

# Lösungsvorschlag: Inhalte in Methoden auslagern

## 1. Menüauswahl in Methode auslagern

Wir haben uns vorgenommen, dass wir Duplikate im Code vermeiden möchten. Im Programm `UserDialog` listen wir dem User an zwei Stellen im Code seine Auswahlmöglichkeiten auf:

```
// Menue Anzeigen
System.out.println();
System.out.println("=====");
System.out.println("Was möchten Sie mit dieser Applikation machen?");
System.out.println("Wählen Sie 'r' für Taschenrechner");
System.out.println("Wählen Sie 'h' für Hasen und Hühner Problem");
System.out.println("Wählen Sie 's' für Soccermanager");
System.out.println("Wählen Sie 'x' um das Programm zu beenden!");
System.out.print(">> ");
```

Wir möchten diesen Programmteil nun in eine externe Methode packen. Wir arbeiten uns also durch unsere 3 Schritte durch:

### 1. Schnittstelle vereinbaren

- **Sichtbarkeit:** Die Methode wird nur innerhalb der Klasse gebraucht, es reicht also `private`
- **Rückgabewert:** Die Methode macht mehrere `System.out.print` gibt aber keinen Wert zurück, also `void`
- **Name:** Die Methode druckt die Menüauswahl auf die Konsole aus, mögliche Namen wären z.B `printMenue`, `printDialog`, etc.
- **Parameter:** Der Methode müssen keine Informationen übergeben werden. Die Parameterliste ist also Leer ( )

```
private void printMenue(){}
```

### 2. Methodenkopf schreiben

Ein kurzer beschreibender Satz.

```
/**
 * Diese Methode zeigt dem User das Menue an.
 */
private void printMenue(){}
```

