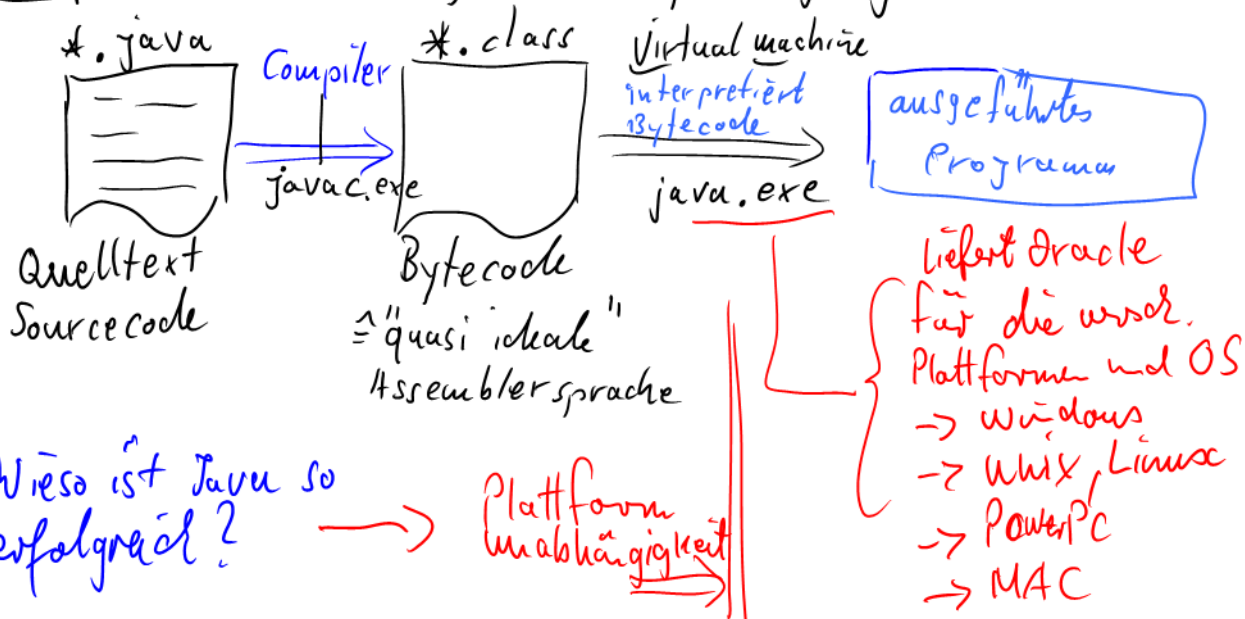


Einführung in Java



Wieso ist Java so erfolgreich?

→ Plattformunabhängigkeit

```

import java.util.Scanner;
public class schachbrett {
    public static void main(String [] a)
    {
        int start = 1;
        int ende = 9;
        for(int x=1; x<9; x++)
        {
            for(int y=start; y<ende; y++)
            {
                if(y < 10)
                {
                    System.out.print(" ");
                }
                System.out.print(y);
                System.out.print(" ");
            }
            System.out.println("");
            start++;
            ende++;
        }
    }
}
  
```

Annotations:

- `schachbrett`: **Klassennamen gross!**
- `public static void`: **Modifizierer**
- `main`: **Parameter**
- `int start = 1;`: **Einsparungspunkt beim Start**
- `for(int x=1; x<9; x++)`: **Zeilen**
- `for(int y=start; y<ende; y++)`: **Spalten**
- `System.out.print(" ");`: **Leerzeichen falls einer Stelle**
- `System.out.print(y);`: **Zahl ausgehen**
- `System.out.print(" ");`: **Leerzeichen**
- `System.out.println("");`: **Inkrement / Dekrement**
- `start++;`: **Inkrement**
- `ende++;`: **Dekrement**

(Markus)

`c--`

```
import java.util.Scanner;
public class Tannenbaum{

public static void main(String [] a)
{
    String stern = "*";
    String leer = "";
    int zahl = 0;
    Scanner s = new Scanner(System.in);
    System.out.println("wie viele Zeilen? :");
    int eingabe = Integer.parseInt(s.nextLine());
    zahl = eingabe;
    for(int i = 0; i < eingabe; i++)
    {
        for(int x = 0; x < zahl; x++)
        {
            leer = leer + " ";
        }
        System.out.println(leer + stern);
        stern = stern + "***";
        zahl--;
        leer = "";
    }
    if(eingabe != 0 && eingabe < 20)
    {
        for(int y = 0; y < stern.length()/2; y++)
        {
            System.out.print(" ");
        }
        System.out.print("I");
    }
}
```

