

OOP

Rückgabewerte von Methoden

Henrik Horstmann

14. September 2014

Inhaltsverzeichnis

| | |
|--|---|
| 1 Bedeutung der Symbole..... | 1 |
| 2 Rückgabewerte von Methoden..... | 2 |
| 3 Der freundliche Computer..... | 2 |
| 3.1 Der Programm Anfang..... | 2 |
| 3.2 Abfragen des Benutzernamen..... | 2 |
| 3.3 Den Inhalt einer Variablen auf der Konsole ausgeben..... | 5 |
| 3.4 Den Inhalt einer Variablen auf der Konsole ausgeben..... | 6 |
| 4 Lösungen zu den Aufgaben..... | 7 |
| 4.1 Lösung zu Aufgabe 1..... | 7 |
| 4.2 Lösung zu Aufgabe 2..... | 8 |
| 4.3 Lösung zu Aufgabe 3..... | 9 |

1 Bedeutung der Symbole

Symbol**Beschreibung**

Dieses Symbol kennzeichnet einen Tipp oder Hinweis.



Aufgepasst, hier passieren leicht Fehler oder es ist mit Schwierigkeiten zu rechnen.



Hier erfahren Sie, wie es gemacht wird.



Es folgt eine Aufgabe die zu lösen ist.

2 Rückgabewerte von Methoden

Es gibt Methode, wie z.B. *println* der Klasse *Computer* deren Aufruf lediglich Befehle ausführen. Das Ergebnis ist immer gleich.

Andere Methoden jedoch berechnen Werte oder fragen solche ab, die im weiteren Programm verwendet werden sollen. So sollte eine Methode, die den Benutzer auffordert eine Eingabe zu machen, diese auch an das Programm weiter geben. Die Methode *read* der Klasse *Computer* ist für die Abfrage von Benutzereingaben zuständig:

```
String read (String label)
```

Ihr Rückgabewert ist ein Objekt der Klasse *String*. *String* ist genau wie *Computer* eine Klasse, die standardmäßig in HOOPLU vorhanden ist. Die Klasse *String* speichert Zeichenketten (=Texte).

3 Der freundliche Computer

Im folgenden soll ein Programm entstehen, dass den Namen des Benutzers abfragt und diesen dann mit der Meldung

Hallo *name*!

Ich bin Dein freundlicher Computer.

begrüßt. Dabei steht *name* für den vom Benutzer eingegebenen Namen.

3.1 Der Programm Anfang



Aufgabe 1:

Erstellen Sie ein Programm, dass den Text "Hallo" auf der Konsole ausgibt und speichern Sie das Programm unter dem Namen Begrueßung.oop.

Kommentieren Sie Ihren Programmcode.

[Hier geht es zur Lösung auf Seite 7.](#)

3.2 Abfragen des Benutzernamen

Da die Abfrage des Benutzernamen, den Name in einem Objekt von der Klasse *String* speichert, benötigen wir eine Variable, in der der Name gespeichert werden kann.

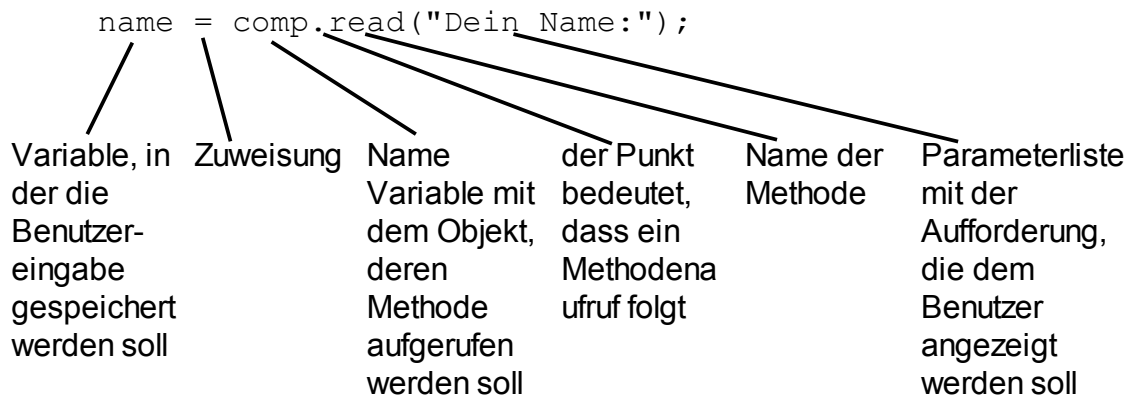


Aufgabe 2:

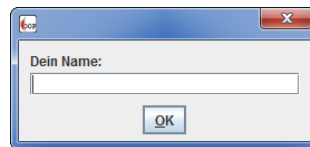
Fügen Sie der Methode *main* eine Variable *name* von der Klasse *String* hinzu.