### 3. Logik Programmieren

#### UserDialog.java

```
import java.util.Scanner;
/**
 * UserDialog-Klasse um die Verwendung von Methoden zu üben
 *
 * @author Kevin Maurizi
 * @since 2042.01.01
 * @version 0.1
 */
public class UserDialog {

    static Scanner scanner;

    public UserDialog(){
        scanner = new Scanner(System.in);
    }

    public static void main(String[] args) {
        UserDialog userDialog = new UserDialog();
        userDialog.run();
        scanner.close();
    }

    private void run() {
        char auswahl;
        double zahl1, zahl2, summe;
        int shirtNo, countRating;
        double totalRating, rating;
        int animals, legs, rabbit, chicken, maxLegs, minLegs;

        System.out.println("Lieber Benutzer, herzlich willkommen zu diesem Programm");

        printMenue();
        auswahl = scanner.nextLine().charAt(0);

        while (auswahl != 'x'){
            if(auswahl == 'r'){

                System.out.println("Bitte geben Sie die erste Zahl ein");

                while(!scanner.hasNextDouble()){
                    System.out.println("Bitte eine Zahl eingeben");
                    scanner.nextLine();
                }
            }
        }
    }
}
```

```
    zahl1 = scanner.nextDouble();
    scanner.nextLine();

    System.out.println("Bitte geben Sie die zweite Zahl
ein");

    while(!scanner.hasNextDouble()){
        System.out.println("Bitte eine Zahl eingeben");
        scanner.nextLine();
    }
    zahl2 = scanner.nextDouble();
    scanner.nextLine();

    summe = zahl1 + zahl2;
    System.out.println("Die Summe ist: " + summe);

}else if(auswahl == 'h'){

    System.out.print("Anzahl Tiere >");
    while(!scanner.hasNextInt()){
        System.out.print("Bitte eine Zahl eingeben. Anzahl
Tiere >");
        scanner.nextLine();
    }
    animals = scanner.nextInt();
    scanner.nextLine();

    maxLegs = animals * 4;
    minLegs = animals *2;

    System.out.print("Anzahl Beine >");
    while(!scanner.hasNextInt()){
        System.out.print("Bitte eine Zahl eingeben. Anzahl
Beine >");
        scanner.nextLine();
    }

    legs = scanner.nextInt();
    scanner.nextLine();

    while(legs > maxLegs || legs < minLegs){
        System.out.print("Anzahl Beine ungültig, bitte
erneut eingeben (Min: "+minLegs +" Max: " + maxLegs +" ) >");
        while(!scanner.hasNextInt()){
            System.out.print("Bitte eine Zahl eingeben.
Anzahl Beine >");
            scanner.nextLine();
        }
        legs = scanner.nextInt();
        scanner.nextLine();
    }
}
```

```
        rabbit = animals;
        chicken = 0;

        while ((rabbit * 4 + chicken * 2) > legs) {
            rabbit = rabbit - 1;
            chicken = chicken + 1;
        }

        System.out.println("Es sind " + rabbit + " Hasen und "
+ chicken + " Huehner");

    }else if(auswahl == 's'){

        System.out.print ("Trikotnummer (1 - 99 / 0=Ende) >");
        shirtNo = scanner.nextInt ();
        scanner.nextLine();

        while (shirtNo != 0) {
            totalRating = 0.0;
            countRating = 0;
            System.out.print ("Bewertung (1.0 - 6.0 / 0=Ende)
>");

            rating = scanner.nextDouble ();
            scanner.nextLine();

            while (rating > 0) {
                totalRating = totalRating + rating;
                countRating = countRating + 1;
                System.out.print ("Bewertung (1.0 - 6.0 /
0=Ende) >");

                rating = scanner.nextDouble ();
                scanner.nextLine();

            }
            System.out.println ("Spieler: " + shirtNo + " /
Durchschnitt: " + (totalRating / countRating));
            System.out.print ("Trikotnummer (1 - 99 / 0=Ende)
>");

            shirtNo = scanner.nextInt ();
            scanner.nextLine();

        }
    }else{

        System.out.println("Buchstabe nicht bekannt, bitte
erneut eingeben!");

    }
}
```

```

        printMenue();
        auswahl = scanner.nextLine().charAt(0);
    }
}

/**
 * Diese Methode zeigt dem User das Menue an.
 */
private void printMenue(){
    System.out.println();
System.out.println("=====");
    System.out.println("Was möchten Sie mit dieser Applikation
machen?");
    System.out.println("Wählen Sie 'r' für Taschenrechner");
    System.out.println("Wählen Sie 'h' für Hasen und Hühner
Problem");
    System.out.println("Wählen Sie 's' für Soccermanager");
    System.out.println("Wählen Sie 'x' um das Programm zu
beenden!");
    System.out.print(">> ");
}
}

```

## 2. Unterprogramme auslagern

Um die Übersichtlichkeit des Codes zu erhöhen, lagern wir die Unterprogramme (Rechner, Hasen- und Hühnerproblem und Soccermanager) in eigene Methoden aus.

### 1. Schnittstelle vereinbaren

	Sichtbarkeit	Rückgabewert	Name	Parameter	Methodenkopf
<b>Rechner</b>	private	void	rechner	()	private void rechner(){}
<b>Hasen &amp; Hühner</b>	private	void	hasenUndHuehner	()	private void hasenUndHuehner(){}
<b>Soccermanager</b>	private	void	soccermanager	()	private void soccermanager(){}

### 2. Methodenkopf schreiben

	Methodenkopf
<b>Rechner</b>	Verrechnet zwei Werte zu einer Summe
<b>Hasen &amp; Hühner</b>	Löst das Hasen- und Hühnerproblem
<b>Soccermanager</b>	Anhand der Spielernummern können Bewertungen vergeben werden