String-Parameter
↳ Rückgabewert
der Methode zählt

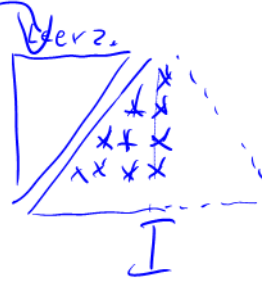
Lesbarkeit



Funktion OK

Abstand erstellen

spart sich
Schleife

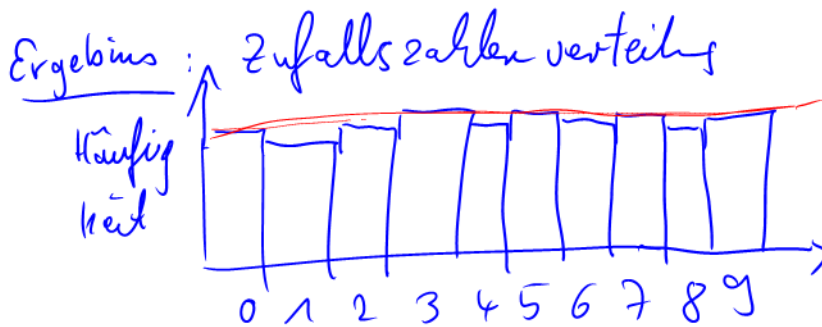


Stamm berechnen

Statistik

(Einschub #) $\uparrow \square \rightarrow \uparrow \Omega$

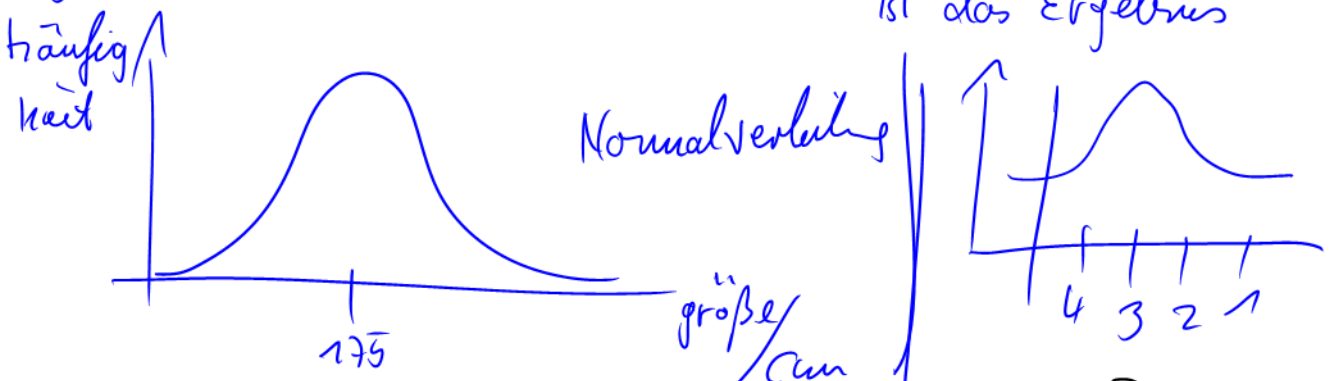
23.9.15



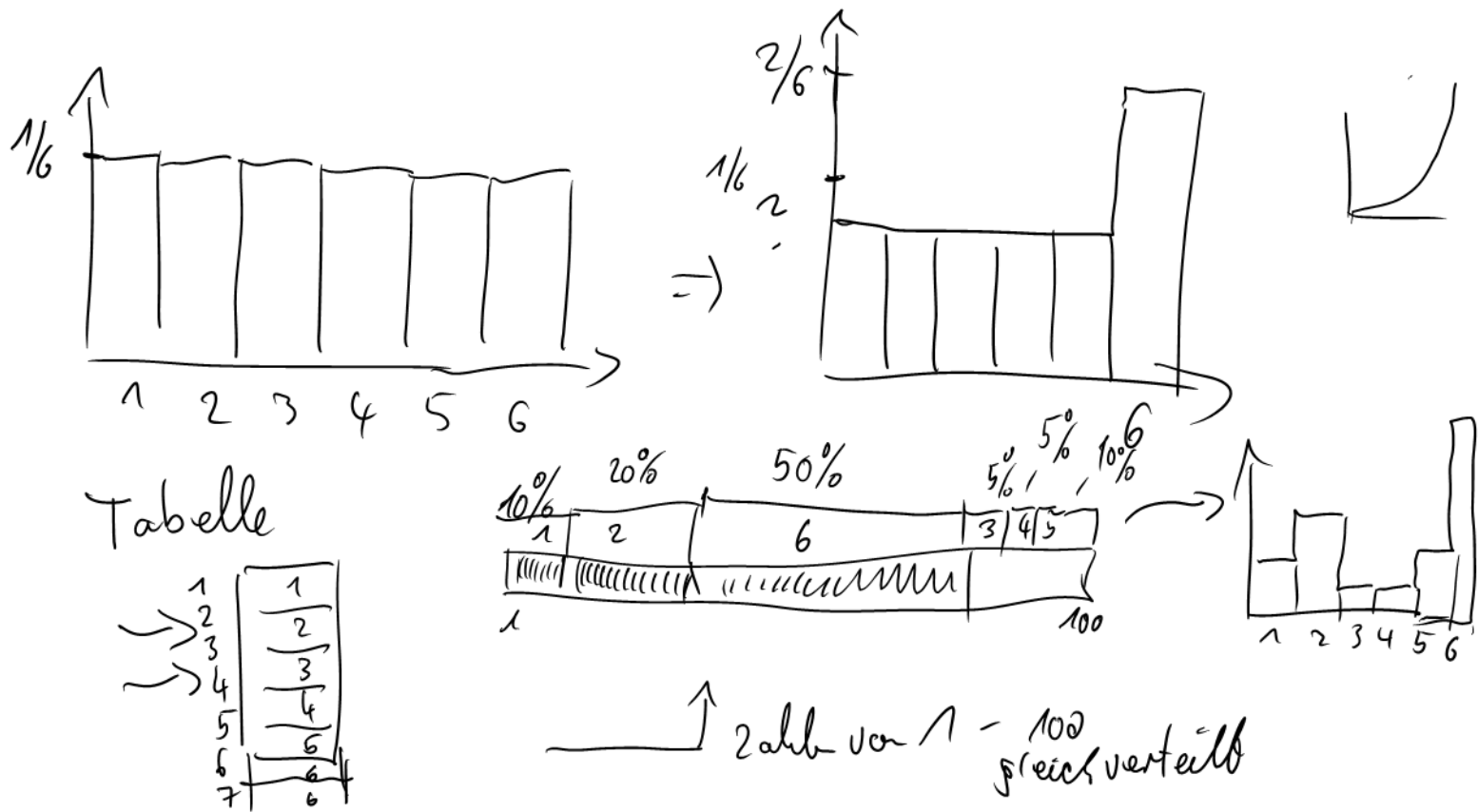
bei hoher Zahl
über Versuche

25.9.15

je höher die Zahl der Versuche \Rightarrow umso gleichverteilter
ist das Ergebnis



Wie können anders lautende Häufigkeiten erzeugt werden?



Korrektur Statistik

1) Version ohne Arrays

2) Version mit Arrays

```
import java.util.Scanner;  
import java.util.Random;
```

```
public class StatistikOLD{
```

```
    public static void main(String [] a)  
    {
```

```
        System.out.println("Statistik");
```

```
        Scanner s = new Scanner(System.in);  
        System.out.print("Geben sie Anzahl der Zahlen ein: ");  
        int eingabe = s.nextInt();  
        System.out.println();
```

```
        int zahl0=0;  
        int zahl1=0;  
        int zahl2=0;  
        int zahl3=0;  
        int zahl4=0;  
        int zahl5=0;  
        int zahl6=0;  
        int zahl7=0;  
        int zahl8=0;  
        int zahl9=0;
```

```
        int maxVal = 0;  
        int durchschnitt;
```

```
        for(int i = 0; i < eingabe; i++)  
        {
```

```
            switch(getZufall())  
            {
```

```
                case 0: zahl0++;  
                        break;
```

```
                case 1: zahl1++;  
                        break;
```

```
                case 2: zahl2++;  
                        break;
```

```
                case 3: zahl3++;  
                        break;
```

```
                case 4: zahl4++;  
                        break;
```

```
                case 5: zahl5++;  
                        break;
```

```
                case 6: zahl6++;  
                        break;
