

Prof. Dr. Verena Majuntke

Grundlagen der Programmierung

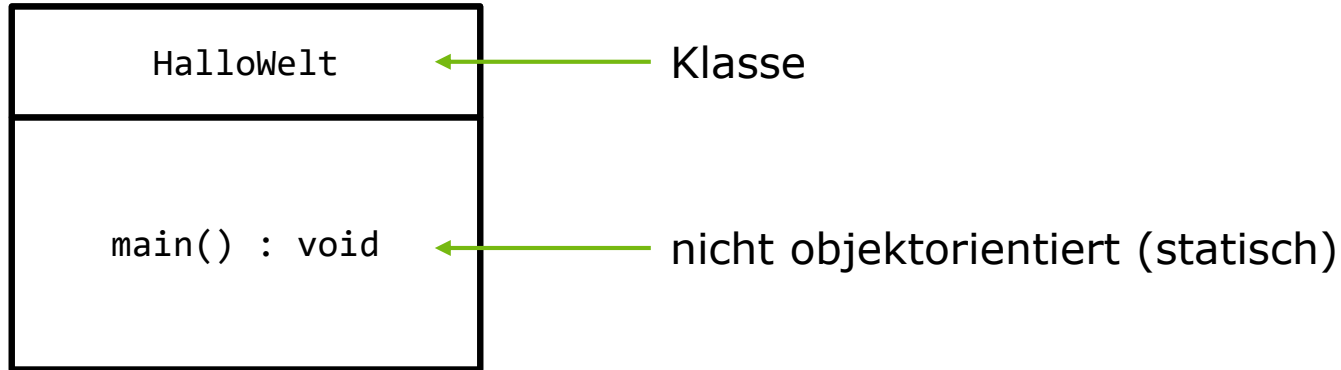
Objektorientierung

08.12.2020

Was haben wir bis heute gelernt?

- Datentypen
- Variablen
- Kontrollstrukturen
- Methoden
 - Umgang mit der main-Methode
 - Eigene Methoden

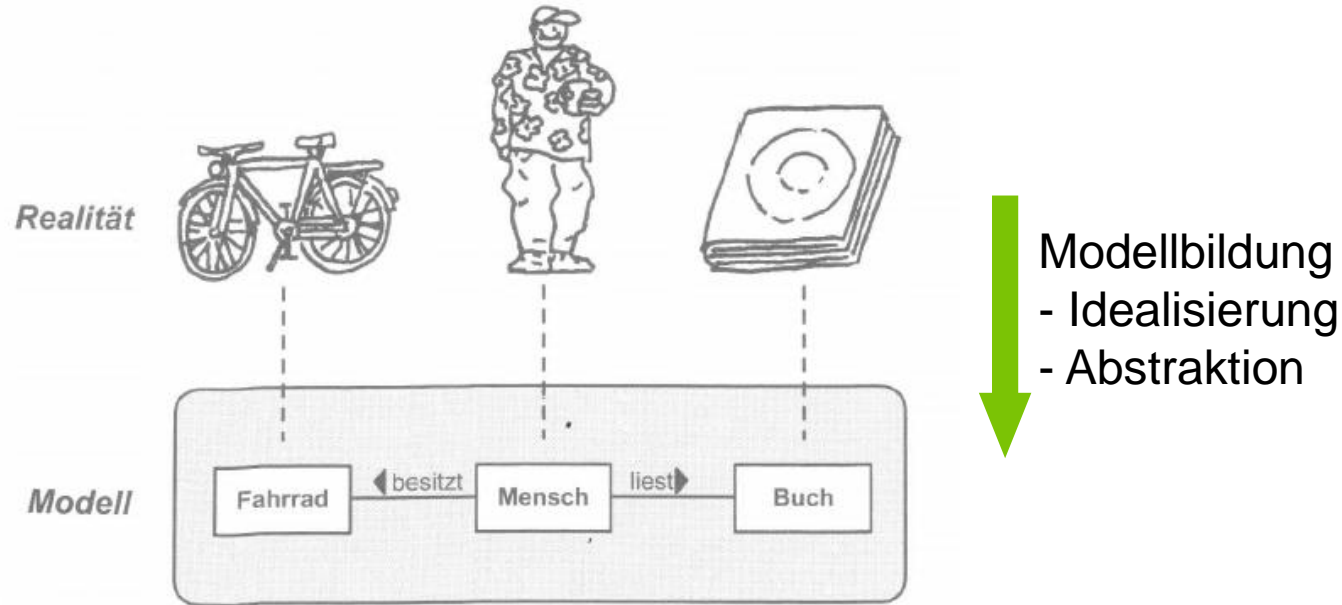
Bis jetzt nicht objektorientiert ...