### 3. Logik Programmieren

#### UserDialog.java

```
import java.util.Scanner;
/**
 * UserDialog-Klasse um die Verwendung von Methoden zu üben
 *
 * @author Kevin Maurizi
 * @since 2042.01.01
 * @version 0.1
 */
public class UserDialog {

    static Scanner scanner;

    public UserDialog(){
        scanner = new Scanner(System.in);
    }

    public static void main(String[] args) {
        UserDialog userDialog = new UserDialog();
        userDialog.run();
        scanner.close();
    }

    private void run() {
        char auswahl;

        System.out.println("Lieber Benutzer, herzlich willkommen zu diesem Programm");

        printMenue();
        auswahl = scanner.nextLine().charAt(0);

        while (auswahl != 'x'){
            if(auswahl == 'r'){
                rechner();
            }else if(auswahl == 'h'){
                hasenUndHuehner();
            }else if(auswahl == 's'){
                soccermanager();
            }else{
                System.out.println("Buchstabe nicht bekannt, bitte erneut eingeben!");
            }

            printMenue();
        }
    }
}
```

```
        auswahl = scanner.nextLine().charAt(0);
    }
}

/**
 * Diese Methode zeigt dem User das Menue an.
 */
private void printMenue(){
    System.out.println();
System.out.println("=====");
    System.out.println("Was möchten Sie mit dieser Applikation
machen?");
    System.out.println("Wählen Sie 'r' für Taschenrechner");
    System.out.println("Wählen Sie 'h' für Hasen und Hühner
Problem");
    System.out.println("Wählen Sie 's' für Soccermanager");
    System.out.println("Wählen Sie 'x' um das Programm zu
beenden!");
    System.out.print(">> ");
}

/**
 * Verrechnet zwei Werte zu einer Summe
 */
private void rechner(){

    double zahl1, zahl2, summe;

    System.out.println("Bitte geben Sie die erste Zahl ein");
    while(!scanner.hasNextDouble()){
        System.out.println("Bitte eine Zahl eingeben");
        scanner.nextLine();
    }
    zahl1 = scanner.nextDouble();
    scanner.nextLine();

    System.out.println("Bitte geben Sie die zweite Zahl ein");
    while(!scanner.hasNextDouble()){
        System.out.println("Bitte eine Zahl eingeben");
        scanner.nextLine();
    }
    zahl2 = scanner.nextDouble();
    scanner.nextLine();

    summe = zahl1 + zahl2;
    System.out.println("Die Summe ist: " + summe);
}

/**
 * Löst das Hasen- und Hühnerproblem
 */
```

```
private void hasenUndHuehner(){
    int animals, legs, rabbit, chicken, maxLegs, minLegs;

    System.out.print("Anzahl Tiere >");
    while(!scanner.hasNextInt()){
        System.out.print("Bitte eine Zahl eingeben. Anzahl Tiere
>");
        scanner.nextLine();
    }
    animals = scanner.nextInt();
    scanner.nextLine();

    maxLegs = animals * 4;
    minLegs = animals * 2;

    System.out.print("Anzahl Beine >");
    while(!scanner.hasNextInt()){
        System.out.print("Bitte eine Zahl eingeben. Anzahl Beine
>");
        scanner.nextLine();
    }

    legs = scanner.nextInt();
    scanner.nextLine();

    while(legs > maxLegs || legs < minLegs){
        System.out.print("Anzahl Beine ungültig, bitte erneut
eingeben (Min: "+minLegs+" Max: "+maxLegs+" ) >");
        while(!scanner.hasNextInt()){
            System.out.print("Bitte eine Zahl eingeben. Anzahl
Beine >");
            scanner.nextLine();
        }
        legs = scanner.nextInt();
        scanner.nextLine();
    }

    rabbit = animals;
    chicken = 0;

    while ((rabbit * 4 + chicken * 2) > legs) {
        rabbit = rabbit - 1;
        chicken = chicken + 1;
    }

    System.out.println("Es sind " + rabbit + " Hasen und " +
chicken + " Huehner");
}
```



```

/**
 * Anhand der Spielernummern können Bewertungen vergeben werden
 */
private void soccermanager() {
    int shirtNo, countRating;
    double totalRating, rating;

    System.out.print("Trikotnummer (1 - 99 / 0=Ende) >");
    shirtNo = scanner.nextInt();
    scanner.nextLine();

    while (shirtNo != 0) {
        totalRating = 0.0;
        countRating = 0;
        System.out.print("Bewertung (1.0 - 6.0 / 0=Ende) >");
        rating = scanner.nextDouble();
        scanner.nextLine();

        while (rating > 0) {
            totalRating = totalRating + rating;
            countRating = countRating + 1;
            System.out.print("Bewertung (1.0 - 6.0 / 0=Ende) >");
            rating = scanner.nextDouble();
            scanner.nextLine();
        }
        System.out.println("Spieler: " + shirtNo + " /
Durchschnitt: " + (totalRating / countRating));
        System.out.print("Trikotnummer (1 - 99 / 0=Ende) >");
        shirtNo = scanner.nextInt();
        scanner.nextLine();
    }
}
}
}

```

### 3. Inputvalidierung auslagern

Wir haben noch duplizierten Code in unserer Anwendung.

