



University  
of Basel

# Programmieren I

**Woche 5 – Objektorientierung – Java AWT/Swing**

Marcel Lüthi, Departement Mathematik und Informatik, Universität Basel

# Heutiges Programm

Zeit	Inhalt
09:15 – 10:30	Übungsbesprechung
10:30 – 10:45	Kaffeepause
10:45 – 11:00	Administratives / Prüfung
11:00 – 12:00	Fallstudie TurtleGym (II)
12:00 – 13:15	Mittagspause
13:15 – 14:15	Theorie: Vererbung, Interfaces, Abstrakte Klassen
14:15 – 14:45	Kaffeepause
14:45 – 15:30	Theorie: AWT / Swing
15:30 – 17:00	Übungen

---

# Wo stehen wir?

Wir kennen:

- Anweisungen
- Kontrollstrukturen (if, while, for, ...)
- Methoden/Funktionen
- Primitive Datentypen
- Arrays
- Klassen und Objekte



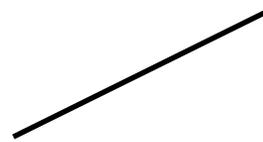
Können Sprache um eigene  
Worte/Konzepte erweitern

Was fehlt:

- Erfahrung, Beispiele
- Sprachkonzepte um Objekthierarchien zu bilden

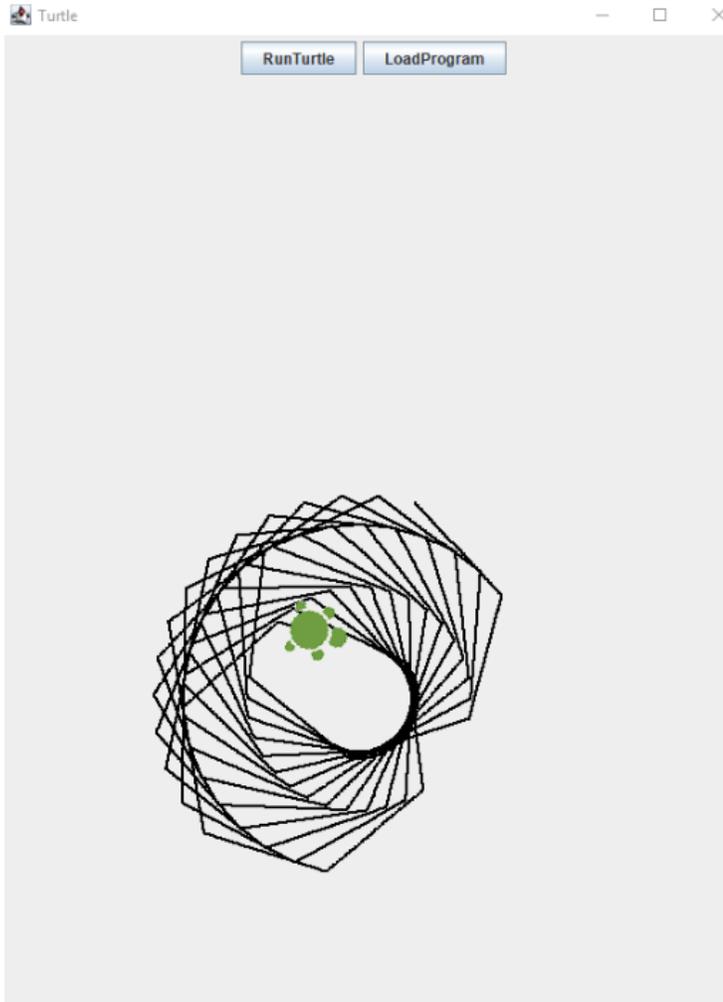
Neue Konzepte:

- Vererbung, Interfaces und Abstrakte Klassen



Erlaubt Konzepte in Hierarchien  
anzuordnen

# Fallstudie: TurtleGym



In 4 Schritten zur eigenen Turtle-Implementation

1. Implementation der Sprache (Letzte Woche)
2. **Implementation der Turtle Logik (Heute)**
3. Erstellen einer Grafischen Oberfläche (4. April)
4. Laden von Programmen (4. April)

Anwendungsbeispiel von Objektorientierung

# Prüfung



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

---

# Prüfung: Variante 1 – Stay-home Exam (Computer)

Prüfung besteht aus 2 Teilen.

1. Basics: Implementation kleiner Programme (ca. 90 Minuten)
  - Umgebung: Jupyter Notebooks
  - Schwierigkeitsgrad: Analog den Miniübungen
2. Implementation eines grösseren Programms (ca. 90 Minuten)
  - Umgebung: Java Entwicklungsumgebung mit automatisierten Tests
  - Schwierigkeitsgrad: Ähnlich wie Histogrammaufgabe

Die Note 4 kann durch perfektes Lösen des ersten Teils erreicht werden.

---

# Prüfung: Variante 2 - Papier

- (Multiple-choice) Fragen zur Theorie
- Kleine Programme schreiben
- Fehlersuche

Zeit: Ca 90 min.