```

```
                case 7: zahl7++;  
                        break;
```

```
                case 8: zahl8++;  
                        break;
```

```
                case 9: zahl9++;  
                        break;
```

```
            }
```

```
            if(maxVal < zahl9)  
                maxVal = zahl9;
```

```
            if(maxVal < zahl8)
```

```

        Stati st i kOLD. j ava
        maxVal = zahl 8;

        i f(maxVal < zahl 7)
            maxVal = zahl 7;

        i f(maxVal < zahl 6)
            maxVal = zahl 6;

        i f(maxVal < zahl 5)
            maxVal = zahl 5;

        i f(maxVal < zahl 4)
            maxVal = zahl 4;

        i f(maxVal < zahl 3)
            maxVal = zahl 3;

        i f(maxVal < zahl 2)
            maxVal = zahl 2;

        i f(maxVal < zahl 1)
            maxVal = zahl 1;

        i f(maxVal < zahl 0)
            maxVal = zahl 0;
    }

```

```

    durchschni tt = maxVal / 50;
    i f(durchschni tt == 0)
    {
        durchschni tt = 1;
    }

```

falsche Normierung

" Treffern");

```

    System.out. pri nt("Legende: ei n X entsprich t " + (durchschni tt) +
    System.out. pri ntln());

```

```

    System.out. pri nt("0: ");
    for(i nt y = 0; y < (zahl 0/durchschni tt); y++)
    {
        System.out. pri nt("X");
    }
    System.out. pri ntln();

```

```

    System.out. pri nt("1: ");
    for(i nt y = 0; y < (zahl 1/durchschni tt); y++)
    {
        System.out. pri nt("X");
    }
    System.out. pri ntln();

```

```

    System.out. pri nt("2: ");
    for(i nt y = 0; y < (zahl 2/durchschni tt); y++)
    {
        System.out. pri nt("X");
    }
    System.out. pri ntln();

```

```

    System.out. pri nt("3: ");
    for(i nt y = 0; y < (zahl 3/durchschni tt); y++)
    {
        System.out. pri nt("X");
    }
    System.out. pri ntln();

```

```

    System.out. pri nt("4: ");
    for(i nt y = 0; y < (zahl 4/durchschni tt); y++)
    {