[Hier geht es zur Lösung auf Seite 8.](#)

Im nächsten Schritt muss die Codezeile für die Benutzereingabe am Ende der Methode *main* eingefügt werden:



Der Methodenaufruf führt dazu, dass sich folgendes Dialogfenster öffnet:

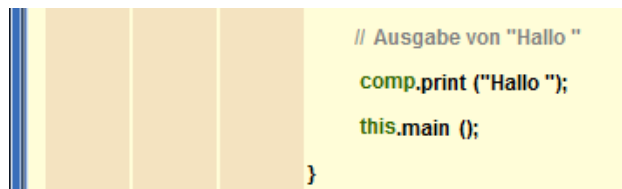


Nachdem der Benutzer seinen Namen eingegeben und mit *Ok* bestätigt hat, ist die Benutzereingabe in der Variable *name* gespeichert.

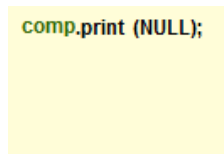
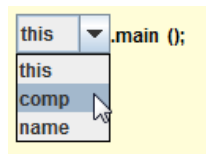
Einfügen der Programmzeile in HOOPLU:

| | |
|--|---|
| <p>Mauszeiger auf die Zeile positionieren, nach der die Codezeile eingefügt werden soll:</p> | <p>Rechte Maustaste drücken und <i>nach der Zeile</i> → <i>Methodenaufruf einfügen</i> anklicken:</p> |
|--|---|

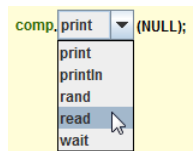
Anschließend ist ein Methodenaufruf eingefügt. Allerdings ist es nicht der gewünschte Methodenaufruf:



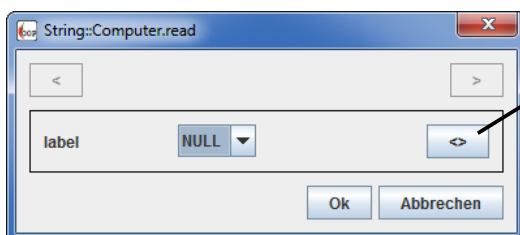
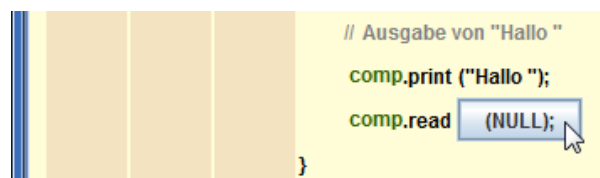
Die Variable *this* muss durch *comp* ersetzt werden. Dazu wird der Mauszeiger auf den Variablen Namen *this* positioniert. Eine Liste der zur Auswahl stehenden Variablen wird eingeblendet. Mit einem Mausklick auf *comp* ist die Variable ausgewählt:



Jetzt wird allerdings noch nicht die gewünschte Methode aufgerufen. Der Mauszeiger wird auf den Methoden Namen *print* positioniert. Eine Liste der zur Auswahl stehenden Methoden wird eingeblendet. Mit einem Mausklick auf *read* ist die Methode ausgewählt:

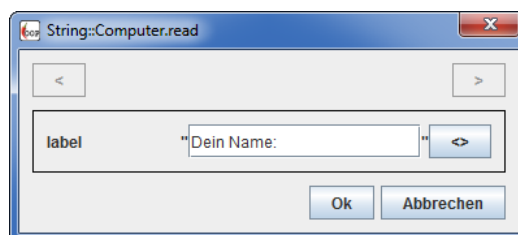


Wird der Mauszeiger auf die Parameterliste positioniert, so erscheint ein Button. Ein Mausklick auf den Button öffnet ein Dialogfenster zur Bearbeitung der Parameterliste.



Auswahlbutton, mit dem bestimmt wird, wie der Text übergeben wird.

Mit einem Mausklick auf den Auswahlbutton (siehe Abbildung), erscheint ein Textfeld, in dem der Text direkt eingeben werden kann:



Mit einem Mausklick auf *Ok* die Eingabe bestätigen.

Als letztes fehlt noch die Zuweisung des Rückgabewertes an die Variable *name*. Dazu den Mauszeiger ganz vorne auf der Codezeile positionieren. Eine Liste von Variablen öffnet sich:



aus der Liste die Variable *name* auswählen.

```
comp.print ("Hallo ");
name = comp.read ("Dein Name:");
```

Die Codezeile ist komplett. Kommentieren nicht vergessen:

```
// Name des Benutzers abfragen
name = comp.read ("Dein Name:");
```

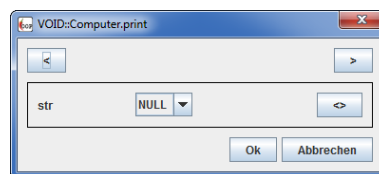
3.3 Den Inhalt einer Variablen auf der Konsole ausgeben

Das Ausgeben des Inhalts einer Variablen funktioniert prinzipiell nicht anders, als wenn Text wie "Hello world!" ausgegeben werden soll.

