

Selbststudium OOP1

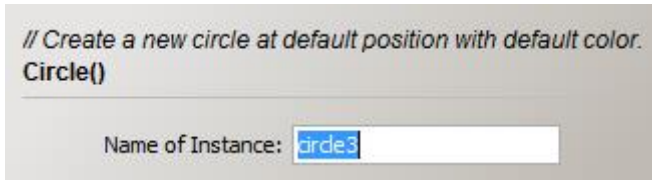
Kapitel 1.2 Creating Objects

1. zu bearbeitende Aufgabe: 1.1

done

2. Wie kann dem Objekt ein eigener Name gegeben werden?

Beim Erstellen (Instanz) von einem Objekt kann resp. muss der Name (kleingeschrieben) angegeben werden.



Kapitel 1.3 Calling methods

3. zu bearbeitende Aufgabe: 1.2

moveDown(): Das Objekt bewegt sich bei jedem Aufruf schrittweise nach Unten.

makeInvisible(): Beim ersten Aufruf wird das Objekt unsichtbar, danach bleibt es unsichtbar.

4. Formulieren Sie Ihre Beobachtung aus Aufgabe 1.2 in allgemeiner Form.

Jede Methode kann mehrmals aufgerufen werden, jedoch muss sich nicht zwingend immer etwas ändern dadurch.

Kapitel 1.4 Parameters

5. zu bearbeitende Aufgabe: 1.3

Der Parameter muss -70 sein: moveHorizontal(-70)

6. Wozu verwendet man Parameter?

Um der aufzurufenden Methode einen oder mehrere Werte mitzugeben. Z.B. die Farbe, Grösse, Anz., etc.

7. Von welchem Datentyp muss ein Parameter sein? Gibt es Einschränkungen?

Je nach Methode kann dieser Variieren. Farbe: String, die restlichen in diesem Beispiel int (Integer).

8. Soll man in Java Source-Code kommentieren? (mit Begründung)

Ja, der Source-Code sollte kommentiert werden. Z.B. jeweils oberhalb der Funktion was für eine Funktion es ist, was sie bewirkt, welche Parameter (Typen, etc) usw.

Der Source-Code wird evt. von mehreren User verwendet oder er muss später angepasst werden. So ist es viel leichter diesen „wieder“ zu verstehen. Sowie ist dann auch der Code und die Dokumentation „zusammen“ und die Doku muss nicht zuerst gesucht werden.

Kapitel 1.5 Data types

9. zu bearbeitende Aufgaben: 1.4 bis 1.6

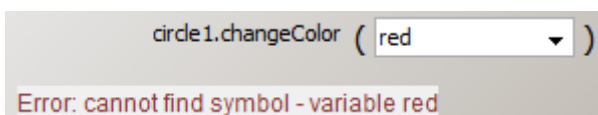
1.4:

done

1.5:

Bei einer unbekannten Farbe wird schwarz verwendet.

1.6:



10. Wie erklären Sie sich das Verhalten des Programms in Aufgabe 1.6

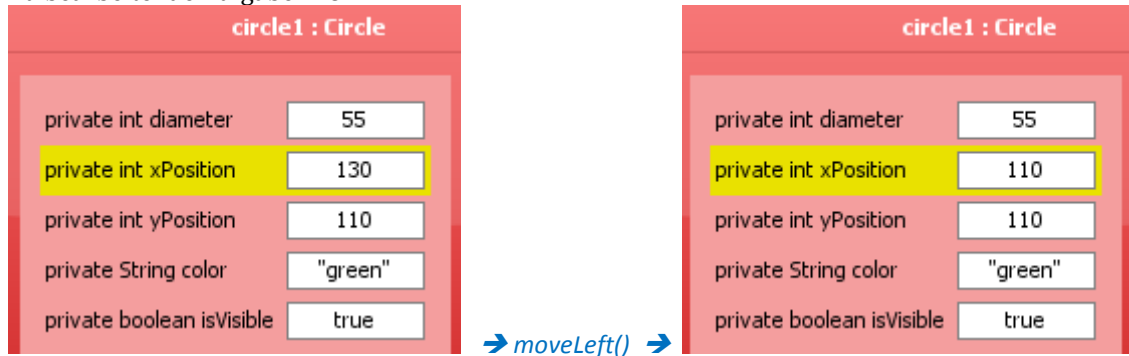
Für alle unbekannt Parameter (Strings) wird der Default-Wert, in diesem Fall schwarz, verwendet. Der String muss einem bestimmten Wert zugewiesen werden welcher die Farbe darstellt. Ist der String nicht definiert wird in diesen Fällen Schwarz verwendet.

