DA3

```
import java.util.Arrays;
import java.util.Comparator;

class Item {
    int weight;
    int value;
    double valuePerWeight;

    public Item(int weight, int value) {
        this.weight = weight;
        this.value = value;
        this.valuePerWeight = (double) value / weight;
    }
}

public class FractionalKnapsack {

    public static double knapsack(int[] weights, int[] values, int capacity) {
        int n = weights.length;
        Item[] items = new Item[n];

        for (int i = 0; i < n; i++) {
            items[i] = new Item(weights[i], values[i]);
        }

        // Sort items by their value per weight in descending order (greedy step)
        Arrays.sort(items, Comparator.comparing((Item item) -> item.valuePerWeight).reversed());
```

```
    double maxValue = 0;
    int currentWeight = 0;

    for (int i = 0; i < n; i++) {
        if (currentWeight + items[i].weight <= capacity) {
            // Take the whole item
            maxValue += items[i].value;
            currentWeight += items[i].weight;
        } else {
            // Take a fraction of the item to fill the knapsack
            double remainingCapacity = capacity - currentWeight;
            maxValue += items[i].valuePerWeight * remainingCapacity;
            break;
        }
    }
    return maxValue;
}

public static void main(String[] args) {
    int[] weights = {10, 20, 30};
    int[] values = {60, 100, 120};
    int capacity = 50;
    double maxValue = knapsack(weights, values, capacity);
    System.out.println("Maximum value that can be obtained = " + maxValue);
}
```

Output

```
java -cp /tmp/v9O7WjanqH FractionalKnapsack
Maximum value that can be obtained = 240.0
```

}