

# Mathematics Support

von Niklas Wille und Paul Fischer

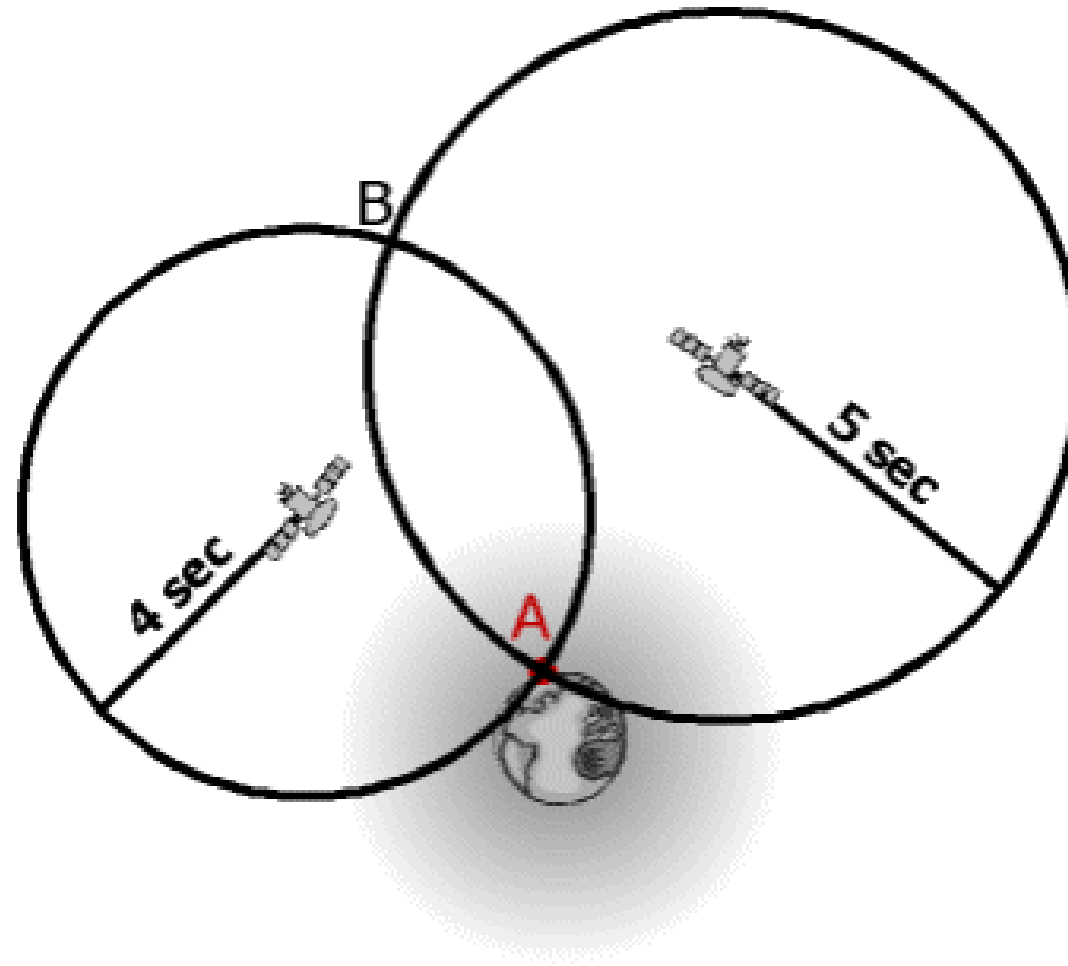
$\pi$

# Gliederung

1. Einstieg
2. Grundrechenarten
3. Runden
4. Winkelbeziehungen
5. Potenzen
6. Logarithmus
7. Zufällige Zahlen
8. Big Integer
9. Big Decimal
10. Quellen

$\pi$

# 1. Einstieg GPS



## 2. GRUNDRECHENARTEN

- Addition
- Subtraktion
- Multiplikation
- Division

*int c = (a+b)*

*int c = (a-b)*

*int c = (a\*b)*

*int= (a/b)*

### 3. RUNDEN

- Runden nach bekannten Regeln und aufrunden bei ,5:
- Runden nach bekannten Regeln und abrunden bei ,5:
- Aufrunden zur nächsten Ganzzahl:
- Abrunden zur nächsten Ganzzahl:

*Math.round(c)*

*Math rint(c)*

*Math.ceil (c)*

*Math.floor(c)*

## 4. Winkelbeziehungen

- Umwandlungen zwischen Winkeln und dem Bogenmaß:
  - *Rad = toRadians(deg)*
    - Range: 0.0 bis Math.PI
  - *Deg = toDegree(rad)*
- Winkelfunktionen:
- Sinus (rad):
  - *double sin = Math.sin(rad)*
- Cosinus (rad)
  - *double cos = Math.cos(rad)*

## 4. Winkelbeziehungen

- Tangens (rad)
  - *double tan= Math.tan(rad)*
- Umkehrfunktionen:
- Arkuskosinus:
  - *double acos= Math.acos(rad)*

## 5. Potenzen

- Basis hoch Exponent:
  - *double c = Math.pow(b, e)*
- Euler:
  - *double c = Math.exp(e)*
- Wurzeln:
  - Quadratwurzel:
    - *double c = Math.sqrt(r)*
  - Kubikwurzel:
    - *double c = Math.cbrt(r)*



## 6. Logarithmus

- Natürlicher Logarithmus:
  - *double ln= Math.log(n)*
- Logarithmus zur Basis 10:
  - *double log10= Math.log10(n)*
- Beliebiger Logarithmus mit Log-Gesetzen:
  - $\log_b r = \frac{\log_a r}{\log_a b}$

## 7. ZUFÄLLIGE ZAHLEN

- Pseudo Zufallszahlen
- Random()
  - > Util importieren vor class:
    - Auch Methode bei Klasse Mathe möglich!!!
- Einleiten des Random Prozesses:
- Zufällige Integer Zahl ohne Begrenzung:
- Zufällige Int Zahl von a bis b:
- Zufälliges Boolean:

```
import java.util.*
```

```
Random ran =new Random();
```

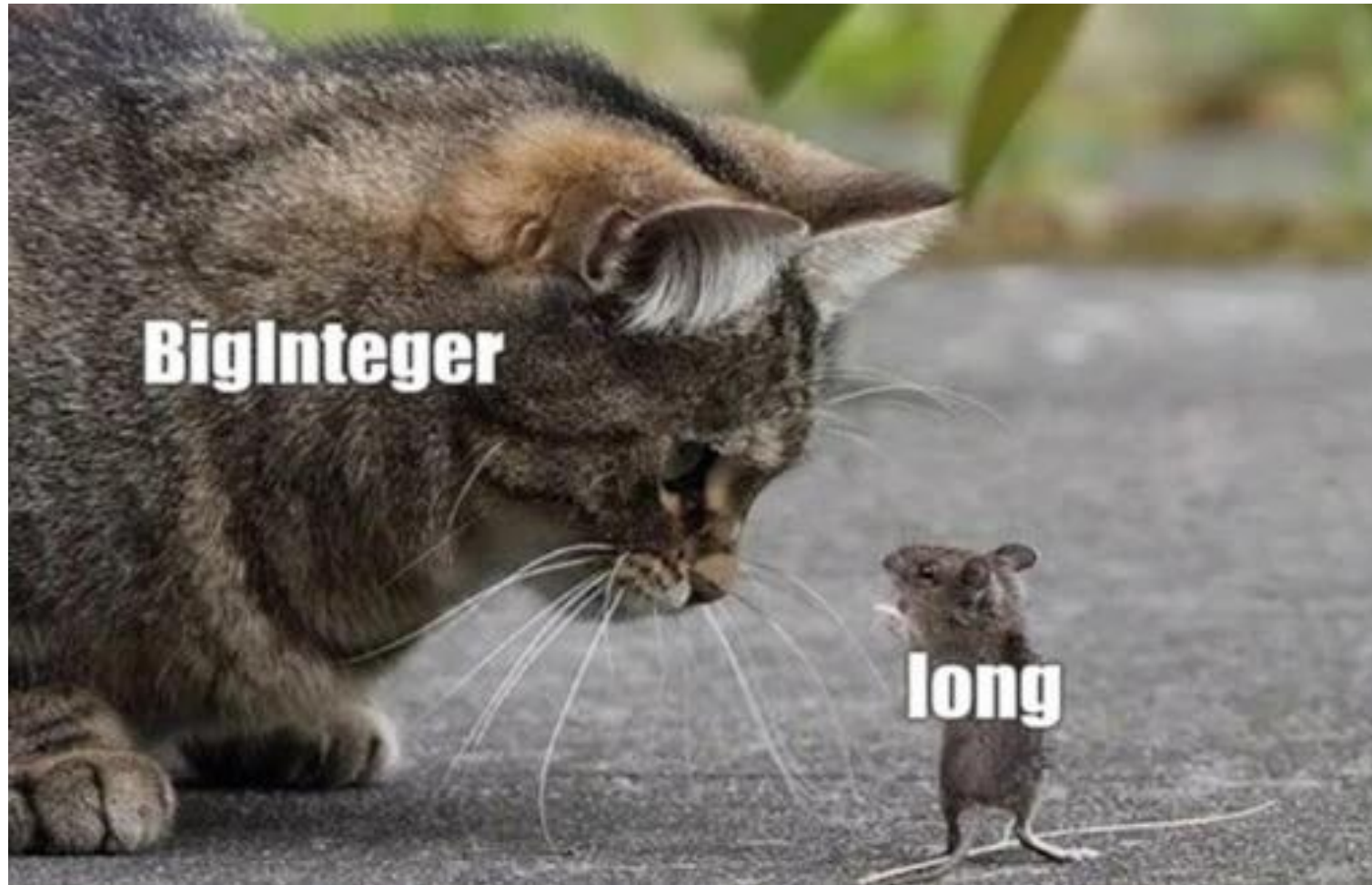
```
random.nextInt()
```

```
random.nextInt(b+1)
```

```
random.nextBoolean()
```