```

```

        StatistikoLD.java
        System.out.println("X");
    }
    System.out.println();

    System.out.println("5: ");
    for(int y = 0; y < (zahl5/durchschnitt); y++)
    {
        System.out.println("X");
    }
    System.out.println();

    System.out.println("6: ");
    for(int y = 0; y < (zahl6/durchschnitt); y++)
    {
        System.out.println("X");
    }
    System.out.println();

    System.out.println("7: ");
    for(int y = 0; y < (zahl7/durchschnitt); y++)
    {
        System.out.println("X");
    }
    System.out.println();

    System.out.println("8: ");
    for(int y = 0; y < (zahl8/durchschnitt); y++)
    {
        System.out.println("X");
    }
    System.out.println();

    System.out.println("9: ");
    for(int y = 0; y < (zahl9/durchschnitt); y++)
    {
        System.out.println("X");
    }
    System.out.println();
}
static int getZufall()
{
    Random random = new Random();
    int randomNumber = random.nextInt(10);
    return randomNumber;
}
}

```

← jedes mal neues Objekt sollte man vermeiden
 → sollte ins Memberfeld ↑

besser mit Arrays



```

import java.util.Scanner;
public class StatistikArr{
    public static void main(String [] a){
        Scanner s = new Scanner(System.in);
        System.out.print("Anzahl der Zufallszahlen: ");
        int max = 0, rndm, liste[] = new int[10], anzahl =
Integer.parseInt(s.nextLine()); //Deklaration und Eingabe
        System.out.println("");
        for(int i = 1; i <= anzahl; i++){ //Generiert die Zufallszahlen
            und zählt sie
                rndm = getZufall();
                liste[rndm]++;
        }
        if(groestezahl(liste) < 50){
            max = 50;
        }
        else{
            max = groestezahl(liste);
        }
        for(int j=0; j <=9; j++){ //Geht den index des Arrays durch(0-9)
            System.out.print(j + ": ");
            for(int z=1; z <=liste[j]/(max/50); z++){ //Schaut, dass
                es nie über 50 "X" werden und gibt sie dementsprechend aus.
                    System.out.print("X");
            }
            System.out.println("");
        }
        System.out.println("Legende: Ein X entspricht " + max/50 + "
Treffer(n)");
    }
    public static int getZufall(){ //Gibt eine Zufällige Zahl zwischen 0-9
        zurück
            int randomNum = 0 + (int)(Math.random()*10);
            return randomNum;
        }
        public static int groestezahl(int array[]) //Gibt die größte Zahl aus
        dem Array zurück
            int max = 0;
            for(int i = 0; i <= array.length - 1; i++){
                if(array[i] > max){
                    max = array[i];
                }
            }
            return max;
        }
    }
} //Oj is DMG.

```

Zahlenraten korrektur

Zahlenraten.java

Koczwar

```
import java.util.Scanner;
import java.util.Random;
public class Zahlenraten { //Ale c
    static Random random = new Random();
    static int getZufall()
    {
        int randomNum = random.nextInt(1000);
        return randomNum++;
    }
    public static void main(String [] a)
    {
        Scanner s = new Scanner(System.in);
        int eingabe, antwort, randNum, versuch=0;
        System.out.println("\nZahlenraten");
        System.out.println("Erraten Sie eine Zahl zwischen 1 und
        1000.");
        do { //Spiel schlei fe
            randNum = getZufall();
            do { //Versuchschlei fe
                versuch++;
                System.out.println("\nVersuch: " + versuch);
                System.out.print("Geben sie ei n: ");
                eingabe = s.nextInt();
                if(eingabe > randNum){
                    System.out.println("Die Zahl ist zu
                    hoch. \n");
                }
                else if(eingabe < randNum){
                    System.out.println("Die Zahl ist zu
                    niedrig. \n");
                }
            } while(eingabe != randNum && versuch < 10);
            if(eingabe == randNum){
                System.out.print("Die Zahl ist richtig! \n");
            }
            versuch=0;
            System.out.print("\nWollen Sie nochmal spielen? (Ja = 1
            / Nein = 2) \n");
            antwort = s.nextInt();
        } while(antwort == 1);
    }
}
```

prelukremut
erforderlich
oder (a+1)

0-999
1-1000

Vorgabe
wurde ungenau
umgesetzt!

Zeilen umbrechen

IDE verwenden:

USER

Workspace

Dateiverzeichnis

eclipse

• metadata
01 Einfüh
02 Arrays

ordner verwaltet
durch eclipse

← source
und binaries

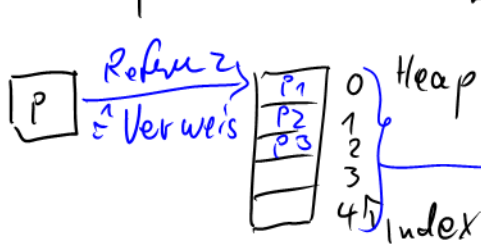
"Finger weg!"

