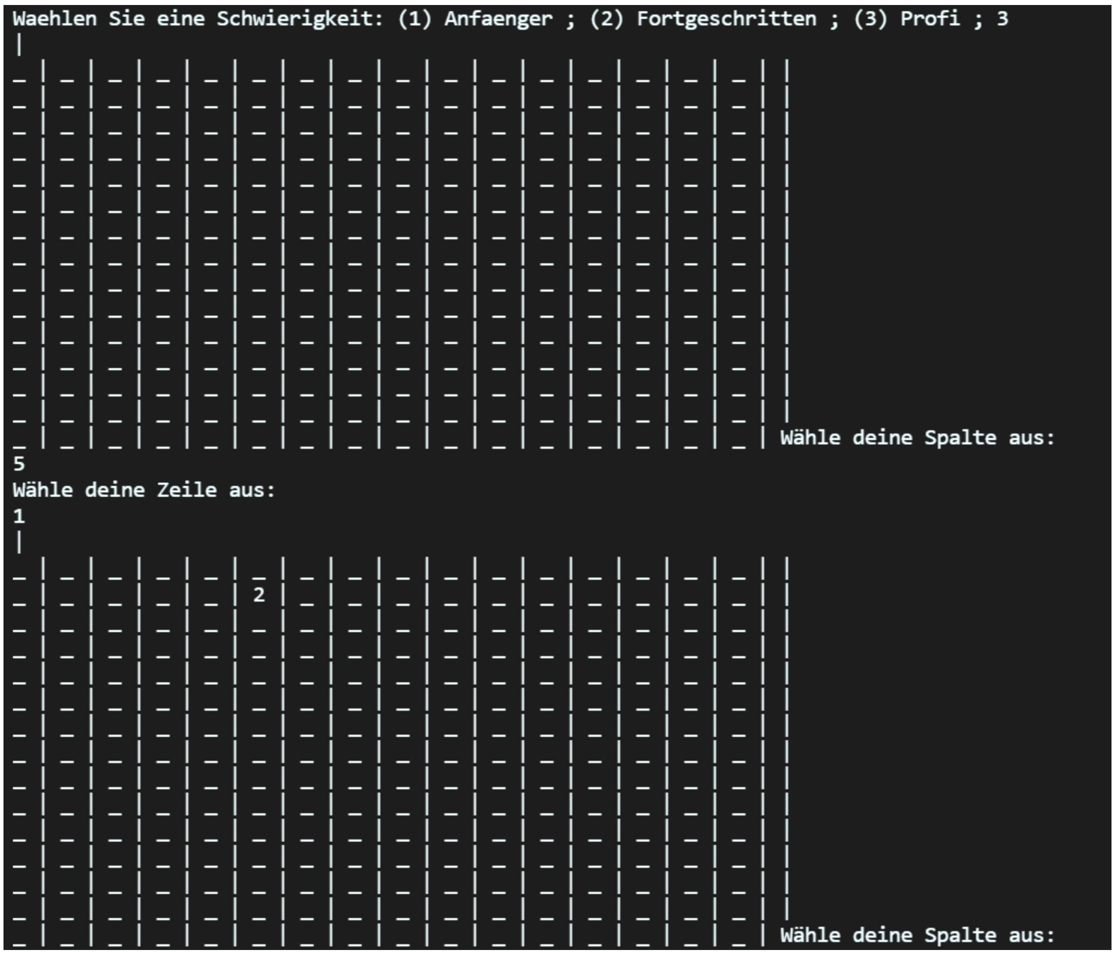
```java
public boolean revealField(int hight, int width) {
    revealed[hight][width] = true;
    if (mines[hight][width]) {
        return true;
    }
    return false;
}
```



```java
public int calculateNumberOfAdjacentMines(int hight, int width) {
    int x = width;
    int y = hight;
    int mineCount = 0;
    if (x != 0 && mines[x - 1][y]) {
        mineCount++;
    }
    if (y != 0 && mines[x][y - 1]) {
        mineCount++;
    }
    if (x < mines[0].length - 1 && mines[x + 1][y]) {
        mineCount++;
    }
    if (y < mines[1].length - 1 && mines[x][y + 1]) {
        mineCount++;
    }
    if (x != 0 && y != 0 && mines[x - 1][y - 1]) {
        mineCount++;
    }
    if (y < mines[1].length - 1 && x < mines[0].length - 1 && mines[x + 1][y + 1]) {
        mineCount++;
    }
    if (x != 0 && y < mines[1].length - 1 && mines[x - 1][y + 1]) {
        mineCount++;
    }
    if (x < mines[0].length - 1 && y != 0 && mines[x + 1][y - 1]) {
        mineCount++;
    }
    return mineCount;
}
```



```java
public void showField() {
    for (int y = 0; y < mines[0].length; y++) {
        System.out.print("");
        System.out.println("| ");
        for (int x = 0; x < mines[1].length; x++) {
            if (revealed[y][x]) {
                if (!mines[y][x]) {
                    System.out.print(calculateNumberOfAdjacentMines(y, x) + " | ");
                } else {
                    System.out.print("X | ");
                }
            }
            else {
                System.out.print("_ | ");
            }
        }
    }
}

/**
 * Überprüft ob alle Felder aufgedeckt wurden.
 *
 * @return True falls alle Felder aufgedeckt wurden, ansonsten False.
 */
public boolean hasWon() {
    for (int y = 0; y < mines[0].length; y++) {
        for (int x = 0; x < mines[1].length; x++) {
            if (!revealed[y][x] && !mines[y][x]) {
                return false;
            }
        }
    }
    return true;
}
```



```
Waehlen Sie eine Schwierigkeit: (1) Anfaenger ; (2) Fortgeschritten ; (3) Profi ; 3
|
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | Wähle deine Spalte aus:
5
Wähle deine Zeile aus:
1
|
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | 2 | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | |
_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | Wähle deine Spalte aus:
```

Das program tut sich schwer zeilen und spalten auszulesen, keine ahnung wieso