

Programmierereinführung

Dr. Hui Shi

WS 09/10



Java Programme

Beispiele für Java Programme

- Denition: $x + y$, where $x, y \in \mathbb{Z}$
- Ein Java-Programm

```
public class Add {  
    public static void main(String[] args) {  
        int x, y;  
        x = 10; y = 15;  
        System.out.println(x+ " + " + y + " = "  
                            + (x+y));  
    }  
}
```

- Fragen: Was ist der Unterschied von “+” in $x + y$ und $x + " + " + y$?
- Diskussion
 - Unterschiede zwischen der mathematischen Formel und dem Programm
 - Klasse
 - **main**-Methode
 - Typ
 - Deklaration
 - Initialisierung

Maximum

- $z = \max(x, y), x, y, z \in \mathbb{Z}$
- Ein Java-Programm

```
public class Max {  
    public static void main(String[] args) {  
        int x, y, z;  
        x = 10; y = 15;  
        if (x <= y) z = y;  
        else z = x;  
        System.out.println("z = " + z);  
    }  
}
```

Wurzelberechnung

- $z = \sqrt{x} \rightarrow z \times z \leq x \leq (z + 1) \times (z + 1)$,
wobei $x \in \mathcal{N}$
- Ein Lösungsansatz
 - Nehmen Sie eine beliebige Zahl x und eine **Variable** z .
 - **Initialisieren** Sie z mit 0.
 - Solange $z \times z \leq x$ ist, addieren Sie eine 1 zu z .
 - Schreiben Sie das Ergebnis auf.

- Ein Programm

```
public class Root {  
    public static void main(String[] args) {  
        int x, z;  
        x = 101; z = 0;  
        while ((z+1)*(z+1) < x)  
            z = z+1;  
        System.out.println("The root of " + x +  
            " ist " + z);  
    }  
}
```

- Ein Programm mit Parametern

```
public class Root {  
    public static void main(String[] args) {  
        int x, z;  
        x = Integer.parseInt(args[0]);  
        z = 0;  
        while ((z+1)*(z+1) < x)  
            z = z+1;  
        System.out.println("The root of " + x +  
                            " ist " + z);  
    }  
}
```

● Fragen

- Wie läuft die Berechnung ab?
- Wie oft wird die **Zuweisung** $z = z + 1$ durchgeführt?

● Diskussion

- Die Kontrollstruktur **while**
- Ist das Programm für beliebige Zahl x **korrekt**?
- Programm-Test
- Gibt es eine andere Lösung?

Fakultät

- Definition: $fac(x) = x \times (x - 1) \times \dots \times 1$
- Ein Lösungsansatz
 - Nehmen Sie eine beliebige Zahl x und zwei **Variablen** y und z .
 - **Initialisieren** Sie y mit x und z mit 1.
 - Solange $y > 0$ ist, multiplizieren Sie y zu z und reduzieren y um 1.
 - Schreiben Sie das Ergebnis auf.

- Ein Programm

```
public class Fac {
    public static void main(String[] args) {
        int x, y, z;
        x = 15; y = x; z = 1;
        while (y >0) {
            z = y * z;
            y = y-1;
        }
        System.out.println("The faculty of " + x +
            " ist " + z);
    }
}
```

● Fragen

- Warum brauchen wir y ? Kann man sie weglassen?
- Wie läuft das Programm ab?
- Terminiert das Programm immer?
- Wie kann man das Programm verbessern? Z.B. terminiert

Suchen

- Definition: $x \in S$, wobei x is eine Zahl und S eine Menge von Zahlen
- Ein Programm

```
public class Suchen {
    public static void main(String[] args) {
        int x;
        int[] S = {3, 6, -2, 7, 11, 32, -8};
        x = Integer.parseInt(args[0]);
        for (int i=0; i<S.length; i++) {
            if (S[i]==x) {
                System.out.println(x + " ist in der Menge!");
                return; } }
        System.out.println(x + " ist nicht in der Menge!");
    }
}
```

- Discussion

- Reihung

- Initialisierung einer Reihung