

## Aufgabe a2x1

```
public void playBlackJack(){

    while (Punkte < 17) {
        card = deck.deal();
        System.out.printf("%s %n", card);
        Rank currentRank = card.getRank();
        switch (currentRank) {
            case JACK, TEN, QUEEN, KING: Punkte +=10; break;
            case ACE: Punkte +=11; break;
            case TWO: Punkte +=2; break;
            case THREE: Punkte +=3; break;
            case FOUR: Punkte +=4; break;
            case FIVE: Punkte +=5; break;
            case SIX: Punkte +=6; break;
            case SEVEN: Punkte +=7; break;
            case EIGHT: Punkte +=8; break;
            case NINE: Punkte +=9; break;
            default: assert (false) : "ERROR: Something went wrong...";
        }

    }

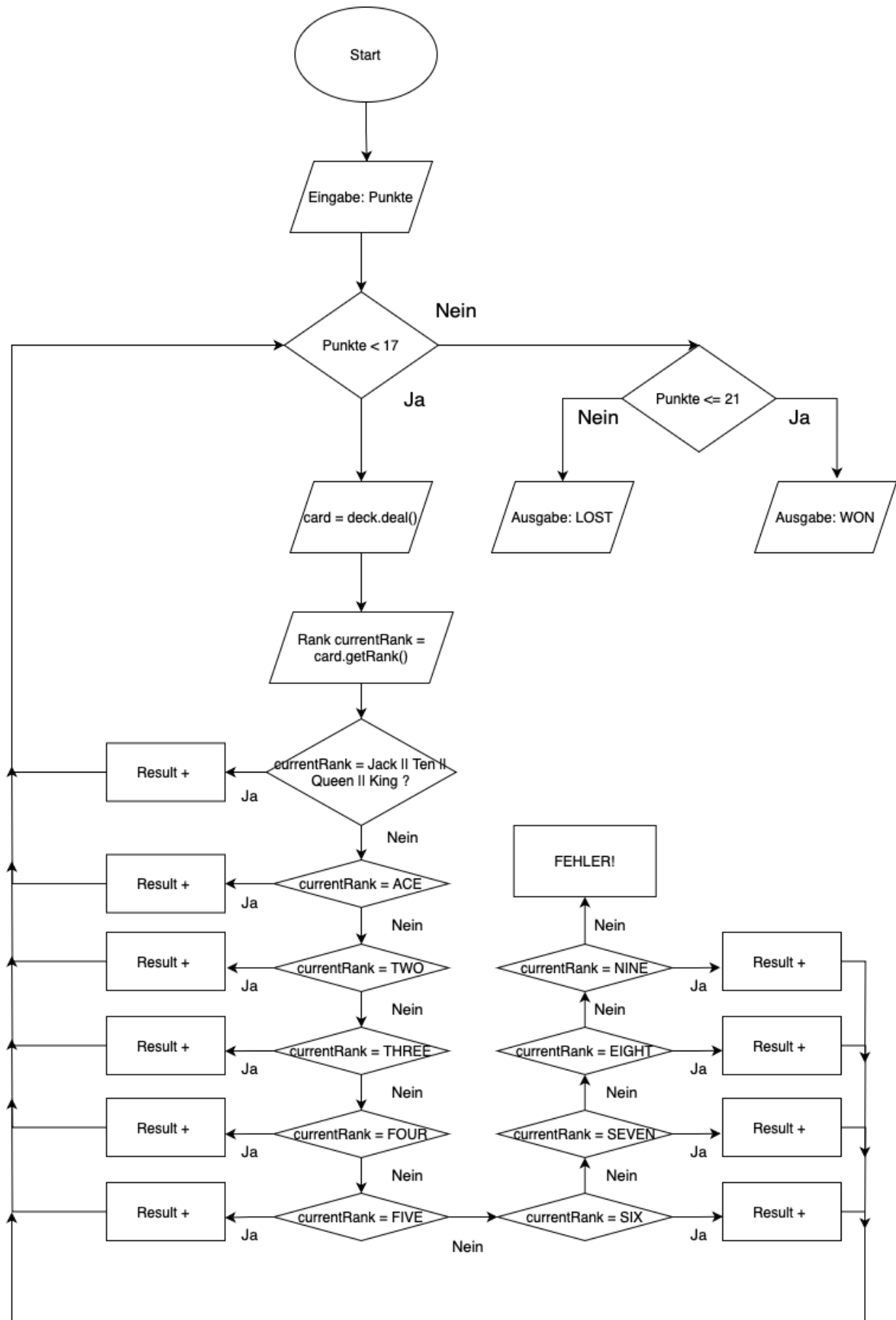
    if (Punkte <= 21) {
        System.out.printf("Wow! %2d Points!", Punkte);
    }
}
```

```
else System.out.printf("LOST: %2d Points", Punkte);
```

```
System.out.flush();
```

```
//method()
```