30.9.15

Arrays

Übungen zu Arrays AB:

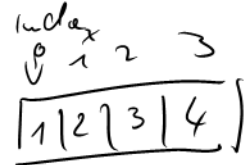
7.10.15

Informationen zu ArraysDeklaration: `char[] a;` `Point[] p;`Erstellen von Arrays: `p = new Point[5];` → 5 ElementeInitialisierung

über Index jedes Element zuweisen! (z.B. Schleife)

`p[0] = new Point();``p[4] = new Point();`Mehrdimensionale Arrays`int t[][] = new int[3][4];`

↓ initialisieren mit 2 geschachtelte Schleife

Array: schneller Initialisieren (Test prog)`int feld[] = {1, 2, 3, 4};` →`int f2[][] = {{1, 2, 3}, {2, 3, 4}};`

`int l1 = f2[1].length` ergibt jeweilige 2-Länge
`f2.length` ergibt Anzahl der Zeilen
`f2[1][2];`

Array Grenzen

geg:
array a

Länge: `int l = a.length;`

~~Seine Methode~~ → ~~length()~~
Parameter
"member"

`int b = new int[3];`

(Programmabbruch)

`b[3] = 5;`

array Out of Bound Exception⁰

Ändern von Array-Größe

`int[] c = {3, 7, 2};`

~~`c.add(5)`~~

nur new erstellen! → größer: `int c1[] = new int[4]`
⊕ Elementweise kopieren!
→ kleiner

Arrays kopieren (seit Java 6)

`java.util.Arrays` → `public static T[] copyOf(T[] org, neue Länge int l)`

sehr alt: (Java 1.0)

`System.arraycopy(Quelle Länge Ziel Länge)`

Verbesserte Schleife

`for (int element : array) { * }`

iteriert alle Elemente von array

* `System.out.println(element);` ⇒
1
2
3
4

array

1
2
3
4

Strings kopieren

String s[] = { "H", "a", "l", "l", "o" }; 5 Elemente

s Referenz



s1

Seine Referenz
int bei primitives
a = 3 (zahlen)

b = 9

a = 4

b => 3

Anwendung:

die einzelne String
sind nicht als Text
im Speicherplatz
dort stehen auch
Referenzen.

"H" → [H]

"H" → * Literal
a = "H"

aber: new String("H");

"H" → neuer Speicher [H]

String s1 = s;

↑
kopiere hier
nur Referenz

s[4] = ";";

s1[4] ist jetzt auch (!)

≡ gleicher Speicherplatz

shallow copy

geg: String s mit 5 Elementen

String [] s1 = Arrays.copyOf(s, 5);



s[4] = ";";

⇓
Halli



s1[4] → o

Hallo

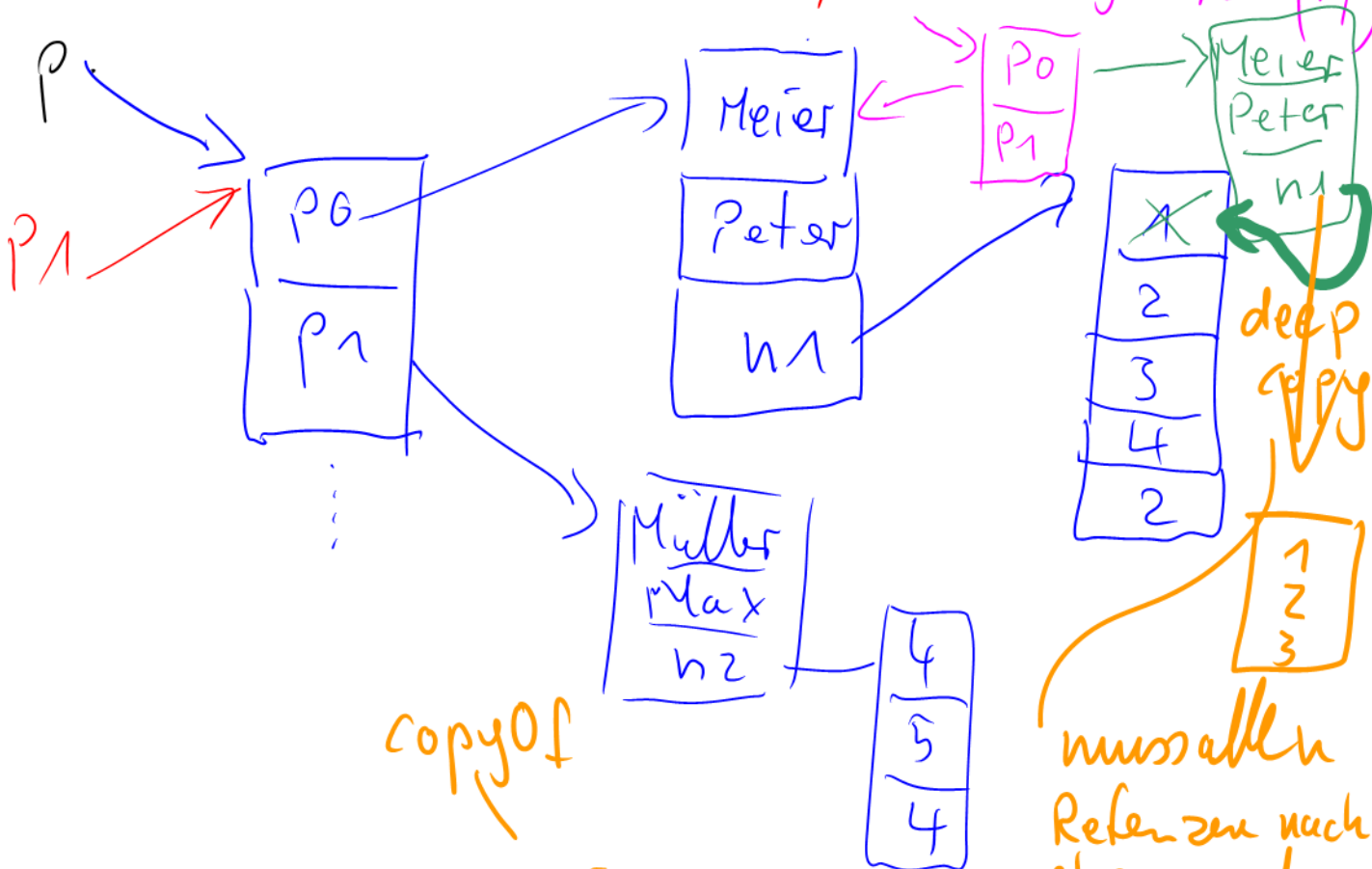
deep Copy

```
public class Person {  
    String name;  
    String Vorname;  
    Note [] n;  
    ...  
}
```

```
class Note {  
    int note;  
    double schnitt;  
    ...  
}
```

