hhu.

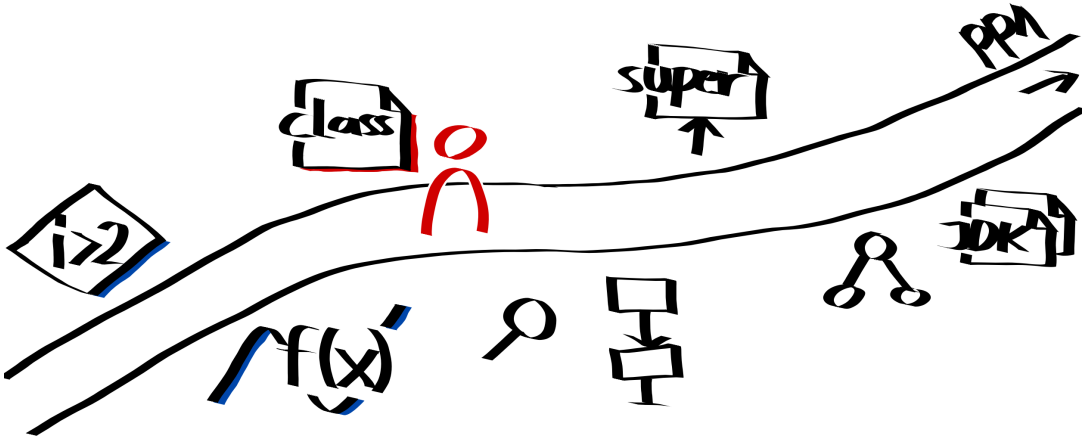
Kapitel 3: Objekte, Speicher & Klassen

VL 13: Arrays von Objekten & null

Wintersemester 2023/24

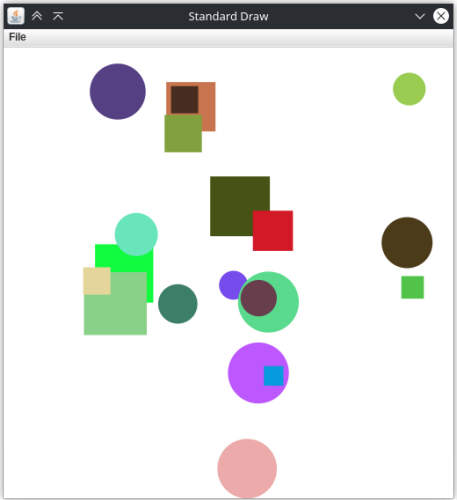
Stand 20. November 2023, 17:13 Uhr

Wo stehen wir gerade?



Vorlesungsfeedback

Unser Ziel:

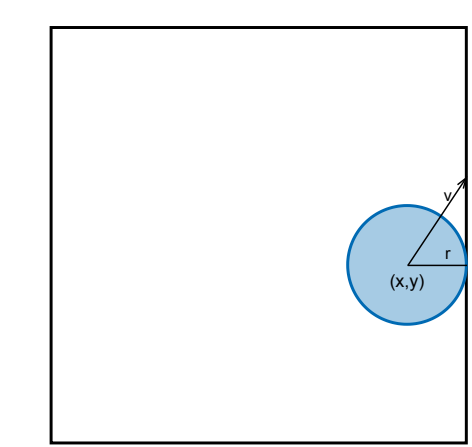


Planung: Was brauchen wir alles?

Planung: Was brauchen wir alles?

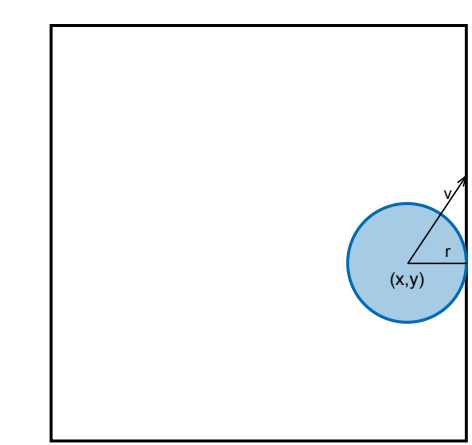
- Kreise
  - Position (2D-Punkt)
  - Geschwindigkeit (2D-Vektor)
  - Farbe
  - Radius
  - sich zeichnen
  - sich bewegen
- Animationsschleife
- Quadrate

Kollision mit rechter Wand



Wie erkennen?

