

Praktikum zu
**Einführung in die Informatik für
LogWilngs und WiMas**
Wintersemester 2018/19

**Tower-Defense-
Übungsblatt**

Aufgabe 3: Schwächsten/Stärksten Gegner finden

Die Spieler haben die Möglichkeit, die Zielstrategie der Türme auszuwählen, also einem Turm mitzuteilen, welchen Gegner er bevorzugt angreifen soll.

In dieser Aufgabe sollen Sie zwei Funktionen implementieren, die in den Strategien zum Wählen des schwächsten bzw. stärksten Gegners verwendet werden. Es werden die Gegner mit den wenigsten bzw. meisten verbliebenen Lebenspunkten gesucht. Diese Funktionen heißen **indexOfWeakest** und **indexOfStrongest** und sind in der Klasse **Aufgabe3** zu finden.

Beide Funktionen erhalten ein Array mit den Lebenspunkten der Gegner in Reichweite eines Turms und sollen den **Index** im Array zurückgeben, das den kleinsten bzw. größten Wert beinhaltet.

Iterieren Sie dazu über das Array und merken sich den Index mit dem kleinsten bzw. größten Wert. Innerhalb der Schleife müssen Sie dann nur noch den Wert am gemerkten Index mit dem Wert am Index der Schleife vergleichen und den gemerkten Index anpassen, sollten Sie einen kleineren bzw. größeren Wert finden. Das Ergebnis Ihrer Berechnung ist dann der gemerkte Index.

Lösung:

```
package aufgaben;

public class Aufgabe3 {
    public static int indexOfWeakest(int[] healthpoints) {
        int weakest = 0;
        for(int i = 1; i < healthpoints.length; ++i) {
            if(healthpoints[i] < healthpoints[weakest]) {
                weakest = i;
            }
        }
        return weakest;
    }

    public static int indexOfStrongest(int[] healthpoints) {
        int strongest = 0;
        for(int i = 1; i < healthpoints.length; ++i) {
            if(healthpoints[i] > healthpoints[strongest]) {
                strongest = i;
            }
        }
        return strongest;
    }
}
```