Person p[] = { new Person(... new Note...) }

$p_1 = p;$ $p_2 = \text{Arrays.copyOf}(p, 2)$



copyOf

müssen
Referenzen nach
gehen und

deren Objekte duplizieren

deepcopy: es müssen die Objekte, die

alle ∇
hinter den kopierten Verweisen/Referenzen
stehen auch dupliziert werden.

Gefahr \Rightarrow unendlich

Ablaufsteuerung handy ✓ Modifizier ^{privat} _{public}

Ablaufsteuerung handy mit Keyboard ✓

L14 für Mi 7.10, Reihenfolge ∇

b.H. am nächsten Anwesende weitergeben!

in Klassen Diagramm: `public int tuWas(String s) { ... }`
`tuWas(String s): int` → UML-Klassendiagramm

05 Aufgabe: Übungen zu Polymorphie in Java

Projekt: Polymorphie

Gegeben sei in Hauptprogramm PolyHaupt.

```
public class PolyHaupt {
```

```
    public static void main(String[] args) {
```

```
        PolyPerson pp0 = new PolyPerson();
        PolyPerson pp1 = new PolyVater();
        PolyPerson pp2 = new PolyKind();
        PolyPerson pp3 = new PolyEnkel();
```

```
        pp0.sagMeinung();
        pp1.sagMeinung();
        pp2.sagMeinung();
        pp3.sagMeinung();
```

```
        System.out.println("Ende der Vorstellung,
        jetzt experimentieren wir");
```

```
        pp0 = pp2;
```

```
        PolyEnkel ppx;
```

```
        ppx = (PolyEnkel) pp3 //Zuseisung pp3
        ppx.sagMeinung();
```

```
    }
```

Aufgabe1: Codiere folgende Klassen!

Die jeweiligen Methoden `sagMeinung(): void` geben auf der Konsole folgende Texte aus:

- PolyPerson: „Eine Person ist neutral“
- PolyVater: „Ich bin ein Vater“
- PolyKind: „Ich bin ein Kind“
- PolyEnkel: „Ich bin ein Enkel“

Beobachtung:

Konsolenausgabe:

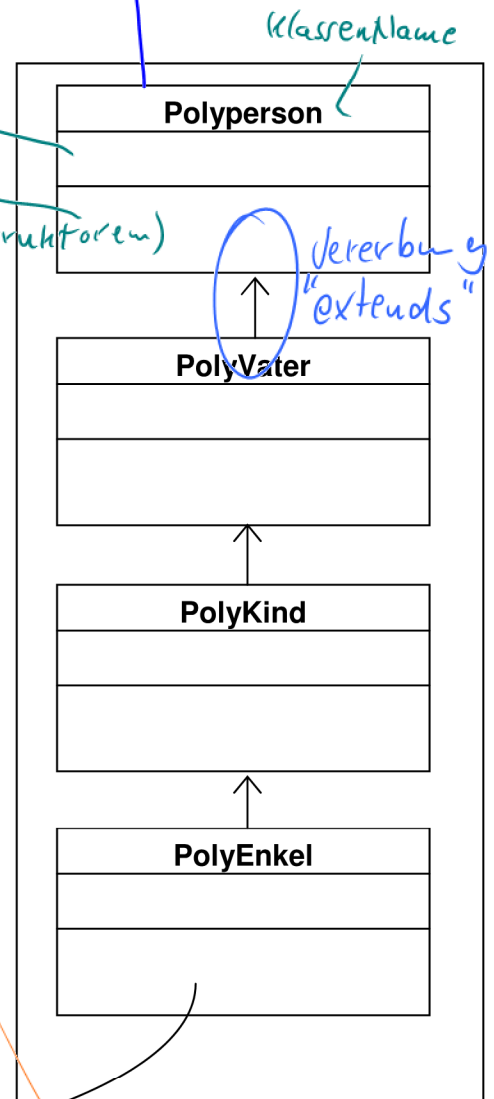
```
Eine Person ist neutral
Ich bin ein Vater
Ich bin ein Kind
Ich bin ein Enkel
Ende der Vorstellung, jetzt experimentieren
wir
Ich bin ein Kind
Ich bin ein Enkel
```

