



TECHNISCHE HOCHSCHULE MITTELHESSEN

THM

CAMPUS  
GIESSEN

MNI

Mathematik, Naturwissenschaften  
und Informatik

# Brückenkurs Programmieren

Extra: Ausblick auf Java und C

Jakob Czekansky, M.Sc.

Technische Hochschule Mittelhessen

06. September 2024



# Rückblick - Tag 1: Variablen und Verzweigungen

## Variablen

```
int i = 10;  
float f = 3.5;  
boolean b = true;  
char c = 'x';  
String blabla = "Blabla";
```

## Logische Operatoren

>   >=   ==   !=  
&&   ||   !

## If-Abfrage

```
if (<Bedingung>) {  
    <Anweisungsblock1>  
} else {  
    <Anweisungsblock2>  
}
```

## Arithmetische Operatoren

+   -   \*   /   %

## Beispiel: For-Schleife

```
for (int i=0;i<end;i++) {  
    ellipse(i*10,20,10,10);  
}
```

## Syntax: While-Schleife

```
while(<Bedingung>) {  
    <Anweisungsblock>  
}
```

# Rückblick - Tag 2: Animationen und Arrays

## setup und draw

```
void setup() {  
    //einmal am Anfang  
}  
void draw() {  
    //alle 15 ms  
}
```

## Mausposition

```
void draw() {  
    ellipse(mouseX,20,10,10);  
}
```

## Arrays

```
int[] numbers = new int[10];  
numbers[0] = 20;  
numbers[3+2] = numbers[0];  
int val = numbers[5];
```

## Funktionsdefinition

```
int add(int a, int b) {  
    return a + b;  
}
```

## Funktionsaufruf

```
int x = add(10,3);
```

# Rückblick - Tag 3: Events

## keyPressed

```
int counter = 0;
void keyPressed() {
  if(key == 'a') {
    counter++;
  } else if (key == 'x') {
    exit(0);
  }
}
```

## mousePressed

```
void mousePressed() {
  if(mouseButton == LEFT){
    background(255);
  }
  else{
    background(0);
  }
}
```

## Klasse als eigener Datentyp

```
class Ball {  
    float x, y;  
    color c;  
    Ball(float x, float y, color c) {  
        this.x = x;  
        this.y = y;  
        this.c = c;  
    }  
}
```

## Objekte und Methodenaufruf

```
Ball b = new Ball(100,200);  
b.display();
```

## Klasse mit Methoden

```
class Ball {  
    float x, y;  
    color c;  
    Ball(float x, float y, color c) {  
        this.x = x;  
        this.y = y;  
        this.c = c;  
    }  
    void display() {  
        fill(this.c);  
        ellipse(x,y,50,50);  
    }  
    void moveX(int x){  
        this.x = this.x + x;  
    }  
}
```

Rückblick

Java

C



Rückblick

Java

C

## Processing

```
float x,y;
void setup() {
  size(400,300);
  x = width/2.0;
  y = height/2.0;
  background(0);
  noStroke();
}
void draw() {
  x += (random(1)-0.5);
  y += (random(1)-0.5);
  ellipse(x,y,5,5);
}
```

## Java

```
import ...

public class PdeJava extends PApplet {

    float x,y;
    public void setup() {
        size(400,300);
        x = width/2.0f;
        y = height/2.0f;
        background(0);
        noStroke();
    }
    public void draw() {
        x += (random(1)-0.5f);
        y += (random(1)-0.5f);
        ellipse(x,y,5,5);
    }
    static public void main(String[] passedArgs) {
        String[] appletArgs = new String[] { "pde_java" };
        if (passedArgs != null) {
            PApplet.main(concat(appletArgs, passedArgs));
        } else {
            PApplet.main(appletArgs);
        }
    }
}
```

# Unterschiede zwischen Processing und Java

Processing-Code wird **vor der eigentlichen Ausführung** in Java-Code übersetzt .

Was ist an Java anders?

- ▶ Code muss in einer Klasse stehen
- ▶ Einstiegspunkt `main` statt `setup` und `draw`
- ▶ Sichtbarkeiten (`public` / `private`)
- ▶ Schlüsselwort `static` für Funktionen
- ▶ Organisation von Klassen über Pakete und Importe
- ▶ Processing-Befehle durch Vererbung von Klasse `PApplet`
- ▶ Aufruf von Compiler und virtueller Maschine statt Play-Button
- ▶ Eclipse, Netbeans oder IntelliJ als IDE

Schreiben Sie eine Funktion zur Entfernung von Duplikaten in einem Array von ganzen Zahlen.

Schreiben Sie eine Funktion zur Entfernung von Duplikaten in einem Array von ganzen Zahlen.

```
import java.util.Arrays;
```

```
public class NoDuplicates {  
    public int[] noDuplicates(int[] input) {  
        int[] result = new int[input.length];  
        int n = 0;  
        for(int i = 0; i < input.length; i++) {  
            int x = input[i];  
            if (!contains(result,n,x)) {  
                result[n++] = x;  
            }  
        }  
        //trim result  
        int[] trimmed = new int[n];  
        System.arraycopy(result,0,trimmed,0,n);  
        return trimmed;  
    }  
}
```

```
public boolean contains(int[] ar, int n, int x) {  
    boolean found = false;  
    for(int j = 0; j < n; j++) {  
        if(ar[j] == x) found = true;  
    }  
    return found;  
}  
public static void main(String[] args) {  
    NoDuplicates nd = new NoDuplicates();  
    int[] ar = new int[]{2, 1, 2, 5, 5, 5, 1, 3};  
    int[] ar2 = nd.noDuplicates(ar);  
    System.out.println(Arrays.toString(ar2));  
}
```

Rückblick

Java

C

# Unterschiede zwischen Java und C

---

Die Syntax von Java ist größtenteils von der Syntax der Programmiersprache C abgeleitet.

Was ist an C anders?

- ▶ keine Klassen (prozedural)
- ▶ explizite Pointer-Arithmetik (Referenzierung, Dereferenzierung)
- ▶ manuelle Speicherverwaltung
- ▶ Definition vor Verwendung
- ▶ kompiliert direkt zu Maschinencode
- ▶ plattformabhängig
- ▶ Teilung in header- und source-Files (\*.h bzw. \*.c)



# C-Programm aus dem ersten Semester

---

Schreiben Sie eine Funktion zur Entfernung von Duplikaten in einem Array von ganzen Zahlen.

# C-Programm aus dem ersten Semester

Schreiben Sie eine Funktion zur Entfernung von Duplikaten in einem Array von ganzen Zahlen.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

void printArray(int len, int *a) {
    for(int i = 0; i < len; i++) {
        printf("%d_", a[i]);
    }
    printf("\n");
}

bool contains(int n, int *ar, int value) {
    for(int i = 0; i < n; i++) {
        if (ar[i] == value) return true;
    }
    return false;
}

void noDuplicates(int in_sz, int *in, int *out_sz, int *out) {
    int n = 0;
    for(int i = 0; i < in_sz; i++) {
        if(!contains(n, out, in[i])) {
            out[n++] = in[i];
        }
    }
    *out_sz = n;
}

int main(int argc, char **argv) {
    int a[10] = {1, 1, 3, 4, 3, 3, 5, 5, 1, 10};
    int *b = (int*) malloc(sizeof(int) * 10);
    int b_sz = 0;
    noDuplicates(10, a, &b_sz, b);
    printArray(10, a);
    printArray(b_sz, b);
    return EXIT_SUCCESS;
}
```