Bis jetzt nicht objektorientiert ...

```
public class HalloWelt {  
  
    public static void main(String[] args) {  
  
        System.out.println(computeSum(11));  
  
    }  
  
    public static int computeSum(int n) {  
        int sum = 0;  
        for (int i = 0; i <= n; i++) {  
            sum = sum + i;  
        }  
        return sum;  
    }  
}
```

Wie lässt sich die Welt abbilden?



Lernziele

- Welche Merkmale hat ein Klasse?
- Was ist der Unterschied zwischen Klasse und Objekt?
- Wie sehen Klassen, Methoden und Attribute in Java aus?
- Wie konstruiere ich ein Objekt in Java?
- Wie rufe ich eine Methode auf?

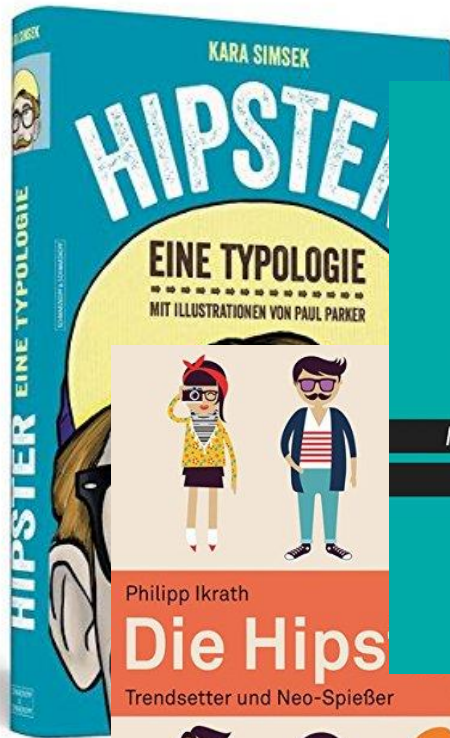
Klassen und Objekte

Klassen und Objekte

- Die Welt besteht aus lauter Objekten, die miteinander in Beziehung stehen
- Man kann ganz konkret über bestimmte Objekte sprechen:
 - „das ist ein schöner Stuhl“
 - „das ist ein teurer Tisch“
 - „das Auto ist blau“
- Oder ganz allgemein über eine Klasse von Objekten:
 - „Studenten sind immer sehr fleißig“
 - „Computer machen immer, was sie wollen“
 - „die Bahnen kommen immer zu spät“







Philipp Ikrath

Die Hips

Trendsetter und Neo-Spießer

PROMEDIA



HIPSTER WIR

DER

Hipster

Eine transatlantische

Diskussion

edition suhrkamp

SV

Hipster - SPIEGEL ONLINE

Hipster-Weihnachten 2016: Selfie an der Krippe SPIEGEL ONLINE - 22.11.2016 . Was sich viele Menschen Jahr für Jahr unter den Tannenbaum stellen, sieht ...

[S spiegel.de/thema/hipster/](https://www.spiegel.de/thema/hipster/)

Studenten in Berlin: Damn, sind wir hip - SPIEGEL ONLINE

Kein Wunder, das Grimm-Zentrum steht in der Hipster-Hauptstadt Berlin. Eine etwas andere Modestrecke. Leben und Lernen. Schlagzeilen I; Wetter I; DAX 12 ...