Erkläre den gezeigten Codeabschnitt und überlege die damit verbundene Problematik

```
PolyEnkel ppx;
```

```
ppx = (PolyEnkel) pp3 //Zuseisung pp3
```

```
ppx.sagMeinung();
```



pp3 ist PolyPerson und weiß nicht dass es "Enkel" ist

Abhilfe:
 (PolyEnkel)
 Typecast

erzwingt
 Typumwandlung

Aufgabe 2:

Erweitern Sie die PolyHaupt mit einer Methode, die die Nutzung eines polymorphen Argumentes demonstrieren kann.

Zeigen Sie auch die Anwendung des instanceof Operators in diesem Beispiel.

Aufgabe 3:

Erstellen Sie eine Klasse Statistik (als Hauptprogramm) die eine vararg-Methode enthält zum Errechnen des Durchschnitts von ganzen Zahlen.

Aufgabe 4: freiwillige Zusatzaufgabe

Kopieren sie die Klasse MeinDatum aus dem Vorgängerprojekt. Überschreiben Sie die geerbte equals(), die hash() und die toString() Methode

(Hinweis, der Rückgabewert int der hash() Methode sollte in geeigneter Weise gleiche Datums mit einer gleichen int Zahl signalisieren)

zu 1) Polymorphie ermöglicht Objekte
aus einer Vererbungshierarchie
auf einer Variable der Oberklasse
zu verwenden. Die Methoden
funktionen bleiben erhalten

zu 2) Polymorphes Argument am Beispiel `public void fuhras (PolyPerson p)`
Man kann abgeleitete Elemente
übergeben: p als PolyKund oder
PolyEnkel
`{ p.sagMeinung();`
Voraussetzung: Vererbungshierarchie
121025-PolymorphieAB.docx
}

instanceOf : Test ob Objekt in einer Vererbungs-
 hierarchie stehen!
 p instanceof Polyperson / true / false ✓

zu AB var Arg()

public void fuWas (String... s) { }

1) Aufruf:

fuWas("1");

⇒

s[] → "1" Länge [1]

2) Aufruf:

fuWas("1", "2", "3");

⇒ s[] der Länge [3]

3) Aufruf:

fuWas()

⇒

[1|2|3]

s[] → Länge 0

105 Polymorphie

9.10.15

Wrapper Klassen

Primitive Variablen in Java:

4 Byte: int, 8 Byte: double, 1 Byte: boolean, 2 Byte: char, 4 Byte: float
 2 Byte: short, 1 Byte: byte, 8 Byte: long

?

~~String~~

String s = "X"

s → [X]

int a = 7;

a [7]

Integer
Double
Boolean
Character
Float
Short

Byte
Long

Integer i = new Integer("2");

i [2]

`int i = 5;`
`Integer iw = Integer.valueOf(i);`

Rückgabe wert ist Objekt

Boxing

`int i2 = iw.intValue();`

Rückgabe wert primitiv

Unboxing

`Character aw = 'a';` ✓

Autoboxing

`Integer iw2 = 3;` ✓

(seit Java 5)

`int z = iw2;` ✓

Auto unboxing

Übung zu Wrapper Klassen:

System.currentTimeMillis() → Long

16.10.15

bei Schleife für Zeitmessung

`Integer iw = i;`
`Integer iw_copy = iw;`

(Autoboxing)

↳ Compiler optimiert ggf.
soll vermieden werden

Ergebnisse		zeit /ms		
Anzahl	printire kopi	Boxing	Autoboxing	
1.000.000	15	16	15	surface 1
1.000.000.000	2	6	1100	Asus(i5)
1.000.000	3	4	9	
342.352.352	9	378	377	
100.000.000.	44	495	345	

Aufzählungstypen

≙ Enumeration "Enum's"

