

lokale Variablen landen immer
auf dem Stack

STACK

HEAP

Konsolenaufruf:

java Foo

Die Klasse Foo wird geladen

```
class Foo {  
  
    ... main ... {  
  
        int x;  
        x = 22;  
  
        {  
            int y = 17;  
            y = -5;  
        }  
  
        {  
            int y = 3;  
            y = 1000;  
        }  
  
    }  
}
```

STATIC



Class (für die Klasse Foo)

STACK

HEAP

in die main einsteigen

***Achtung! Das ist keine exakte
komplette Darstellung der geladenen
Objekte***

```
class Foo {  
  
    ... main ... {  
  
        int x;  
        x = 22;  
  
        {  
            int y = 17;  
            y = -5;  
        }  
  
        {  
            int y = 3;  
            y = 1000;  
        }  
    }  
}
```

12000
X
müll
12031

STATIC

Class (für die Klasse Foo)



STACK

HEAP

***Achtung! Alle Werte zu Vereinfachung
werden auf den Folien
dezimal dargestellt
(in Wirklichkeit binär)***

```
class Foo {  
  
    ... main ... {  
  
        int x;  
        x = 22;  
  
        {  
            int y = 17;  
            y = -5;  
        }  
  
        {  
            int y = 3;  
            y = 1000;  
        }  
  
    }  
}
```

12000 12031
x 22

STATIC

Class (für die Klasse Foo)



STACK

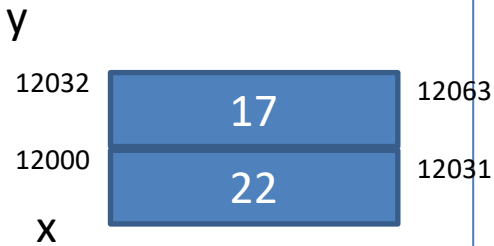
HEAP

Blockvariable

```
class Foo {  
  
    ... main ... {  
  
        int x;  
        x = 22;  
  
        {  
            int y = 17;  
            y = -5;  
        }  
  
        {  
            int y = 3;  
            y = 1000;  
        }  
    }  
}
```

STATIC

Class (für die Klasse Foo)



STACK

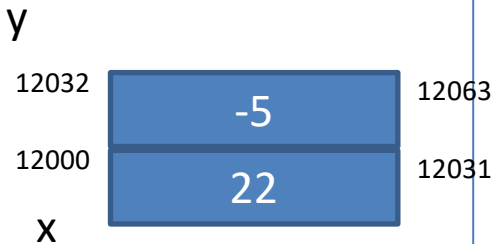
HEAP

Blockvariable

```
class Foo {  
  
    ... main ... {  
  
        int x;  
        x = 22;  
  
        {  
            int y = 17;  
            y = -5;  
        }  
  
        {  
            int y = 3;  
            y = 1000;  
        }  
    }  
}
```

STATIC

Class (für die Klasse Foo)



STACK

HEAP

Blockvariable

```
class Foo {  
  
    ... main ... {  
  
        int x;  
        x = 22;  
  
        {  
            int y = 17;  
            y = -5;  
        }  
        // hier anhalten  
        {  
            int y = 3;  
            y = 1000;  
        }  
    }  
}
```

STATIC

Class (für die Klasse Foo)

12000
x
12031



STACK

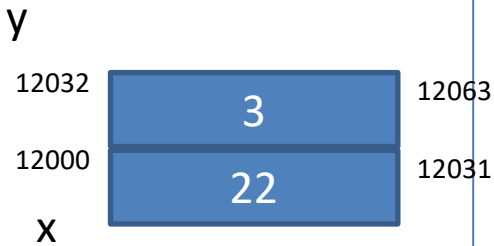
HEAP

Blockvariable

```
class Foo {  
  
    ... main ... {  
  
        int x;  
        x = 22;  
  
        {  
            int y = 17;  
            y = -5;  
        }  
  
        {  
            int y = 3;  
            y = 1000;  
        }  
    }  
}
```

STATIC

Class (für die Klasse Foo)



STACK

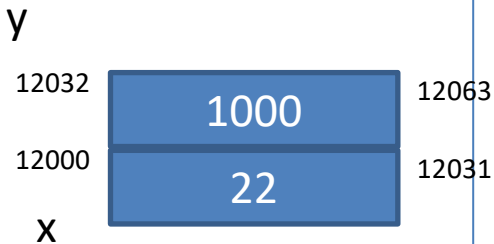
HEAP

Blockvariable

```
class Foo {  
  
    ... main ... {  
  
        int x;  
        x = 22;  
  
        {  
            int y = 17;  
            y = -5;  
        }  
  
        {  
            int y = 3;  
            y = 1000;  
        }  
    }  
}
```