[S spiegel.de/lebenundlernen/uni/studenten-in-berlin-da...](https://www.spiegel.de/lebenundlernen/uni/studenten-in-berlin-da...)

Hipster: Beim Barte des Ästheten | ZEIT ONLINE

Das Stadtmarketing verschickt an ausländische Journalisten den sogenannten Hipster Guide. Damit sie wissen, wo in Hamburg die coolen Leute leben. Wir haben ...

[Z zeit.de/2015/29/hipster-hamburg-tourismus-guide](https://www.zeit.de/2015/29/hipster-hamburg-tourismus-guide)

Hipster: Ironiker, der Lenz ist da | ZEIT ONLINE

Der Hipster ist ein beliebtes Feindbild. In den Tiraden gegen ihn vermischt sich vulgäre Konsumkritik mit einer kuriosen Beschwörung des Ernstes.

[Z zeit.de/kultur/2013-03/hipster-kritik-zeitgeist](https://www.zeit.de/kultur/2013-03/hipster-kritik-zeitgeist)

Leipzig Is the New Berlin - SPIEGEL ONLINE

Calling All Hipsters Leipzig Is the New Berlin. Berlin used to be Germany's hippest city, but the once scruffy capital has long since succumbed to gentrification.

[S spiegel.de/international/zeitgeist/leipzig-is-the-ne...](https://www.spiegel.de/international/zeitgeist/leipzig-is-the-ne...)

Hipster aus dem Irak: "Aus Ziegenhaar kann man tolle ...

Hipster im Irak Bart in der Menge. 14 Jahre lebte er in Ludwigshafen, jetzt ist Ahmed Nauzad zurück im Irak - und hat dort einen Modeklub für Männer gegründet.

[S spiegel.de/lebenundlernen/job/hipster-aus-dem-irak-a...](https://www.spiegel.de/lebenundlernen/job/hipster-aus-dem-irak-a...)