*Entscheidung durch Programmkomitee noch ausstehend*

---

# Beispielfragen aus Vorjahren

## Aufgabe 1 - Multiple Choice [9 Punkte]

Richtig angekreuzt: +1 Punkt

Falsch angekreuzt: -1 Punkt

Beide angekreuzt: ±0 Punkte

Keins angekreuzt: ±0 Punkte

`1+1==3` ist ein gültiger Java Ausdruck und liefert `false`.

Ja  Nein

Jede Java Klasse muss genau einen Konstruktor haben.

Ja  Nein

Folgende Ausdrücke sind äquivalent

```
for (; x<30; ++x) { x *= 2; }
```

und

```
while(x<30) { x *= 2; ++x; }
```

Ja  Nein

Jede Methode hat höchstens eine `return` Anweisung

Ja  Nein

— —

---

# Beispielfragen aus Vorjahren

3

## Aufgabe 2 - Datentypen und Konversion [5 Punkte]

Gegeben seien folgende Variablendeklarationen:

```
byte  b = 1;
short s = 2;
int   i = 3;
long  lg = 4;
double f = 4.5;
double d = 5.5;
char  c = 'a';
```

Geben Sie für jeden der unten stehenden Ausdrücke an, welchen Wert sie liefern, und von welchem Typ sie sind (z.B. `2*d` Wert: 11; Typ double).

`s+b`                      Wert: .....    Typ: .....

`(2.0*i) % lg`            Wert: .....    Typ: .....

`(lg/2) == (i/2+1)`      Wert: .....    Typ: .....

`(char)(c+s)`            Wert: .....    Typ: .....

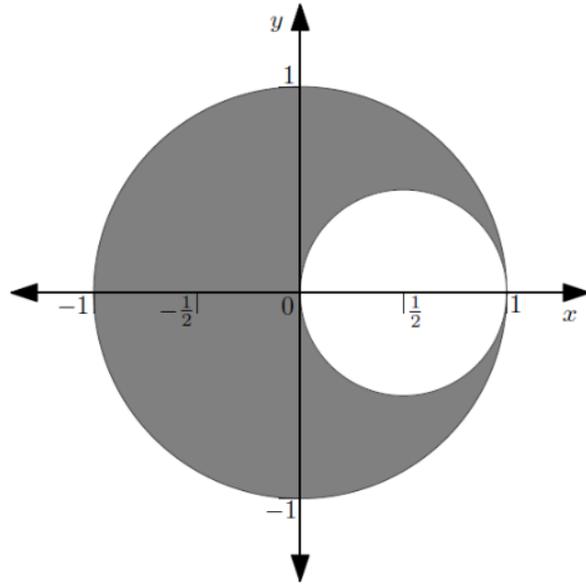
`f+d+b`                    Wert: .....    Typ: .....

---

# Beispielfragen aus Vorjahren

## Aufgabe 3 - Bedingungen [5 Punkte]

Welche Bedingungen müssen für  $x$  und  $y$  gelten, damit der Punkt  $(x, y)$  innerhalb des schraffierten Bereichs liegt? Nehmen Sie an dass  $x$  und  $y$  Fließkommazahlen sind, und drücken Sie die Bedingung in Java Syntax aus, indem Sie die untenstehende Methode ergänzen. (Hinweis: Überlegen Sie zuerst, welche Bedingungen die Punkte innerhalb des Einheitskreises erfüllen.)



Ergänzen Sie diese Funktion, so dass für die innerhalb der schraffierten Fläche liegenden Punkte `True` zurückgegeben wird.

```
static boolean isInside(double x, double y) {
```

```
}
```

---

# Beispielfragen aus Vorjahren

## Aufgabe 9 - Fehlersuche [7 Punkte]

Der folgende Code enthält 7 Fehler. Beschreiben Sie jeden der Fehler kurz, und geben Sie eine korrigierte Version des Codes an. Machen Sie ihre Anmerkungen direkt im Sourcecode.

```
/** The container interface allows insertion and extraction of elements. */  
public interface Container {  
    // Insert a new object into the container.  
    void put(Object o) {  
        /* ... */  
    }  
    // Take an object out of the container, returning the object and  
    // removing it from the container.  
    // This function returns null if no more elements can be returned.  
    Object take();  
}
```

---

# Beispielfragen aus Vorjahren

## Aufgabe 10 - Java [12 Punkte]

Schauen Sie sich das untenstehende Programm an, und beantworten Sie die Fragen am Ende dieser Aufgabe.

```
[ 1] import java.util.Vector;
```

```
[ 2] class NotEnoughMoneyException extends
```

Was bedeutet private?

```
[ 3] class Account {
```

```
[ 4]     private String title;
```

```
[ 5]     private double credit;
```

```
[ 6]     private double value;
```

```
[ 7]     Account(String title) { this.title = title; credit = 0; value = 0; };
```

```
[ 8]     public double getCredit() { return credit; }
```

```
[ 9]     public double getValue() { return value; }
```

```
[10]     public String toString() { return title + ": " +  
                                     value + " (Limit: " + credit + ")"; }
```

Was bedeutet das + hier?

```
[11]     public Account put(double amount) {
```

```
[12]         value = value + amount;
```

```
[13]         return this;
```

```
}
```