Sowohl in der Methode rechner() wie auch in der Methode hasenUndHuehner() machen wir eine Inputvalidierung. Wir überprüfen mehrfach, ob ein `int` respektive ein `double` eingegeben wurde. Wir können diese Funktionalität auslagern und so einfach wiederverwenden.

#### 1. Schnittstelle vereinbaren

	Sichtbarkeit	Rückgabewert	Name	Parameter	Methodenkopf
--	--------------	--------------	------	-----------	--------------

	Sichtbarkeit	Rückgabewert	Name	Parameter	Methodenkopf
<b>double-Validierung</b>	private	double	getValidDouble	()	private double getValidDouble(){}
<b>int-Validierung</b>	private	int	getValidInt	()	private int getValidInt(){}

## 2. Methodenkopf schreiben

	Methodenkopf	@return
<b>double-Validierung</b>	Prüft ob ein Double eingegeben wurde, wenn nicht, wird ein Fehler ausgegeben und erneut eingelesen.	gibt das eingelesene Double zurück.
<b>int-Validierung</b>	Prüft ob ein Int eingegeben wurde, wenn nicht, wird ein Fehler ausgegeben und erneut eingelesen.	gibt das eingelesene Int zurück.

Beispiel:

```
/**
 * Prüft ob ein double eingegeben wurde, wenn nicht, wird ein Fehler
 ausgegeben und erneut eingelesen.
 * @return gibt das eingelesene Double zurück.
 */
private double getValidDouble(){}
```