Aussehen

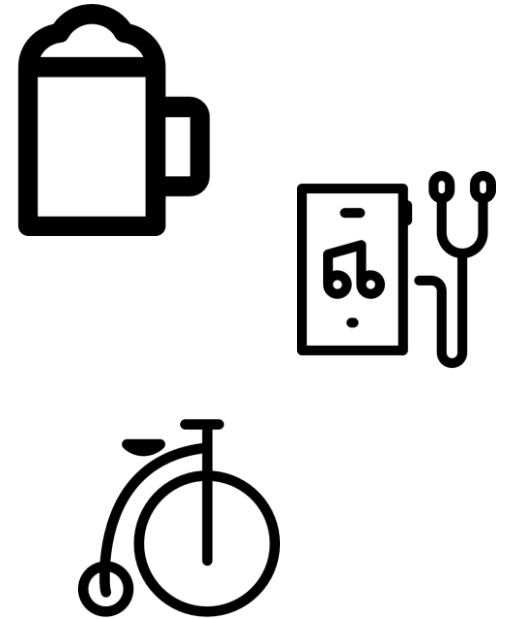
Verhalten



Aussehen



Verhalten



Klasse und Objekt

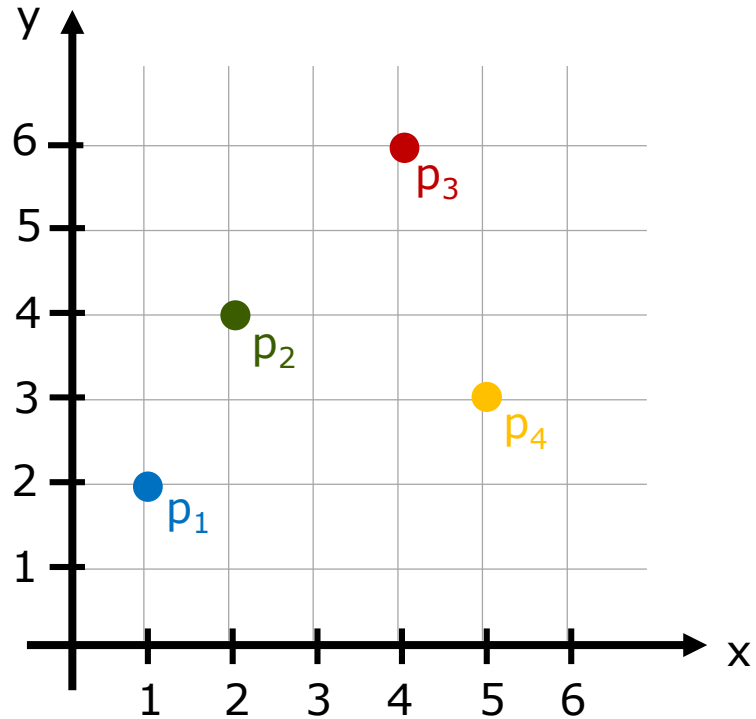
Klasse

- benutzerdefinierter Datentyp
- Bauplan für ein Objekt
- beschreibt welche Eigenschaften (Variablen) ein Objekt hat (**Attribute**)
- definiert was ein Objekt der Klasse tun kann (**Operationen**)

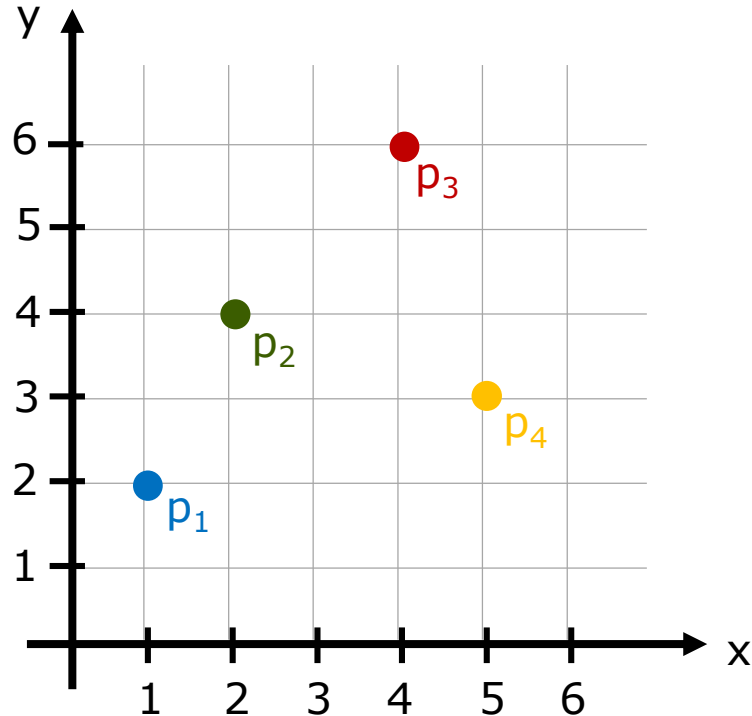
Objekt

- Instanz einer Klasse, d.h. ein konkretes Exemplar
- wird gemäß des Bauplans der Klasse erzeugt
- hat ausgeprägte Attribute