public class Wota {

public final int MONTAG = 1;
 " " " DIENSTAG = 2

...
 }

Verwendung bei Farben

Color.RED

RGB Wert
 = 3 Bytes

int a = Wota.MONTAG
 int b = Wota.DIENSTAG

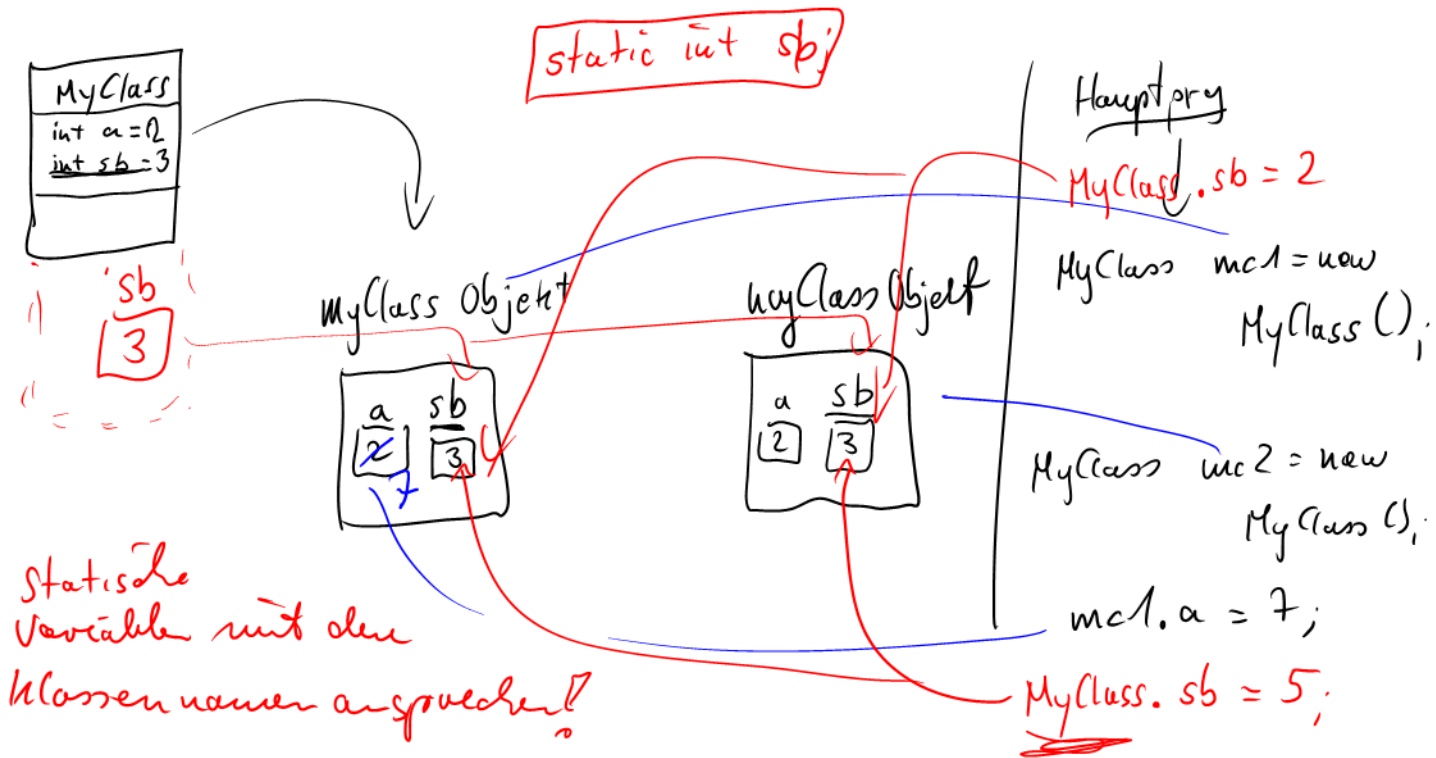
int c = a + b ;
 ↓
 3

Mit Word

↳ oder besser: enum Type

Einschub: statische Variable

16.10.15



Übung Aufgabe 2 3 Arten für Aufzählertypen ✓

HÜ ⇒ M: 21.10 • AB: erw. Klassenfunktion **A1** **objektzähler**
⇒ • Alternative für **switch(Farbe)** → **Farbname: String**

⇒ Hi ⇒ statischer Import
⇒ abstr. Klasse & Interfaces

21.10.15

✓ ⇒ zu HÜ: Statt **switch(Farbe)** ^{zur} Zuordnung der Farbnamen
kann wir ein Array: String verwenden

FarbNr ⇒ index 0 →

rot
gelb
schwarz

 → string

⇒ statischer Import

import static java.lang.Math.sqrt;

✓ ↳ sparsam verwenden
da Namensraum eingeschränkt wird
↳ sqrt(x) aufrufen, als wenn es
eigene Klassenfunktion wäre

Abstrakte Klassen

A.-Klasse legt nur fest welche Methoden in abgeleiteten Klassen ausprogrammiert werden müssen

```
abstract class A {
```

```
    public abstract void tuWas();
```

```
    public void tuNochWas() {};
```

```
}
```

← "Body" ⇒ abstract
ohne Rumpf

← tut nix ist aber implementiert

```
class B extends A {
```

```
    void tuWas() { ..... };
```

22.10.15

Interfaces

```
public class C implements I {
```

```
    public void tuWas() { ..... };
```

```
}
```

• es können mehrere Interfaces implementiert werden!

! abstr. Klassen nur 1x extends

```
interface I {
```

```
    public void tuWas();
```

```
    public static int a;
```

```
    public final static B = 3; ✓
```

```
    public void tuNochWas() { ..... }; nicht erlaubt
```

```
}
```

• Sammlung von Methoden köpfen!

• ~~ausprogrammierte Methoden~~ nicht erlaubt

• Variablen nicht erlaubt

• Konstanten erlaubt + Compiler { public
ergänzt } final