## 3. Logik Programmieren

UserDialog.java

```
import java.util.Scanner;
/**
 * UserDialog-Klasse um die Verwendung von Methoden zu üben
 *
 * @author Kevin Maurizi
 * @since 2042.01.01
 * @version 0.1
 */
public class UserDialog {

    static Scanner scanner;

    public UserDialog(){
        scanner = new Scanner(System.in);
    }

    public static void main(String[] args) {
        UserDialog userDialog = new UserDialog();
        userDialog.run();
        scanner.close();
    }
}
```

```
private void run() {
    char auswahl;

    System.out.println("Lieber Benutzer, herzlich willkommen zu
diesem Programm");

    printMenue();
    auswahl = scanner.nextLine().charAt(0);

    while (auswahl != 'x'){
        if(auswahl == 'r'){
            rechner();
        }else if(auswahl == 'h'){
            hasenUndHuehner();
        }else if(auswahl == 's'){
            soccermanager();
        }else{
            System.out.println("Buchstabe nicht bekannt, bitte
erneut eingeben!");
        }

        printMenue();
        auswahl = scanner.nextLine().charAt(0);
    }
}

/**
 * Diese Methode zeigt dem User das Menue an.
 */
private void printMenue(){
    System.out.println();
System.out.println("=====");
    System.out.println("Was möchten Sie mit dieser Applikation
machen?");
    System.out.println("Wählen Sie 'r' für Taschenrechner");
    System.out.println("Wählen Sie 'h' für Hasen und Hühner
Problem");
    System.out.println("Wählen Sie 's' für Soccermanager");
    System.out.println("Wählen Sie 'x' um das Programm zu
beenden!");
    System.out.print(">> ");
}

/**
 * Verrechnet zwei Werte zu einer Summe
 */
private void rechner(){

    double zahl1, zahl2, summe;
```

```
System.out.println("Bitte geben Sie die erste Zahl ein");
zahl1 = getValidDouble();

System.out.println("Bitte geben Sie die zweite Zahl ein");
zahl2 = getValidDouble();

summe = zahl1 + zahl2;
System.out.println("Die Summe ist: " + summe);
}

/**
 * Löst das Hasen- und Hühnerproblem
 */
private void hasenUndHuehner(){
    int animals, legs, rabbit, chicken, maxLegs, minLegs;

    System.out.print("Anzahl Tiere >");
    animals = getValidInt();

    maxLegs = animals * 4;
    minLegs = animals * 2;

    System.out.print("Anzahl Beine >");
    legs = getValidInt();

    while(legs > maxLegs || legs < minLegs){
        System.out.print("Anzahl Beine ungültig, bitte erneut
eingeben (Min: "+minLegs+" Max: "+maxLegs+" ) >");
        legs = getValidInt();
    }

    rabbit = animals;
    chicken = 0;

    while ((rabbit * 4 + chicken * 2) > legs) {
        rabbit = rabbit - 1;
        chicken = chicken + 1;
    }

    System.out.println("Es sind " + rabbit + " Hasen und " +
chicken + " Huehner");
}

/**
 * Anhand der Spielernummern können Bewertungen vergeben werden
 */
private void soccermanager() {
    int shirtNo, countRating;
    double totalRating, rating;
```

```
System.out.print("Trikotnummer (1 - 99 / 0=Ende) >");
shirtNo = scanner.nextInt();
scanner.nextLine();

while (shirtNo != 0) {
    totalRating = 0.0;
    countRating = 0;
    System.out.print("Bewertung (1.0 - 6.0 / 0=Ende) >");
    rating = scanner.nextDouble();
    scanner.nextLine();

    while (rating > 0) {
        totalRating = totalRating + rating;
        countRating = countRating + 1;
        System.out.print("Bewertung (1.0 - 6.0 / 0=Ende) >");
        rating = scanner.nextDouble();
        scanner.nextLine();
    }
    System.out.println("Spieler: " + shirtNo + " /
Durchschnitt: " + (totalRating / countRating));
    System.out.print("Trikotnummer (1 - 99 / 0=Ende) >");
    shirtNo = scanner.nextInt();
    scanner.nextLine();
}

/**
 * Prüft ob ein Double eingegeben wurde, wenn nicht, wird ein
Fehler ausgegeben und erneut eingelesen.
 * @return gibt das eingelesene Double zurück.
 */
private double getValidDouble(){
    double zahl;
    while(!scanner.hasNextDouble()){
        System.out.print("Bitte eine Zahl eingeben.");
        scanner.nextLine();
    }
    zahl = scanner.nextDouble();
    scanner.nextLine();
    return zahl;
}

/**
 * Prüft ob ein Int eingegeben wurde, wenn nicht, wird ein Fehler
ausgegeben und erneut eingelesen.
 * @return gibt das eingelesene Int zurück.
 */
private int getValidInt(){
    int zahl;
```

```
        while(!scanner.hasNextInt()){
            System.out.print("Bitte eine Zahl eingeben.");
            scanner.nextLine();
        }
        zahl = scanner.nextInt();
        scanner.nextLine();
        return zahl;
    }
}
```

## 4. Code wiederverwenden

```
/**
 * Anhand der Spielernummern können Bewertungen vergeben werden
 */
private void soccermanager() {
    int shirtNo, countRating;
    double totalRating, rating;

    System.out.print("Trikotnummer (1 - 99 / 0=Ende) >");
    shirtNo = getValidInt();

    while (shirtNo != 0) {
        totalRating = 0.0;
        countRating = 0;
        System.out.print("Bewertung (1.0 - 6.0 / 0=Ende) >");
        rating = getValidDouble();

        while (rating > 0) {
            totalRating = totalRating + rating;
            countRating = countRating + 1;
            System.out.print("Bewertung (1.0 - 6.0 / 0=Ende) >");
            rating = getValidDouble();
        }
        System.out.println("Spieler: " + shirtNo + " / Durchschnitt: " +
            (totalRating / countRating));
        System.out.print("Trikotnummer (1 - 99 / 0=Ende) >");
        shirtNo = getValidInt();
    }
}
```

## 5. Letzte Optimierungen

## 1. Schnittstelle vereinbaren

	Sichtbarkeit	Rückgabewert	Name	Parameter	Methodenkopf
<b>double-Validierung</b>	private	double	getValidDouble	(String ausgabe)	private double getValidDouble(String ausgabe){}
<b>int-Validierung</b>	private	int	getValidInt	(String ausgabe)	private int getValidInt(String ausgabe){}