Ein Punkt



Ein Punkt - Eigenschaften



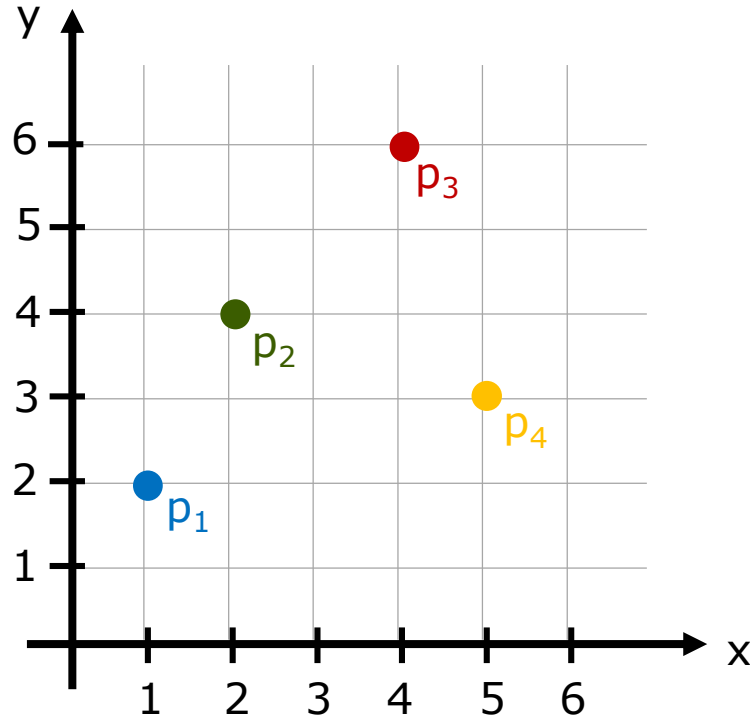
```
int x = 1;  
int y = 2;
```

```
int x = 2;  
int y = 4;
```

```
int x = 4;  
int y = 6;
```

```
int x = 5;  
int y = 3;
```


Ein Punkt - Operationen



Ein Punkt kann:

- gesetzt werden
- verschoben werden
- seine Daten ausgeben

Die Klasse *Point* aus der Standardbibliothek

Klassenname: Point

Attribute:

- x-Koordinate
- y-Koordinate

Operationen:

- Position setzen
- Verschieben
- Daten ausgeben
- ...

Die Klasse *Point* aus der Standardbibliothek

Klassenname: Point

Attribute:

- x-Koordinate
- y-Koordinate

Operationen:

- Position setzen
- Verschieben
- Daten ausgeben
- ...

```
public class Point {  
  
    private int x;  
    private int y;  
  
    ...  
  
}
```

Die Klasse *Point* aus der Standardbibliothek

Klassenname: Point

Attribute:

- x-Koordinate
- y-Koordinate

Operationen:

- Position setzen
- Verschieben
- Daten ausgeben
- ...

```
public class Point {
```

```
    private int x;
```

```
    private int y;
```

```
    ...
```

```
}
```

Deklaration einer Klasse
Point
= es existiert ein neuer
Datentyp Point

Datenfelder für die x- und y-
Koordinate vom Typ int
sog. *Attribute* der Klasse

Breakout Session

Implementieren Sie die Klasse „Lehrveranstaltung“.

- Überlegen Sie sich welche Attribute die Lehrveranstaltung
- Überlegen Sie, ob die Attribute bereits Initialwerte haben
- Überlegen sie, ob die Attribute wiederum Objekte sein könnten/sollten

Exkurs: Sichtbarkeit und Privatsphäre

Sichtbarkeiten in Java:

- öffentlich (**public**) – public-Elemente sind für alle sichtbar
- privat (**private**) – nicht von außerhalb sichtbar
- paketsichtbar (kein Modifizierer)
- geschützt (**protected**) – behandeln wir in Prog2

Klasse *Point* aus der Standardbibliothek

Klassenname: Point

Attribute:

- x-Koordinate
- y-Koordinate

Operationen:

- Position setzen
- Verschieben
- Daten ausgeben
- ...

```
public class Point {  
  
    private int x;  
    private int y;  
  
    ...  
  
}
```

public – Klasse

private – Attribute

Klasse *Point*

```
public class Point {
```

```
    private int x;  
    private int y;
```

} Attribute

```
    public void setX(int value) {  
        x = value;  
    }
```