1. Solange die Punktzahl weniger als 17 beträgt, wird die Schleife wiederholt.
2. Der Rang der gezogenen Karte wird gespeichert.
3. Mit einer switch-Selektion wird geprüft, wie viele Punkte der aktuelle Rang wert ist, und dieser Wert wird zur Punktzahl addiert. Falls kein gültiger Rang gefunden wird, führt dies zum Abbruch des Programms mit einer Fehlermeldung.
4. Sobald die Punktzahl 17 oder mehr beträgt, wird die Schleife nicht weiter ausgeführt.
5. Anschließend wird geprüft, ob die gesammelten Punkte kleiner oder gleich 21 sind: Falls ja → "Won", Falls mehr als 21 → "Lost".



## Aufgabe a2x2

```
public class LastThree {

    int value1;
    int value2;
    int value3;
    int counter = 0;

    public LastThree(){
        value1 = 0;
        value2 = 0;
        value3 = 0;
        counter = 0;
    } //method()

    /**
     * print the last three numbers/values
     */
    public void printLastThree(){
        if (counter == 0) {
            System.out.print("Sie haben noch keine Werte eingegeben...:(");
        }
        else if (counter == 1) {
            System.out.printf("Sie haben nur einen Wert eingegeben: %d",
value1);
```

```

    }
    else if (counter == 2) {
        System.out.printf("Erste eingabe ist %d, die zweite ist %d",
value2, value1);
    }
    else if (counter == 3) {
        System.out.printf("pos1: %d, pos2: %d, pos3: %d", value3,
value2, value1);
    }