Kapitel 1.6 Multiple instances

11. zu bearbeitende Aufgabe: 1.7
done

Kapitel 1.7 State

12. zu bearbeitende Aufgabe: 1.8



Kapitel 1.8 What is in an object?

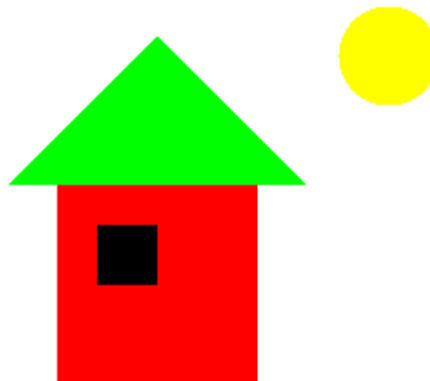
13. zu bearbeitende Aufgabe: 1.9

Es gibt verschiedene Möglichkeit, hier eine davon:

```

square1 = new Square()
square2 = new Square()
triangle1 = new Triangle()
circle1 = new Circle()
square1.makeVisible()
square1.changeSize(100)
square1.moveVertical(100)
square2.makeVisible()
square2.changeColor("black")
square2.moveHorizontal(20)
square2.moveVertical(120)
triangle1.makeVisible()
triangle1.changeSize(75,150)
triangle1.moveHorizontal(60)
triangle1.moveVertical(60)
circle1.makeVisible()
circle1.changeColor("yellow")
circle1.changeSize(50)
circle1.moveHorizontal(180)

```



14. Was ist in einem Objekt enthalten?

Ein Objekt kann mehrere Methoden (Funktionen) haben, es besitzt zudem einen Zustand. (Diverse Variablen)

15. Wie übersetzt man „state“ auf Deutsch? Was bedeutet dieser Begriff?

Zustand, dies beschreibt ganz genau den Zustand von diesem Objekt. Von einer Klasse könnten mehrere Instanzen erstellt werden, es müssen jedoch nicht zwingend alle Objekte von einer Klasse den gleichen Zustand haben. Ein Objekt hat z.B. eine bestimmte Grösse, Farbe, Position, ist sichtbar, etc.

Kapitel 1.9 Java Code

16. zu bearbeitende Aufgabe: 1.10

done

17. zu bearbeitende Aufgaben: 1.11 und 1.12

done

Kapitel 1.10 Object interaction

18. Was ist der Unterschied zwischen einer Klasse und einem Objekt?

Die Klasse ist der „Bauplan“ (Definition) aus ihr werden Objekte instanziiert.

19. zu bearbeitende Aufgaben: 1.13 und 1.14



Kapitel 1.11 Source Code

20. zu bearbeitende Aufgabe: 1.15

done

21. Überlegen Sie sich einen Lösungsweg für die Aufgaben 1.18 und 1.19.