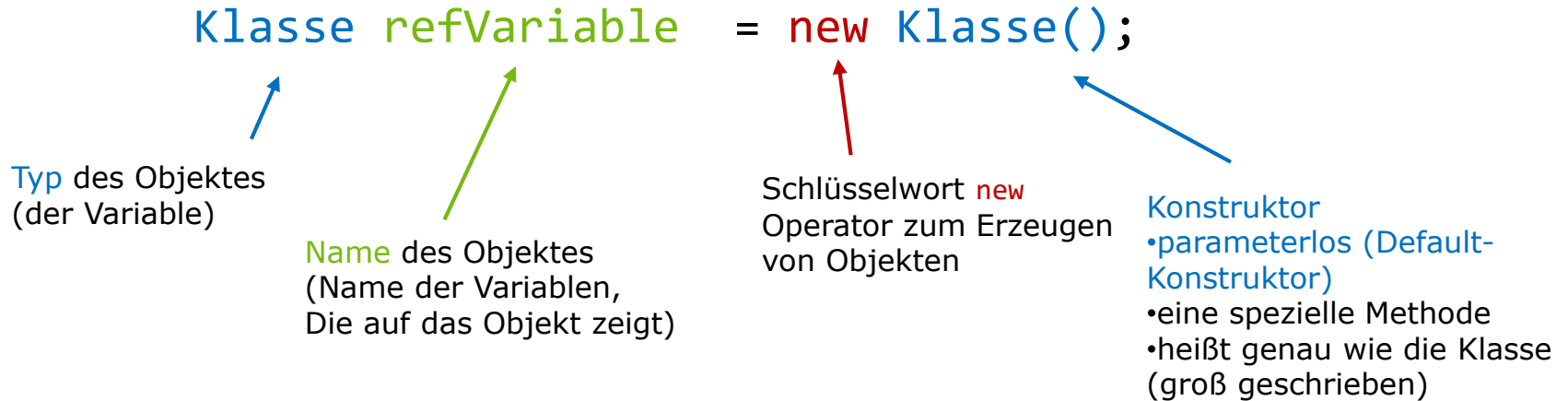
```
    public void setY(int value) {  
        y = value;  
    }
```

} Operationen

```
}
```


Wie werden Objekte erzeugt?

- Mit Aufruf des Konstruktors durch das Schlüsselwort **new**
- Allgemeines Schema:



Klasse *Point*

```
public class Point {
```

```
    private int x;
```

```
    private int y;
```

```
    public Point() {}
```

```
    public void setX(int value) {
```

```
        x = value;
```

```
    }
```

```
    public void setY(int value) {
```

```
        y = value;
```

```
    }
```

```
}
```

}

Attribute

}

Konstruktor

}

Operationen

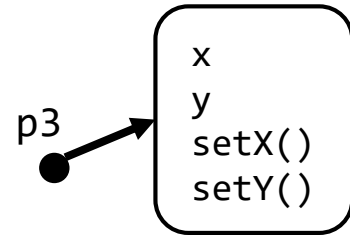
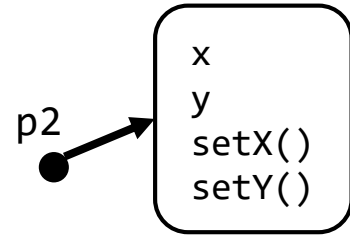
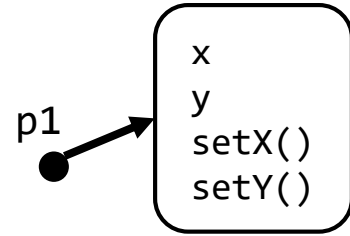
Erzeugen von Objekten: Konstruktor

```
Point p1 = new Point();
```

- Variablendeklaration:
 - `Point p1;`
 - Es existiert eine Variable vom Typ Point (aber noch kein Objekt)
- Objekterzeugung
 - `p1 = new Point();`
 - es wurde ein Point-Objekt erzeugt
 - über p1 kann auf dieses Objekt zugegriffen werden