```

```

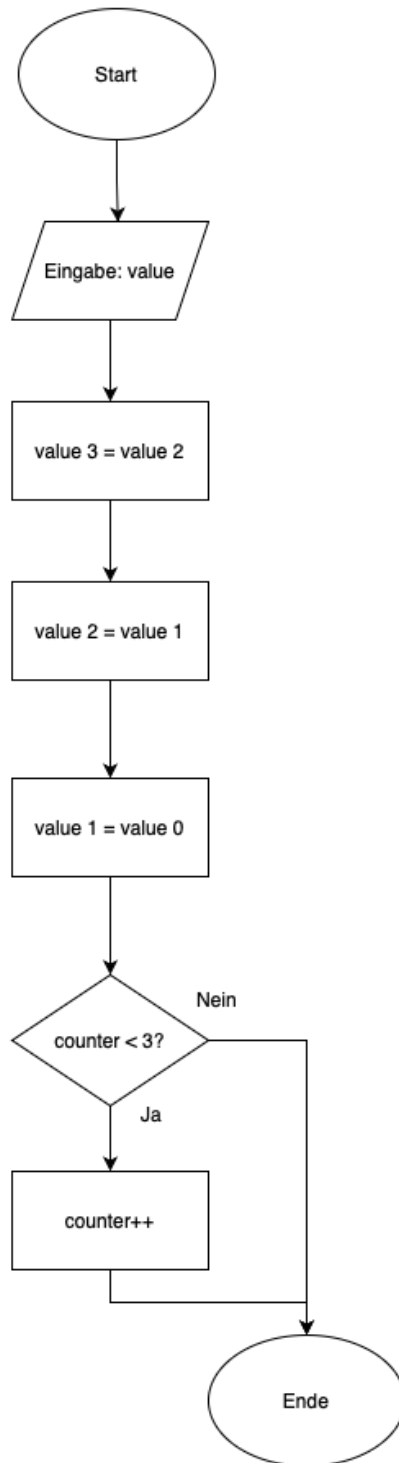
public void processNewValue( int value ){

    value3 = value2;
    value2 = value1;
    value1 = value;
    if (counter < 3) counter++;
}

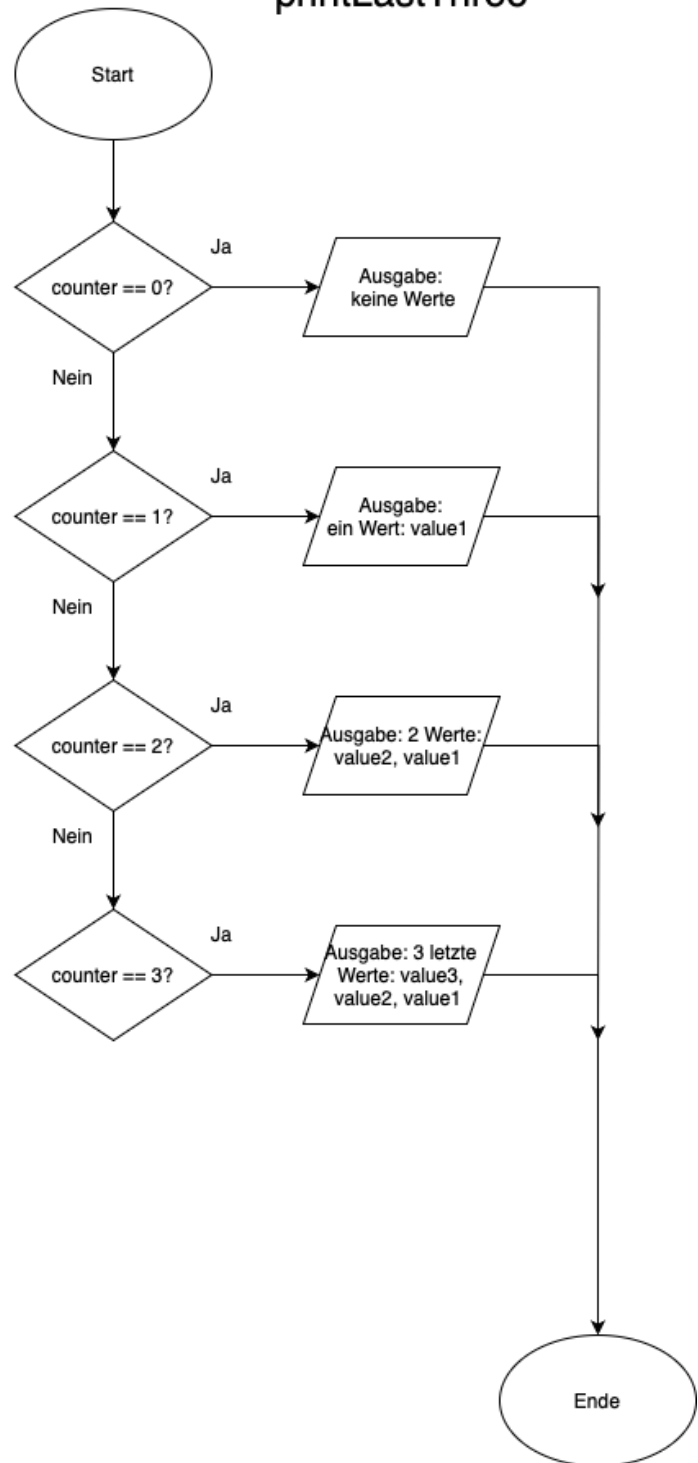
```

1. value3 bekommt den vorherigen Wert von value2,
2. value2 bekommt den vorherigen Wert von value1,
3. value1 speichert den neuen Wert.
4. Falls weniger als drei Werte eingegeben wurden, wird ein interner Zähler (counter) erhöht, damit bei der Ausgabe nur die Werte angezeigt werden, die bereits in den Variablen gespeichert sind.

## prozessNewValue



## printLastThree



## Aufgabe a2x3

```
public void printFactorization( long number ){

    if (number > 1) {
        while (number > 1) {
            int i = 2;
            while (number%i!=0) {
                i++;
            }
            System.out.print(i);
            number = number / i;
            if (number!=1) System.out.print(" * ");
        }
    }

    else assert (false) : "Deine Zahl ist zu klein :(";

    System.out.flush();
}
```

1. Die gegebene Zahl wird in ihre kleinsten Primfaktoren zerlegt.
2. Solange der aktuelle Teiler die Zahl nicht ohne Rest teilt, wird der Teiler erhöht.
3. Sobald ein passender Teiler gefunden wurde, wird die Zahl durch diesen geteilt und der Teiler wird ausgegeben.
4. Dieser Vorgang wiederholt sich, bis die Zahl vollständig faktorisiert ist (1 am Ende bleibt).