1.18

```
public void draw()
{
    [...]
    sun.slowMoveVertical(500);
}
```

1.19

```
public void sunset()
{
    sun.slowMoveVertical(500);
}
```

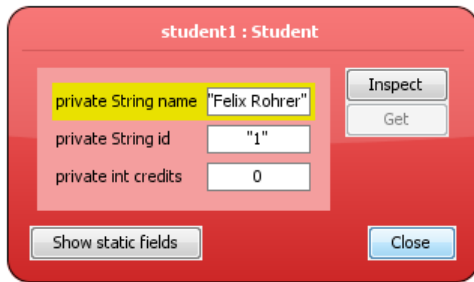
22. * Falls Sie genügend Zeit haben, lösen Sie die Aufgaben 1.18 und 1.19.

siehe Aufgabe 21

Kapitel 1.12 und 1.13 Return values

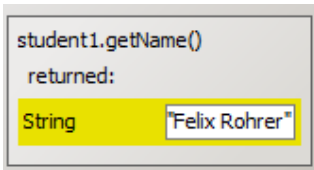
23. zu bearbeitende Aufgaben: 1.21 und 1.22

1.21:



1.22:

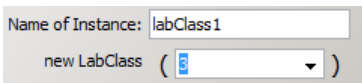
Es wird der jeweilige String (welcher beim Instanzieren als Parameter fullName übergeben wurde) zurückgegeben:



Kapitel 1.14 Objects as parameters

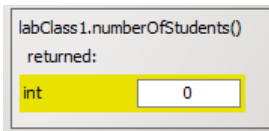
24. zu bearbeitende Aufgaben: 1.23 bis 1.28

1.23:



1.24:

Es wird die Anzahl der eingetragenen Studenten zurückgegeben. Aktuell 0



1.25:



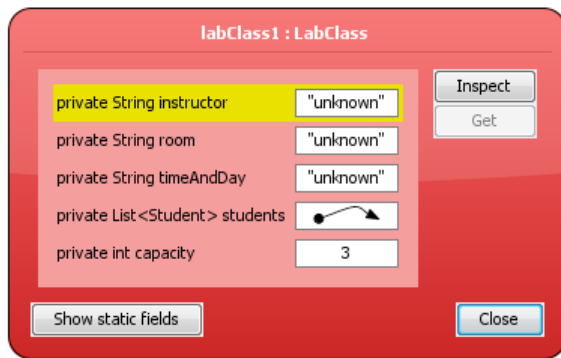
1.26:

```
Lab class unknown
Instructor: unknown   room: unknown
Class list:
Rohrer Felix (1)
Estermann Michael (2)
Bontekoe Christian (3)
Number of students: 3
```

1.27:

```
Lab class unknown
Instructor: unknown   room: unknown
Class list:
Snow white (100234)
Lisa Simpson (122044)
Charlie Brown (12003P)
Number of students: 3
```

1.28:



25. * Falls Sie genügend Zeit haben, bearbeiten Sie die Aufgabe 1.29

```
Lab class 08:30
Instructor: Diethelm   room: c211
Class list:
Rohrer Felix (1)
Estermann Michael (2)
Bontekoe Christian (3)
Number of students: 3
```

26. Wozu kann der "Resultatwert" einer Funktion verwendet werden?

Rückgabewert einer Funktion, z.B. als Resultat einer Berechnung, Inhalt einer Variablen, etc.

27. Wie heisst der "Resultatwert" einer Funktion, wenn er nicht verwendet wird?

void

Kapitel 1.15 Summary

28. zu bearbeitende Aufgaben: 1.31 bis 1.34 sowie 1.36

Bereiten Sie diese Aufgaben soweit vor, dass Sie sie in der Lernteamsitzung lösen können.

1.31:

```
0      byte, short, int, long
„hello“ string
101   byte, short, int, long
-1    byte, short, int, long
true  boolean
„33“  string
3.1415 float, double
```

1.32:

```
public class Circle
{
    private String name;
    [...]

    public void setName(String newName)
    {
        name = newName;
    }
}
```

1.33:

```
/**
 * Send to somewhere the value newMSG (String).
 */
public void send(String newMSG)
{
    // do something with newMSG
}
```

1.34:

```
/**  
 * Durchschnitt von zwei Parameter berechnen  
 */  
public int average(int para1, int para2)  
{  
    return (para1 + para2) / 2;  
}
```

1.36:

Nein

29. * Falls Sie genügend Zeit haben, bearbeiten Sie die Aufgabe 1.30 und die Aufgabe 1.35
siehe Aufgabe 28