

# Erläuterung Clean Code der Anwendung Zahlenraten

## 1. Auslagerung der Funktionen in Methoden

### Alter Code:

```
public static void main(String[] args) {  
  
    Scanner s = new Scanner(System.in);  
    System.out.print("Gib eine Zahl ein ");  
    int x = s.nextInt();  
  
    int zufall = (int) (Math.random() * 10) + 1;  
  
    while (x <= 10) {  
  
        if (x > zufall) {  
            System.out.println("Zahl ist zu groß");  
            System.out.print("Gib eine Zahl ein");  
            x = s.nextInt();  
  
        } else if (x < zufall) {  
            System.out.println("Zahl ist zu klein");  
            System.out.print("Gib eine Zahl ein");  
            x = s.nextInt();  
  
        } else {  
            System.out.println("Richtige Zahl");  
            break;  
        }  
  
    }  
    s.close();  
}
```

Im alten Code waren alle Funktionen in der Main-Methode implementiert und es gab keine ausgelagerten Funktionen.

### Neuer Code:

```
public void initaleAusgabe() {  
    System.out.print("Gib eine Zahl zwischen 1 und 10  
ein: ");  
}  
  
public int getBenutzereingabe() {  
    return scanner.nextInt();  
}  
  
public void eingabeZuGross() {  
    System.out.println("Diese Zahl ist zu groß.");  
    System.out.print("Gib eine neue Zahl ein: ");  
}
```

Der veränderte Code beinhaltet nun mehrere kleine Methoden, die nur für bestimmte Funktionen der Anwendung zuständig sind.

## 2. Methoden und Variablen selbsterklärend benannt

### Alter Code:

```
int x = s.nextInt();

int zufall = (int) (Math.random() * 10) + 1;

while (x <= 10) {

    if (x > zufall) {
        System.out.println("Zahl ist zu groß");
        System.out.print("Gib eine Zahl ein");
        x = s.nextInt();
    }
}
```

Die Variablen waren nicht sehr aussagekräftig benannt. *x* steht zum Beispiel für eine vom Benutzer eingegebene Zahl und *zufall* für eine Zufallszahl.

### Neuer Code:

```
public Zufallszahl() {
    int maxZahl = 10;
    zufallszahl = (int) (Math.random() * maxZahl) + 1;
}

public int zahlenVergleich(int vergleichzahl) {
    if (vergleichzahl > zufallszahl) {
        return 1;
    } else if (vergleichzahl < zufallszahl) {
        return -1;
    } else {
        return 0;
    }
}
```

Die Variablen sind nun selbsterklärend benannt und auch die Methoden sind verständlich und aussagekräftig.

### 3. Geringe Komplexität (KISS)

#### Alter Code:

```
public static void main(String[] args) {  
  
    Scanner s = new Scanner(System.in);  
    System.out.print("Gib eine Zahl ein ");  
    int x = s.nextInt();  
  
    int zufall = (int) (Math.random() * 10) + 1;  
  
    while (x <= 10) {  
  
        if (x > zufall) {  
            System.out.println("Zahl ist zu groß");  
            System.out.print("Gib eine Zahl ein");  
            x = s.nextInt();  
  
        } else if (x < zufall) {  
            System.out.println("Zahl ist zu klein");  
            System.out.print("Gib eine Zahl ein");  
            x = s.nextInt();  
  
        } else {  
            System.out.println("Richtige Zahl");  
            break;  
        }  
  
    }  
    s.close();  
}
```

Alle Aufgaben der Anwendung waren in einer Klasse und einer Methode implementiert, sodass es schnell unübersichtlich werden kann.

#### Neuer Code:

```
public void eingabeZuGross() {  
    System.out.println("Diese Zahl ist zu groß.");  
    System.out.print("Gib eine neue Zahl ein: ");  
}  
  
public void eingabeZuKlein() {  
    System.out.println("Diese Zahl ist zu klein.");  
    System.out.print("Gib eine neue Zahl ein: ");  
}  
  
public void richtigeZahl() {  
    System.out.println("Glückwunsch! Das war die  
richtige Zahl");  
}
```

Jede Methode hat eine Aufgabe und beinhaltet nur wenig Code. Das Programm wird dadurch einfacher und übersichtlicher.

## 4. Aufteilung in Klassen

### Alter Code:

```
public static void main(String[] args) {

    Scanner s = new Scanner(System.in);
    System.out.print("Gib eine Zahl ein ");
    int x = s.nextInt();

    int zufall = (int) (Math.random() * 10) + 1;

    while (x <= 10) {

        if (x > zufall) {
            System.out.println("Zahl ist zu groß");
            System.out.print("Gib eine Zahl ein");
            x = s.nextInt();

        } else if (x < zufall) {
            System.out.println("Zahl ist zu klein");
            System.out.print("Gib eine Zahl ein");
            x = s.nextInt();

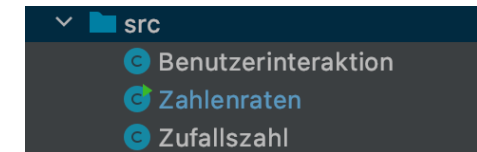
        } else {
            System.out.println("Richtige Zahl");
            break;
        }

    }

    s.close();
}
```

Der alte Code bestand, wie bereits erwähnt, aus genau einer Klasse, in der alle Funktionen implementiert wurden.

### Neuer Code:



```
public class Zufallszahl {

    private int zufallszahl;

    public Zufallszahl() {
        int maxZahl = 10;
        zufallszahl = (int) (Math.random() * maxZahl) +
1;
    }

    public int zahlenVergleich(int vergleichzahl) {
        if (vergleichzahl > zufallszahl) {
            return 1;
        } else if (vergleichzahl < zufallszahl) {
            return -1;
        } else {
            return 0;
        }
    }

}
```

Der neue Code besteht aus drei Klassen, die alle jeweils eine Aufgabe haben. Die Klasse *Zufallszahl* legt zum Beispiel nur die Zufallszahl fest und vergleicht diese Zahl mit der Benutzereingabe, die sie aus der Klasse *Benutzerinteraktion* bekommt. Dort werden auch Konsolenausgaben gesetzt. In der Klasse *Zahlenraten* werden die beiden Klassen dann miteinander verbunden.

## **5. Verwendung eines Versionskontrollsystems**

### **Alter Code:**

Der alte Code war rein lokal bei mir gespeichert und wurde in keinem Versionskontrollsystem verwaltet.

### **Neuer Code:**

Der neue Code wird nun in GitHub verwaltet.