## 2. Methodenkopf schreiben

	Methodenkopf	@return	@param ausgabe
<b>double-Validierung</b>	Gibt den Text aus dem Parameter ausgabe auf der Konsole aus. Prüft anschliessend ob ein Double eingegeben wurde, wenn nicht, wird ein Fehler ausgegeben und erneut eingelesen.	gibt das eingelesene Double zurück.	Text der zuerst ausgegeben wird.
<b>int-Validierung</b>	Gibt den Text aus dem Parameter ausgabe auf der Konsole aus. Prüft anschliessend ob ein Double eingegeben wurde, wenn nicht, wird ein Fehler ausgegeben und erneut eingelesen.	gibt das eingelesene Int zurück.	Text der zuerst ausgegeben wird.

Beispiel:

```
/**
 * Gibt den Text aus dem Parameter ausgabe auf der Konsole aus. Prüft
 * anschliessend ob ein Double eingegeben wurde, wenn nicht, wird ein Fehler
 * ausgegeben und erneut eingelesen.
 * @param ausgabe Text der zuerst ausgegeben wird.
 * @return gibt das eingelesene Double zurück.
 */
private double getValidDouble(String ausgabe){
```

## 3. Logik Programmieren

UserDialog.java

```
import java.util.Scanner;
/**
 * UserDialog-Klasse um die Verwendung von Methoden zu üben
 *
 * @author Kevin Maurizi
 * @since 2042.01.01
 * @version 0.1
 */
public class UserDialog {

    static Scanner scanner;
```

```
public UserDialog(){
    scanner = new Scanner(System.in);
}

public static void main(String[] args) {
    UserDialog userDialog = new UserDialog();
    userDialog.run();
    scanner.close();
}

private void run() {
    char auswahl;

    System.out.println("Lieber Benutzer, herzlich willkommen zu
diesem Programm");

    printMenue();
    auswahl = scanner.nextLine().charAt(0);

    while (auswahl != 'x'){
        if(auswahl == 'r'){
            rechner();
        }else if(auswahl == 'h'){
            hasenUndHuehner();
        }else if(auswahl == 's'){
            soccermanager();
        }else{
            System.out.println("Buchstabe nicht bekannt, bitte
erneut eingeben!");
        }

        printMenue();
        auswahl = scanner.nextLine().charAt(0);
    }
}

/**
 * Diese Methode zeigt dem User das Menue an.
 */
private void printMenue(){
    System.out.println();
System.out.println("=====");
    System.out.println("Was möchten Sie mit dieser Applikation
machen?");
    System.out.println("Wählen Sie 'r' für Taschenrechner");
    System.out.println("Wählen Sie 'h' für Hasen und Hühner
Problem");
    System.out.println("Wählen Sie 's' für Soccermanager");
}
```



```
        System.out.println("Wählen Sie 'x' um das Programm zu
beenden!");
        System.out.print(">> ");
    }

    /**
     * Verrechnet zwei Werte zu einer Summe
     */
    private void rechner(){

        double zahl1, zahl2, summe;

        zahl1 = getValidDouble("Bitte geben Sie die erste Zahl ein >");
        zahl2 = getValidDouble("Bitte geben Sie die zweite Zahl ein
>");

        summe = zahl1 + zahl2;
        System.out.println("Die Summe ist: " + summe);
    }

    /**
     * Löst das Hasen- und Hühnerproblem
     */
    private void hasenUndHuehner(){
        int animals, legs, rabbit, chicken, maxLegs, minLegs;

        animals = getValidInt("Anzahl Tiere >");

        maxLegs = animals * 4;
        minLegs = animals * 2;

        legs = getValidInt("Anzahl Beine >");

        while(legs > maxLegs || legs < minLegs){
            legs = getValidInt("Anzahl Beine ungültig, bitte erneut
eingeben (Min: "+minLegs +" Max: " + maxLegs +" ) >");
        }

        rabbit = animals;
        chicken = 0;

        while ((rabbit * 4 + chicken * 2) > legs) {