STATIC

Class (für die Klasse Foo)



STACK

HEAP

Blockvariable

```
class Foo {  
  
    ... main ... {  
  
        int x;  
        x = 22;  
  
        {  
            int y = 17;  
            y = -5;  
        }  
  
        {  
            int y = 3;  
            y = 1000;  
        }  
        // hier anhalten  
    }  
}
```

STATIC

Class (für die Klasse Foo)

12000
x
12031



22



Heap

Attribute (wie radius) sind keine lokale Variablen und befinden sich in mit new erstellten Objekten auf dem Heap

STACK

HEAP

Konsolenaufruf:

java Foo

Die Klasse Foo wird geladen

```
class Kreis {  
    int radius;  
}
```

```
class Foo {  
  
    ... main ... {  
  
        new Kreis();  
  
        Kreis k1;  
  
        k1 = new Kreis();  
        k1.radius = 5;  
    }  
}
```

STATIC

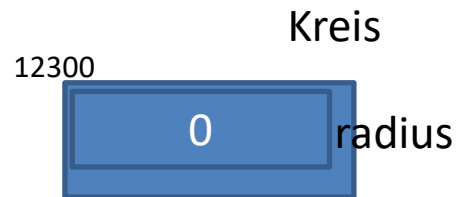
Class (für die Klasse Foo)



STACK

HEAP

1. *Speicher für das neue Kreis reservieren.*
2. *Den reservierten Speicher mit dem Konstruktor initialisieren*



```
class Kreis {  
    int radius;  
}
```

```
class Foo {
```

```
    ... main ... {
```

```
        new Kreis();
```

```
        Kreis k1;
```

```
        k1 = new Kreis();  
        k1.radius = 5;
```

```
    }  
}
```

STATIC

Class (für die Klasse Kreis)



Class (für die Klasse Foo)



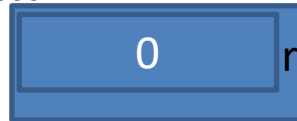
STACK

HEAP

```
class Kreis {  
    int radius;  
}  
  
class Foo {  
  
    ... main ... {  
  
        new Kreis();  
  
        Kreis k1;  
  
        k1 = new Kreis();  
        k1.radius = 5;  
    }  
}
```

Kreis

12300



radius

STATIC

12000

k1



12031

Class (für die Klasse Kreis)



Class (für die Klasse Foo)



STACK

HEAP

*Der new liefert die Anfangsadresse
vom neuen Objekt zurück*

```
class Kreis {  
    int radius;  
}
```

```
class Foo {
```

```
    ... main ... {
```

```
        new Kreis();
```

```
        Kreis k1;
```

```
        k1 = new Kreis();
```

```
        k1.radius = 5;
```

```
    }
```

```
}
```

Kreis

12300

0

radius

Kreis

34567

0

radius

STATIC

Class (für die Klasse Kreis)

Class (für die Klasse Foo)

12000

12031

müll

k1

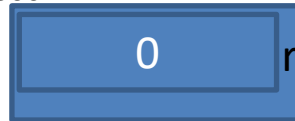
STACK

HEAP

```
class Kreis {  
    int radius;  
}  
  
class Foo {  
  
    ... main ... {  
  
        new Kreis();  
  
        Kreis k1;  
  
        k1 = new Kreis();  
        k1.radius = 5;  
    }  
}
```

Kreis

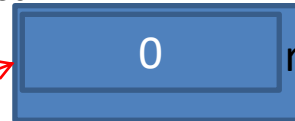
12300



radius

Kreis

34567

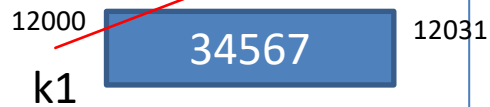


radius

STATIC

Class (für die Klasse Kreis)

Class (für die Klasse Foo)



k1

34567

12031

STACK

HEAP

```
class Kreis {  
    int radius;  
}  
  
class Foo {  
  
    ... main ... {  
  
        new Kreis();  
  
        Kreis k1;  
  
        k1 = new Kreis();  
        k1.radius = 5;  
    }  
}
```

Kreis

12300

0

radius

Kreis

34567

5

radius

STATIC

Class (für die Klasse Kreis)

Class (für die Klasse Foo)

12000

12031

34567

k1

