

Dokumentation Aufgabe 5

Vorgehensweise:

- Zuerst sortiert man die eingegebenen Gewichte der Größe nach, um damit einfacher arbeiten zu können

```
Collections.sort(weights);
```

- Anschließend durchläuft man Schleifen, um erst abzufragen, ob ein Gewicht genau so schwer ist, wie der Sack

```
for (int i = 10; i <= 10000; i += 10) { //i = Sack, von 10 Gramm bis 10 Kg
    for (int j = 0; j < weights.size() - 1; j++) { //j = Gewichts-Index der Liste
        if (i == weights.get(j)) { //Sack = Gewicht
            System.out.println(i + " Gramm schwerer Sack auf der linken Seite und " + weights.get(j) + " Gramm schweres Gewicht auf der rechten Seite\n");
            break;
        }
    }
}
```

- Wenn der Sack nicht das gleiche Gewicht hat, wie ein Gewicht, werden verschiedene Schleifen durchlaufen, ob man durch Addieren der Gewichte auf das Gewicht kommt. Wichtig war, dass jedes Gewicht nur 1 mal pro Durchgang genutzt werden kann

```
private boolean addLowerWeight(int weight, int index) { //weight = Gewicht, index = Gewichts-Index der Liste (schwerer als der Sack)
    for (int i = 0; i < weights.size() - 1; i++) { //i = Gewichts-Index der Liste
        int newWeight = weights.get(index - 1) + weights.get(i); //Die Gewichte auf der rechten Seite
        if (weight == newWeight) { //Sack = Gewichte
            System.out.println(weight + " Gramm schwerer Sack auf der linken Seite und " + weights.get(index - 1) + " und " + weights.get(i) + " Gramm schweres Gewicht auf der rechten Seite\n");
            return true;
        } else if (weight < newWeight) { //Sack < Gewichte
            System.out.println(weight + " Gramm ist leichter als " + newWeight + " Gramm schweres Gewicht");
            if (i != 0) {
                return addLowerWeight1(weight, weights.get(index - 1), i);
            } else {
                return false;
            }
        }
    }
}
```

```

    }
    return false;
}

private boolean addLowerWeight1(int weight, int minWeight, int add1) {
    for (int i = 0; i < weights.size() - 1; i++) { //i = Gewichts-Index
        der Liste
        if (i != add1) { //Nicht das gleiche Gewicht
            int newWeight = weights.get(i) + minWeight +
weights.get(add1); //Die Gewichte auf der rechten Seite
            if (weight == newWeight) { //Sack = Gewichte
                System.out.println(weight + " Gramm schwerer Sack auf
der linken Seite und " + minWeight + ", " + weights.get(add1 - 1) + "
und " + weights.get(i) + " Gramm schweres Gewicht auf der rechten
Seite\n");
                return true;
            } else if (weight < newWeight) { //Sack < Gewichte
                if (i != 0) {
                    return addLowerWeight2(weight, minWeight, add1, i);
                } else {
                    return false;
                }
            }
        }
    }
    return false;
}

private boolean addLowerWeight2(int weight, int minWeight, int add1, int
add2) {
    for (int i = 0; i < weights.size() - 1; i++) { //i = Gewichts-Index
        der Liste
        if (i != add1 && i != add2) { //Nicht das gleiche Gewicht
            int newWeight = weights.get(i) + minWeight +
weights.get(add1) + weights.get(add2); //Die Gewichte auf der rechten
Seite
            if (weight == newWeight) { //Sack = Gewichte
                System.out.println(weight + " Gramm schwerer Sack auf
der linken Seite und " + minWeight + ", " + weights.get(i) + ", " +
weights.get(add1) + " und " + weights.get(add2) + " Gramm schweres
Gewicht auf der rechten Seite\n");
                return true;
            } else if (weight < newWeight) { //Sack < Gewichte
                if (i != 0) {
                    return addLowerWeight3(weight, minWeight, add1,
add2, i);
                } else {
                    return false;
                }
            }
        }
    }
    return false;
}
}
...

```

- Sollte das nicht funktionieren, versucht man zu dem Sack Gewichte zu legen, um so die Waage ins Gleichgewicht zu bekommen

```

private boolean addHigherWeight(int weight, int higherIndex) {
    int max = weights.get(higherIndex); //Die Gewichte auf der rechten
Seite
    for (int i = 0; i < weights.size() - 1; i++) { //i = Gewichts-Index
der Liste
        int newSack = weight + weights.get(i);
        if (newSack == max) { //Sack + Gewicht = Gewicht
            System.out.println(weight + " Gramm schwerer Sack und " +
weights.get(i) + " Gramm schweres Gewicht auf der linken Seite und " +
max + " Gramm schweres Gewicht auf der rechten Seite\n");
            return true;
        } else if (newSack < max) { //Sack + Gewicht < Gewicht
            System.out.println(weight + " Gramm schwerer Sack und " +
weights.get(i) + " Gramm schweres Gewicht sind leichter als " + max + "
Gramm schweres Gewicht");
            if (i != 0) {
                return addHigherWeight1(weight, higherIndex, i);
            }
        }
    }
    return false;
}...

```

- Anschließend bekam man eine Ausgabe, wie es möglich ist, auf das Gewicht bzw. nah an das Gewicht heranzukommen

```

10 Gramm schwerer Sack auf der linken Seite und 10 Gramm schweres
Gewicht auf der rechten Seite
20 Gramm schwerer Sack ist leichter als 50 Gramm schweres Gewicht
20 Gramm schwerer Sack auf der linken Seite und 10 und 10 Gramm schweres
Gewicht auf der rechten Seite
30 Gramm schwerer Sack ist leichter als 50 Gramm schweres Gewicht
30 Gramm ist leichter als 60 Gramm schweres Gewicht
30 Gramm schwerer Sack und 10 Gramm schweres Gewicht sind leichter als
50 Gramm schweres Gewicht
30 Gramm schwerer Sack und 10 Gramm schweres Gewicht sind leichter als
50 Gramm schweres Gewicht
30 Gramm schwerer Sack, 10 und 10 Gramm schweres Gewicht auf der linken
Seite und 50 Gramm schweres Gewicht auf der rechten Seite
40 Gramm schwerer Sack ist leichter als 50 Gramm schweres Gewicht
40 Gramm ist leichter als 60 Gramm schweres Gewicht
40 Gramm schwerer Sack und 10 Gramm schweres Gewicht auf der linken
Seite und 50 Gramm schweres Gewicht auf der rechten Seite
50 Gramm schwerer Sack auf der linken Seite und 50 Gramm schweres
Gewicht auf der rechten Seite

```