# Abstrakte Klassen und Methoden

## Abstrakte Klassen und Methoden

Klassen mit Methoden ohne Implementierung ('body')

```
abstract class Benchmark {

    abstract void benchmark();       // nur Signatur

    public final long repeat (int count) {
        long start = System.currentTimeMillis();
        for ( int i = 0; i < count ; i++)
            benchmark();
        return ( System.currentTimeMillis() - start );
    }
}
```

- Jede Klasse mit *abstract*-Methoden muss selbst als *abstract* deklariert sein.
- Von abstrakten Klassen können KEINE Objekte erzeugt werden.

## Unterklassen einer abstrakten Klasse

Jede Unterklasse **muss** die abstrakten Methoden der abstrakten Überklasse implementieren!

```
class TestBenchmark extends Benchmark{

   void benchmark() {         // macht nichts nur
    }                         // Methodenaufruf

    public static void main(String[] args ) {
        int count = Integer.parseInt(args[0]);
        long time = new TestBenchmark().repeat(count)
        System.out.println(count + " method calls in " +
                                time + " Milliseconds" );
    }
}
```

## Interfaces

## Interface, die 'Abstraktion' einer abstrakten Klasse

Wenn in einer "Klasse" **KEINE** Methode implementiert ist, wird sie als interface "Schnittstelle" bezeichnet.

```
public interface  GraphicalComponent {

        void draw();              // nur Signatur
        void rotate(int deg);     // nur Signatur
        void fill(int pattern);   // nur Signatur

}
```

- Alle Methoden sind automatisch *public* .
- Von einem *interface* können KEINE Objekte erzeugt werden.
- Nur *static* und *final* Feldvariablen sind erlaubt.

## Verwendung von Interface

mit implements kann eine Klasse von einem Interface gebildet werden

```
public class Triangle implements GraphicalComponent {

        void draw() { ....}
        void rotate(int deg) { ....}
        void fill(int pattern) { ....}
}
```

- eine Klasse kann gleichzeitig mehrere Interfaces implementieren
  *class MyClass implements A, B*
  (Mehrfachvererbung bei Klassen in Java  nicht erlaubt.)

*abstract* oder *interface*?
wenn keine Methode implementiert werden soll, ist interface zu bevorzugen.