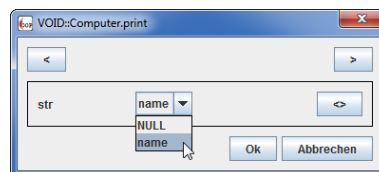
Zunächst wird eine Codezeile am Ende der *main* Methode eingefügt, in der die Methode *print* des Objektes in *comp* aufgerufen wird:

```
name = comp.read ("Dein Name:");
comp.print (NULL);
}
```

Den Mauszeiger auf die Parameterliste positionieren und mit einem Mausklick auf den nun sichtbaren Button das Dialogfenster zur Bearbeitung der Parameterliste öffnen:



In der Liste hinter *str* befinden sich alle Variablen, deren Inhalt mit *printf* ausgegeben werden kann. In unserem Fall soll der Inhalt der Variablen *name* ausgegeben werden:



Die Eingabe mit *Ok* bestätigen.

```
name = comp.read ("Dein Name:");
comp.print (name);
}
```

Damit ist die Codezeile komplett. Kommentieren nicht vergessen:

```
// Name des Benutzers ausgegeben
comp.print (name);
}
```

3.4 Den Inhalt einer Variablen auf der Konsole ausgeben



Aufgabe 3:

Fügen Sie Codezeilen ein, die den restlichen Begrüßungstext auf der Konsole ausgeben:

```
Hallo name!
```

```
Ich bin Dein freundlicher Computer.
```

Beachten Sie den Zeilenvorschub und denken Sie daran den Code zu kommentieren.

Testen Sie Ihr Programm.

[Hier geht es zur Lösung auf Seite 9.](#)

ENDE

4 Lösungen zu den Aufgaben

4.1 Lösung zu Aufgabe 1

```
public class Program
{
    // Anfang Attribute
    // Ende Attribute

    public Program ()
    {
        // Anfang Variablen
        // Ende Variablen
    }

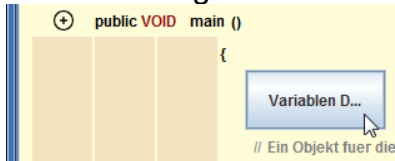
    public void main ()
    {
        // Anfang Variablen
        Computer comp;
        // Ende Variablen

        // Ein Objekt fuer die Variable comp erzeugen
        comp = new Computer();
        // Ausgabe von "Hallo "
        comp.print("Hallo ");
    }
}
```

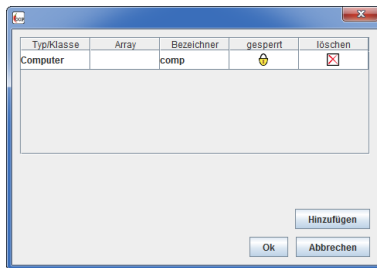
[Zurück zur Aufgabe auf Seite 2.](#)

4.2 Lösung zu Aufgabe 2

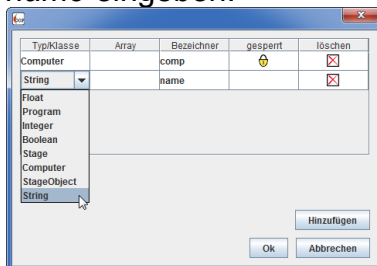
1. Den Mauszeiger auf den Variablen-Bereich positionieren:



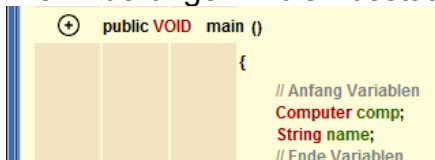
2. Mit einem Mausklick auf den Button öffnet sich das Dialogfenster zum Verwalten der Variablen:



3. Mit *Hinzufügen* eine neue Variable einfügen. Bei *Typ/Klasse* die Klasse *String* auswählen und bei *Bezeichner* den Namen der Variablen, somit *name* eingeben:



4. Die Änderungen mit *Ok* bestätigen:



[Zurück zur Aufgabe auf Seite 2.](#)

4.3 Lösung zu Aufgabe 3

```
public class Program
{
    // Anfang Attribute
    // Ende Attribute

    public Program ()
    {
        // Anfang Variablen
        // Ende Variablen
    }

    public void main ()
    {
        // Anfang Variablen
        Computer comp;
        String name;
        // Ende Variablen

        // Ein Objekt fuer die Variable comp erzeugen
        comp = new Computer();
        // Ausgabe von "Hallo "
        comp.print("Hallo ");
        // Name des Benutzers abfragen
        comp.read("Dein Name:");
        // Name des Benutzers ausgegeben
        comp.print(name);
        // die erste Zeile mit '!' abschliessen
        comp.print("!");
        // Zeilenvorschub einfügen
        comp.println();
        // zweite Zeile der Begrueßung ausgeben
        comp.print("Ich bin Dein freundlicher Computer.");
    }
}
```

[Zurück zur Aufgabe auf Seite 6.](#)