Beispiel

```
public class PointTest {  
  
    public static void main(String[] args)  
    {  
        Point p1 = new Point();  
        Point p2 = new Point();  
        Point p3 = new Point();  
    }  
}
```



Aufgaben des Konstruktors

- Anlegen des **Speichers** für Attribute
- **Initialisieren** der Attribute
- Für Konstruktoren gilt:
 - Sie tragen immer den Namen der Klasse
 - Sie können überladen werden (mehrere Konstruktoren)
 - Sie besitzen keinen Rückgabewert
- Einen **Destruktor** gibt es in Java nicht → Garbage Collection

Objekte manipulieren: Methoden

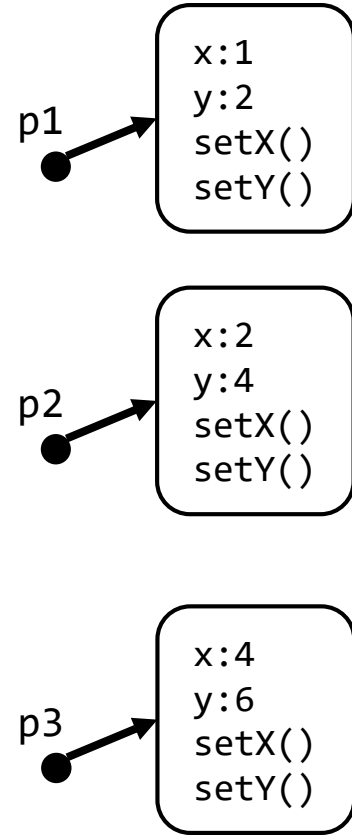
`refVariable.methode()`

 . - Operator

- **Methode** wird immer auf ein **Objekt** in der Klasse angewendet
- Beispiel:
 - `Point p1 = new Point();`
 - x-Koordinate von p1 auf den Wert 1 setzen: `p1.setX(1);`
 - y-Koordinate von p1 auf den Wert 2 setzen: `p1.setY(2);`
 - Point-Daten ausgeben: `p1.print();`
- → Methoden werden stets auf **ein** konkretes Objekt (**refVariable**) angewendet

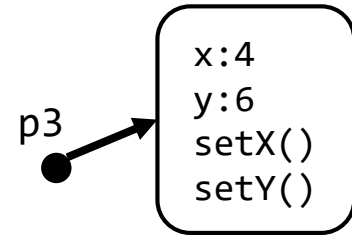
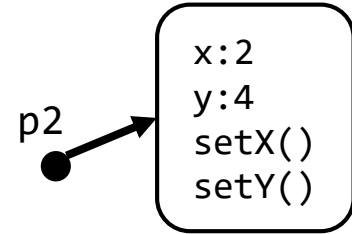
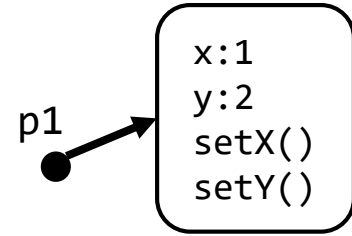
Beispiel

```
public class PointTest {  
  
    public static void main(String[] args)  
    {  
        Point p1 = new Point();  
        Point p2 = new Point();  
        Point p3 = new Point();  
        p1.setX(1); p1.setY(2);  
        p2.setX(2); p2.setY(4);  
        p3.setX(4); p3.setY(6);  
    }  
}
```



Exkurs: Referenzen

Im bereitgestellten Video finden Sie hier eine kurze Diskussion über Referenzen.



Lernziele

- Welche Merkmale hat ein Klasse?
- Was ist der Unterschied zwischen Klasse und Objekt?
- Wie sehen Klassen, Methoden und Attribute in Java aus?
- Wie konstruiere ich ein Objekt in Java?
- Wie rufe ich eine Methode auf?



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

www.htw-berlin.de