Kollision mit rechter Wand

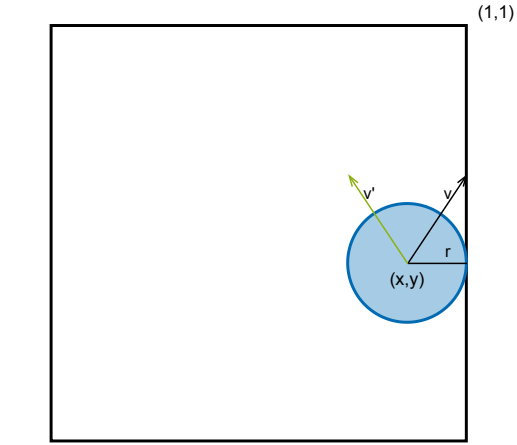


Wie erkennen?

$x + r \geq 1$

Was tun?

Kollision mit rechter Wand



Wie erkennen?
$x + r \geq 1$

Was tun?
Vorzeichen des Richtungsvektors in x-Komponente umdrehen

null-Referenz

- Ein Array von Integern hat als Standardwerte 0

```
4 int[] numbers = new int[10];
5 System.out.println(numbers[0]);
```

```
0
```

- Ein Array von Objekt-Typen hat als Standardwerte ?

```
7 String[] strings = new String[10];
8 System.out.println(strings[0]);
```

null-Referenz

- Ein Array von Integern hat als Standardwerte 0

```
4 int[] numbers = new int[10];
5 System.out.println(numbers[0]);
```

```
0
```

- Ein Array von Objekt-Typen hat als Standardwerte ?

```
7 String[] strings = new String[10];
8 System.out.println(strings[0]);
```

```
null
```

- `null`: spezielle „Referenz“, die auf nichts zeigt

Was passiert hier?

Heinrich Heine Universität Düsseldorf

Null.java

```java
public class Null {
    public static void main(String[] args) {
        String name = null;

        System.out.println(name.toUpperCase());
    }
}
```

Was passiert hier?

Heinrich Heine Universität Düsseldorf

Null.java

```java
public class Null {
    public static void main(String[] args) {
        String name = null;

        System.out.println(name.toUpperCase());
    }
}
```

```
% java Null
Exception in thread "main" java.lang.NullPointerException
        at Null.main(Null.java:5)
```

- `NullPointerException`: Laufzeit-Fehler, wenn Methoden auf Variable mit null-Wert aufgerufen wird

Überprüfung auf null

- Wenn nicht 100 %ig sicher, dass Referenz nicht `null` ist: prüfen!

NullCheck.java — java

```java
public class Null {

    public static void main(String[] args) {
        String name = null;

        if (name != null) {
            System.out.println(name.toUpperCase());
        }
    }

}
```

Online-Shop

- Verkauft zwei Produkte:
  - Fahrräder (249,99 €)
  - Helme (29,99 €, in beliebigen Farben)
- Warenkorb:
  - aus einem Produkt oder Liste von Produkten erstellbar
  - berechnet Gesamtpreis

Zuweisung zu Interface-Typ: Verlust von Verhalten

- nur Methoden des Interfaces verwendbar
- Umgehen: Abfrage des Typs zur Laufzeit + expliziter Cast
- Praxis: Casts nur in Ausnahmefällen sinnvoll, normalerweise reichen Methoden des Interfaces

```
1  Produkt[] produkte = new Produkt[2];
2  produkte[0] = new Helm(Color.BLACK);
3  produkte[1] = new Fahrrad();
4
5  // funktioniert, da getPreis in Typ Produkt vorhanden (deklarierter Typ zählt)
6  System.out.println(produkte[0].getPreis());
7  // funktioniert nicht, da getFarbe nicht in Typ Produkt vorhanden
8  // System.out.println(produkte[0].getFarbe());
9  // Abfrage zur Laufzeit: Ist in produkte[0] eine Instanz der Klasse Helm?
10 if(produkte[0] instanceof Helm) {
11     // funktioniert: expliziter Cast auf anderen Typen zur Laufzeit
12     System.out.println(((Helm)produkte[0]).getFarbe());
13 }
```