            rabbit = rabbit - 1;
            chicken = chicken + 1;
        }

        System.out.println("Es sind " + rabbit + " Hasen und " +
chicken + " Huehner");
    }

    /**
```

```
    * Anhand der Spielernummern können Bewertungen vergeben werden
    */
private void soccermanager() {
    int shirtNo, countRating;
    double totalRating, rating;

    shirtNo = getValidInt("Trikotnummer (1 - 99 / 0=Ende) >");

    while (shirtNo != 0) {
        totalRating = 0.0;
        countRating = 0;
        rating = getValidDouble("Bewertung (1.0 - 6.0 / 0=Ende)
>");

        while (rating > 0) {
            totalRating = totalRating + rating;
            countRating = countRating + 1;
            rating = getValidDouble("Bewertung (1.0 - 6.0 / 0=Ende)
>");
        }
        System.out.println("Spieler: " + shirtNo + " /
Durchschnitt: " + (totalRating / countRating));
        shirtNo = getValidInt("Trikotnummer (1 - 99 / 0=Ende) >");
    }
}

/**
 * Prüft ob ein Double eingegeben wurde, wenn nicht, wird ein
Fehler ausgegeben und erneut eingelesen.
 * @return gibt das eingelesene Double zurück.
 */
private double getValidDouble(){
    double zahl;
    while(!scanner.hasNextDouble()){
        System.out.print("Bitte eine Zahl eingeben >");
        scanner.nextLine();
    }
    zahl = scanner.nextDouble();
    scanner.nextLine();
    return zahl;
}

/**
 * Prüft ob ein Int eingegeben wurde, wenn nicht, wird ein Fehler
ausgegeben und erneut eingelesen.
 * @return gibt das eingelesene Int zurück.
 */
private int getValidInt(){
```

```
    int zahl;  
    while(!scanner.hasNextInt()){  
        System.out.print("Bitte eine Zahl eingeben >");  
        scanner.nextLine();  
    }  
    zahl = scanner.nextInt();  
    scanner.nextLine();  
    return zahl;  
}  
  
/**  
 * Gibt den Text aus dem Parameter ausgabe auf der Konsole aus.  
 * Prüft anschliessend ob ein Double eingegeben wurde, wenn nicht, wird  
 * ein Fehler ausgegeben und erneut eingelesen.  
 * @param ausgabe Text der zuerst ausgegeben wird.  
 * @return gibt das eingelesene Int zurück.  
 */  
private int getValidInt(String ausgabe){  
    System.out.print(ausgabe);  
    int zahl;  
    while(!scanner.hasNextInt()){  
        System.out.print("Bitte eine Zahl eingeben >");  
        scanner.nextLine();  
    }  
    zahl = scanner.nextInt();  
    scanner.nextLine();  
    return zahl;  
}  
  
/**  
 * Gibt den Text aus dem Parameter ausgabe auf der Konsole aus.  
 * Prüft anschliessend ob ein Double eingegeben wurde, wenn nicht, wird  
 * ein Fehler ausgegeben und erneut eingelesen.  
 * @param ausgabe Text der zuerst ausgegeben wird.  
 * @return gibt das eingelesene Double zurück.  
 */  
private double getValidDouble(String ausgabe){  
    System.out.print(ausgabe);  
    double zahl;  
    while(!scanner.hasNextDouble()){  
        System.out.print("Bitte eine Zahl eingeben >");  
        scanner.nextLine();  
    }  
    zahl = scanner.nextDouble();  
    scanner.nextLine();  
    return zahl;  
}  
}
```



Wie sie erkennen können, habe ich die `getValidInt()` und die `getValidDouble()` nicht gelöscht oder angepasst, sondern einfach so stehen lassen. Obwohl die neuen Methoden `getValidInt(String ausgabe)` und `getValidDouble(String ausgabe)` denselben Namen haben kann Java diese unterscheiden. Man nennt das [Methoden überladen](#)

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/modul/archiv/m319/learningunits/lu07/loesungen/methoden>

Last update: **2024/03/28 14:07**