$\pi$

## 8. Einstieg BigInteger / BigDecimal



## 8. Big Integer

- kein primitiver Datentyp

-> Objekt

- Ganzzahlige Operationen
- Problem:
- 13! Mit Typ Int nicht berechenbar:
  - Laut Taschenrechner: 6227020800
  - Laut Int: 1932053504
  - Laut BigInt: 6227020800
- Sinn:
  - Rechnen mit sehr großen Zahlen ermöglichen und Ungenauigkeiten vermeiden

## Quelltext:

```
1 package biginteger;
2
3 import java.math.BigInteger;
4
5 public class Main {
6     public static int getFactorial(int f) {
7         int result = 1;
8         for (int i = 1; i <= f; i++) {
9             result = result * i; // Ermitteln der Fakultät mit Schleifen
10        }
11        return result;
12    }
13
14
15    public static BigInteger getFactorial1(int f) { // Berechnen der Fakultät mit BigInteger (Java Fakultät math)
16        BigInteger result = BigInteger.ONE; // Startwert 1
17        for (int i = 1; i <= f; i++)
18            result = result.multiply(BigInteger.valueOf(i));
19        return result;
20    }
21
22    public static void main(String[] args) {
23        System.out.println("Fakultät von 12 mit Int: "+getFactorial(12));
24        System.out.println("Fakultät von 12 mit BigInt: "+getFactorial1(12));
25        System.out.println("Fakultät von 13 mit Int: "+getFactorial(13));
26        System.out.println("Fakultät von 13 mit BigInt: "+getFactorial1(13));
27    }
28 }
```

## Output:

```
Fakultät von 12 mit Int: 479001600
Fakultät von 12 mit BigInt: 479001600
Fakultät von 13 mit Int: 1932053504
Fakultät von 13 mit BigInt: 6227020800
```

## 9. Big Decimal

- kein primitiver Datentyp
- > Objekt
- Bietet mehr Maschinenzahlen
- Operation mit Kommazahlen
- Häufige Fehler: - vergessen Klasse zu importieren  
- nicht als String geschrieben

Quelltext:

```
1 package bigdecimal;
2 import java.math.BigDecimal;
3 public class Main {
4
5     public static void main(String[] args) {
6         double a= 1.10;
7         double b= 2.15;
8         System.out.println("Subtraktion mit Typ double: "+(b-a));
9
10        BigDecimal aBig = new BigDecimal ("1.10");
11        BigDecimal bBig = new BigDecimal ("2.15");
12        System.out.println("Subtraktion mit BigDeciaml: "+bBig.subtract(aBig));
13
14    }
15
16 }
17
```

Output:

```
Subtraktion mit Typ double: 1.0499999999999998
Subtraktion mit BigDeciaml: 1.05
```

## 10. Quellen

- › <https://www.educative.io/edpresso/how-to-generate-random-numbers-using-random-class-in-java>
- › <http://java-zwischendurch.blogspot.com/2012/08/methoden-in-math.html>
- › <https://docs.oracle.com/javase/7/docs/api/>
- › <https://en.ppt-online.org/559289>
- › <https://www.slideserve.com/ura/chapter-4-mathematical-functions-characters-and-strings>
- › <https://codegym.cc/de/groups/posts/501-die-fakultt-in-java>
- › <https://templates.office.com/de-de/pr%C3%A4sentation-f%C3%BCr-den-mathematikunterricht-mit-pi-breitbild-tm02787947>