Überladen von Konstruktoren

- Konstruktoren können auch überladen werden
- `this` ermöglicht Aufruf eines anderen Konstruktors derselben Klasse
  - nur als erstes Statement im Konstruktor erlaubt

```
3  class Helm {
4      private final double preis;
5      private final Color color;
6
7      // Helm mit Farbe nach Wahl anlegen
8      public Helm(Color color) {
9          this.color = color;
10         this.preis = 29.99;
11     }
12
13     // Helm mit Standardfarbe Schwarz anlegen
14     public Helm() {
15         this(new Color(0, 0, 0));
16     }
17 }
```

Lernziele

Heinrich Heine
Universität Düsseldorf

Sie können am Ende der Woche ...

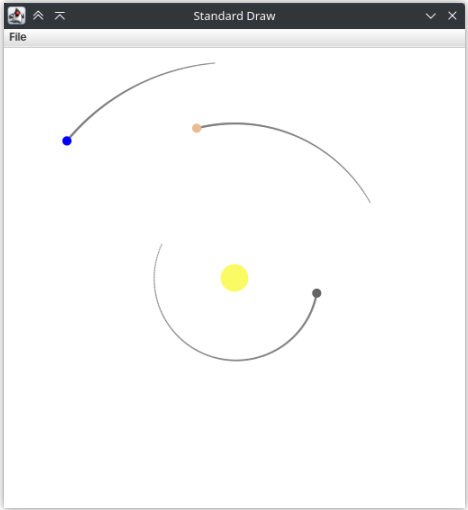
- Polymorphie **benutzen**, um redundanten Code zu vermeiden.
- Klassen **entwerfen**, um Objekte der realen Welt abzubilden.
- **überprüfen**, ob eine Variable mit Objekt-Typ eine Null-Referenz enthält.
- Konstruktoren **überladen**.

Wichtige Begriffe

Heinrich Heine
Universität Düsseldorf

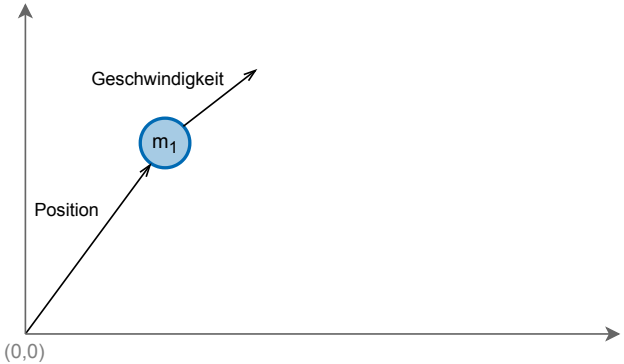
null this()

Simulating the Universe



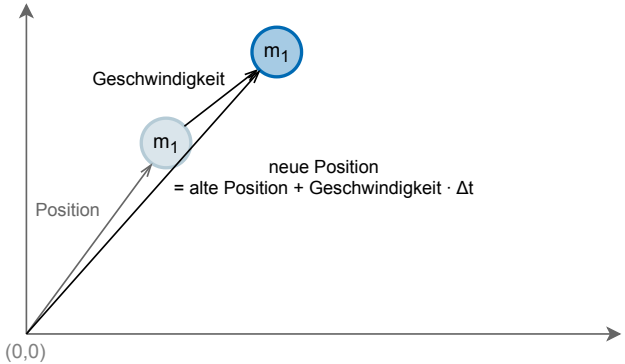
Bewegung eines Körpers (ohne Krafteinwirkung)

- Geschwindigkeit: Betrag + Richtung $\Rightarrow$ Vektor
- Position nach Zeit Δt berechenbar



Bewegung eines Körpers (ohne Krafteinwirkung)

- Geschwindigkeit: Betrag + Richtung $\Rightarrow$ Vektor
- Position nach Zeit Δt berechenbar



