

Große Übung Praktische Informatik 1

2005-12-01

Holger Füßler

fuessler@informatik.uni-mannheim.de

<http://www.informatik.uni-mannheim.de/pi4/people/fuessler>

1: Announcements / Orga

Weihnachtsklausur



- zählt als Übungsblatt, d.h. die Punkte werden sowohl im Zähler, als auch im Nenner gutgeschrieben.
- 22ter Dezember

2: Relevantes in Java

Der Execution Stack

- Java verwaltet intern einen Execution Stack, das ist eine Stapel-Datenstruktur, in der Funktionsaufrufe und Rücksprungadressen gespeichert werden.
- Ein Stack-Überlauf entsteht dann, wenn ein Programm zu oft Methoden aufruft; dies kann insbesondere bei Rekursion passieren.

Beispiel: Rekursion

```
package pi1.examples;

public class StackTest {

    public static long rekFakultaet(long n){
3 → Thread.dumpStack();
        if (n == 0)
            return 0;
        if (n == 1)
            return 1;
2 → long retval = rekFakultaet(n - 1);
        Thread.dumpStack();
        return retval;
    }

    public static void main(String[] args) {
1 → Thread.dumpStack();
        rekFakultaet(2);
    }
}
```

java.lang.Thread.dumpStack
(Thread.java:1176)

pi1.examples.StackTest.rekFakultaet
(StackTest.java:6)

pi1.examples.StackTest.rekFakultaet
(StackTest.java:11)

pi1.examples.StackTest.main
(StackTest.java:19)

Output

```
java.lang.Exception: Stack trace
    at java.lang.Thread.dumpStack(Thread.java:1176)
    at pi1.examples.StackTest.main(StackTest.java:17)
java.lang.Exception: Stack trace
    at java.lang.Thread.dumpStack(Thread.java:1176)
    at pi1.examples.StackTest.rekFakultaet(StackTest.java:6)
    at pi1.examples.StackTest.main(StackTest.java:18)
java.lang.Exception: Stack trace
    at java.lang.Thread.dumpStack(Thread.java:1176)
    at pi1.examples.StackTest.rekFakultaet(StackTest.java:6)
    at pi1.examples.StackTest.rekFakultaet(StackTest.java:11)
    at pi1.examples.StackTest.main(StackTest.java:18)
java.lang.Exception: Stack trace
    at java.lang.Thread.dumpStack(Thread.java:1176)
    at pi1.examples.StackTest.rekFakultaet(StackTest.java:6)
    at pi1.examples.StackTest.rekFakultaet(StackTest.java:11)
    at pi1.examples.StackTest.rekFakultaet(StackTest.java:11)
    at pi1.examples.StackTest.main(StackTest.java:18)
java.lang.Exception: Stack trace
    at java.lang.Thread.dumpStack(Thread.java:1176)
    at pi1.examples.StackTest.rekFakultaet(StackTest.java:12)
    at pi1.examples.StackTest.rekFakultaet(StackTest.java:11)
    at pi1.examples.StackTest.main(StackTest.java:18)
java.lang.Exception: Stack trace
    at java.lang.Thread.dumpStack(Thread.java:1176)
    at pi1.examples.StackTest.rekFakultaet(StackTest.java:12)
    at pi1.examples.StackTest.main(StackTest.java:18)
```

```
package pi1.examples;

public class StackTest {

    public static long rekFakultaet(long n){
        Thread.dumpStack();
        if (n == 0)
            return 0;
        if (n == 1)
            return 1;
        long retVal = rekFakultaet(n - 1);
        Thread.dumpStack();
        return retVal;
    }

    public static void main(String[] args) {
        Thread.dumpStack();
        rekFakultaet(2);
    }
}
```

3: Vorlesung

Geom Reihe

- Berühmte Mathematische Reihe
- $g_1; g_2 = g_1 * q; g_3 = g_2 * q; \dots$
- Im folgenden iterative und rekursive Berechnung

4: Übungsblatt 4/A3

Blatt 4 / A4

- Winkel-Klasse Angle implementieren
 - intern rad-Darstellung
- Geo-Koordinaten-Klasse GeoCoord implementieren
 - intern Erdwinkel-Darstellung
 - kann kart Koordinaten berechnen

5: Übungsblatt 6

Hinweise A1

- Bsp. Haare schneiden
 - Vorbereitung -> Ausführung -> Nachbereitung
 - Dann spezialisieren...

Hinweise A2

- größter Gemeinsamer Teiler
- beide Diagramm-Arten
- einfache Java-Funktion

Hinweis A3

- Programmieraufgabe
- Ziel: Einsatz von abstrakten Klassen und Polymorphie

Ende