Gravitationsgesetz

- Massen ziehen sich gegenseitig an
- Kraft: Betrag + Richtung $\Rightarrow$ Vektor
- Gesamtkraft auf Masse = Vektor-Summe aller einwirkenden Kräfte

$$|\vec{F}| = \frac{G m_1 m_2}{r^2}$$

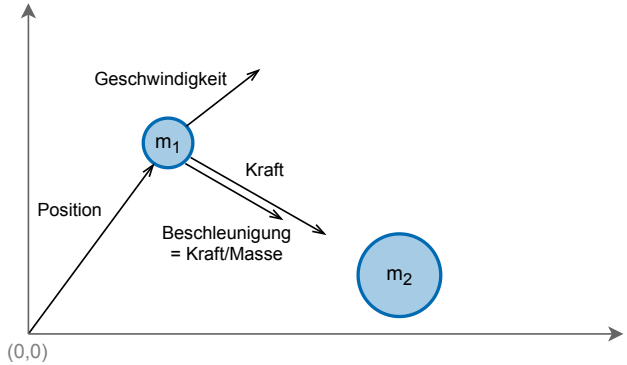
$$G = 6{,}67430 \frac{\mathrm{m}^2}{\mathrm{kg} \cdot \mathrm{s}^2}$$

Auswirkungen auf Geschwindigkeit

- Geschwindigkeit ändert sich stetig durch Einwirkung von Kräften

→ Näherung: Betrachte Geschwindigkeitsänderung nach Zeitspanne Δt

- Beschleunigung: Betrag ($\frac{F}{m}$) + Richtung (normalisierte Differenz der Positionen) $\Rightarrow$ Vektor

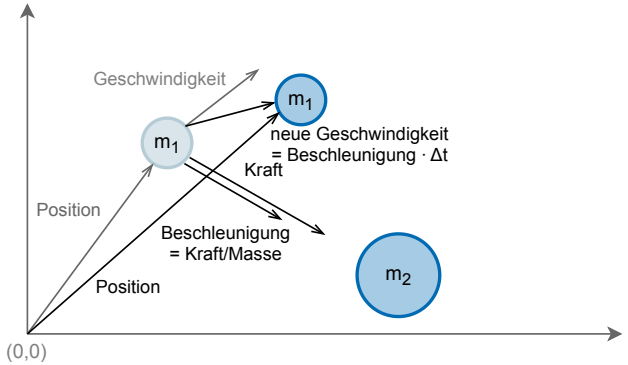


Auswirkungen auf Geschwindigkeit

- Geschwindigkeit ändert sich stetig durch Einwirkung von Kräften

→ Näherung: Betrachte Geschwindigkeitsänderung nach Zeitspanne Δt

- Beschleunigung: Betrag ($\frac{F}{m}$) + Richtung (normalisierte Differenz der Positionen) $\Rightarrow$ Vektor



Planung

- Vektoren:
  - Eigenschaften: x, y
  - Verhalten: addieren, subtrahieren, Betrag bestimmen, skalarmultiplizieren
- Massereiche Körper:
  - Eigenschaften: Position, Geschwindigkeit, Masse
  - Verhalten: zeichnen, bewegen, Anziehung berechnen
- Sonnensystem:
  - Eigenschaften: Sonne, Merkur, Venus, Erde, ...
  - Verhalten: zeichnen, Zeit weiterbewegen
- Animationsschleife

Planeten im Sonnensystem

Körper	Abstand zur Sonne	Geschwindigkeit	Masse
Sonne	$0\,\mathrm{m}$	$0\,\frac{\mathrm{m}}{\mathrm{s}}$	$1{,}99 \cdot 10^{30}\,\mathrm{kg}$
Merkur	$58 \cdot 10^{9}\,\mathrm{m}$	$47 \cdot 10^{3}\,\frac{\mathrm{m}}{\mathrm{s}}$	$3{,}30 \cdot 10^{23}\,\mathrm{kg}$
Venus	$108 \cdot 10^{9}\,\mathrm{m}$	$35 \cdot 10^{3}\,\frac{\mathrm{m}}{\mathrm{s}}$	$4{,}88 \cdot 10^{24}\,\mathrm{kg}$
Erde	$150 \cdot 10^{9}\,\mathrm{m}$	$30 \cdot 10^{3}\,\frac{\mathrm{m}}{\mathrm{s}}$	$5.97 \cdot 10^{24}\,\mathrm{kg}$