
PAC-MAN
READY!

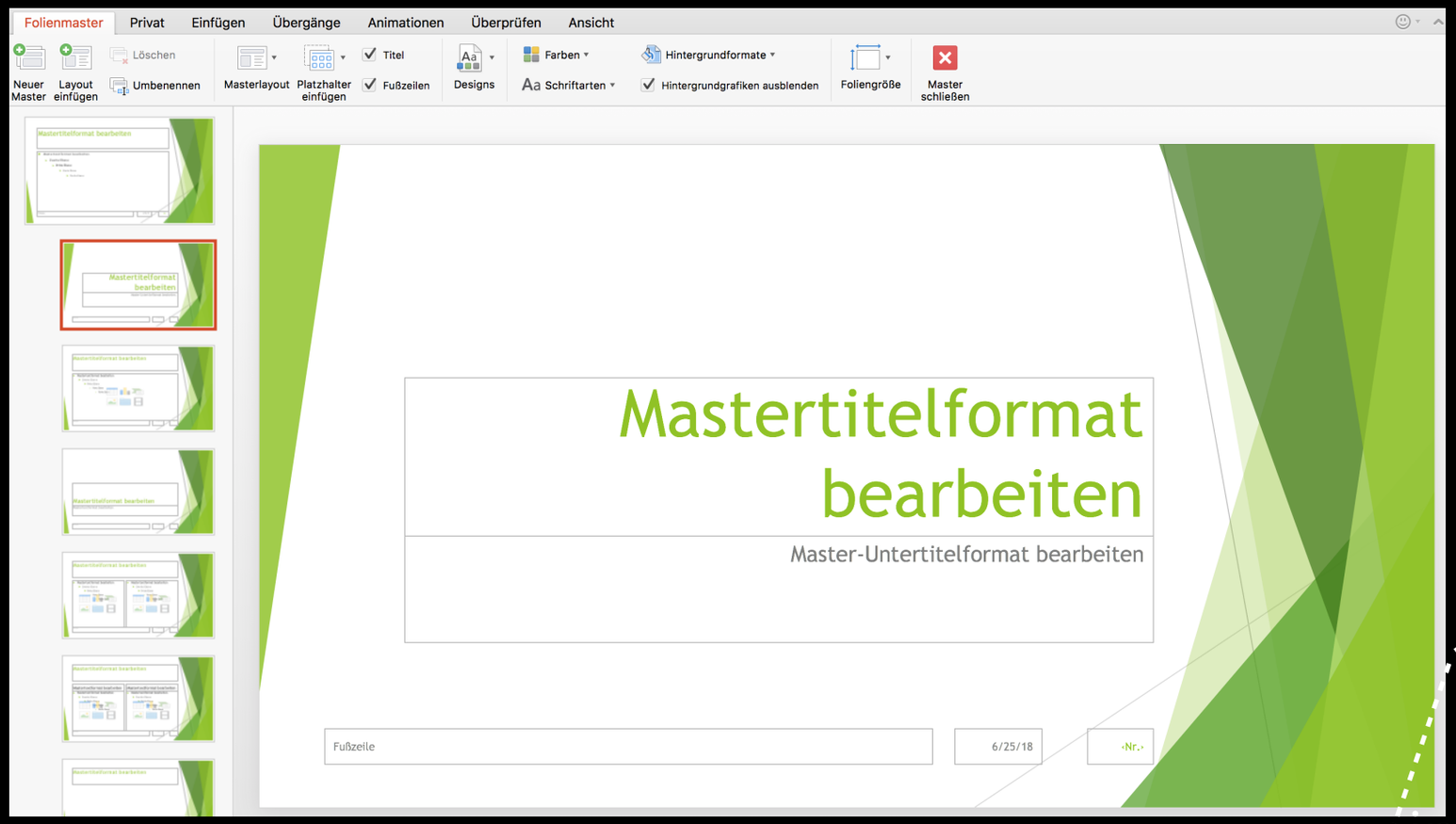
- Lade pacman.pde herunter
- Öffne den Sketch und füge da einen gelben Kreis hinzu, den man über die Tastatur bewegen kann.
- Füge 10 rote Quadrate hinzu über die man mit dem Kreis nicht drüber fahren kann.
- Zusatz: füge eine Animation hinzu, wenn man mit dem Kreis an ein Rechteck stösst.

BRÜCKENKURS PROGRAMMIEREN - FIONA NÜESCH

KLASSEN UND OBJEKTE

more: https://processing.org/tutorials/objects/

Klasse (Template)

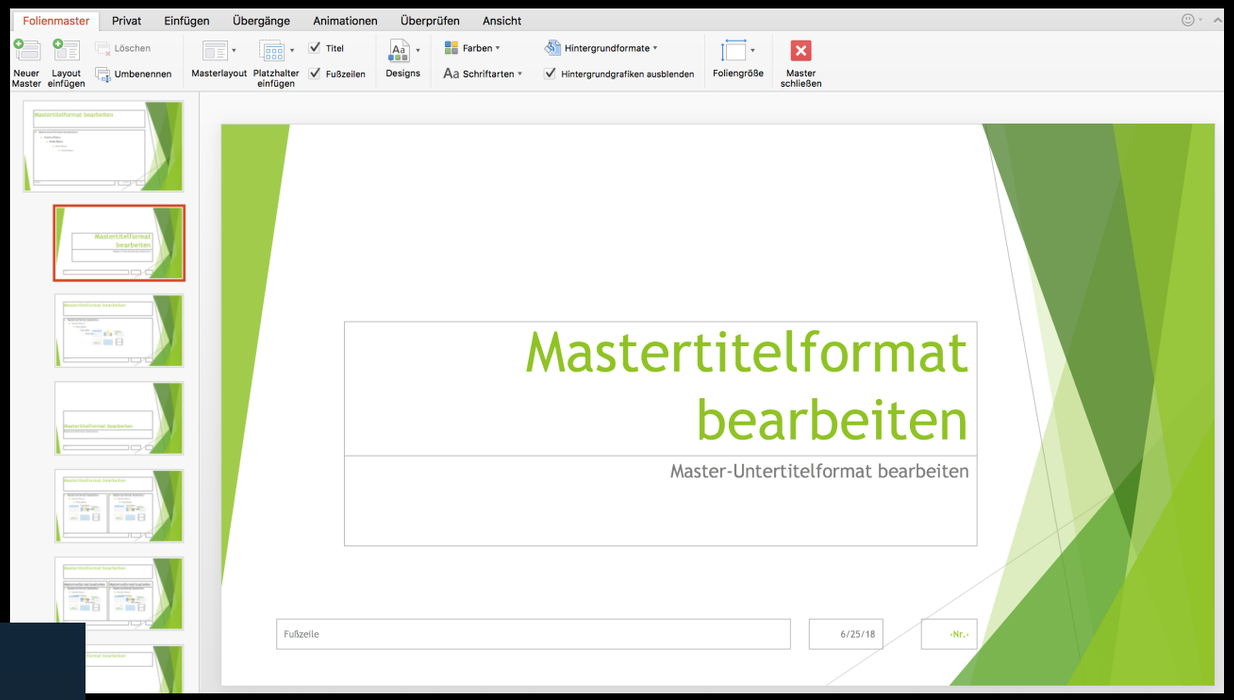


instanziieren

Objekte / Instanzen

Klasse (Template)

Jede Klasse kommt in ein eigenes File mit dem Klassennamen



sketch_180625c | Titelfolie

```
1  class Titelfolie{
2
3
4
5
6
7
8     // Klassen code
9
10
11
12
13 }
```

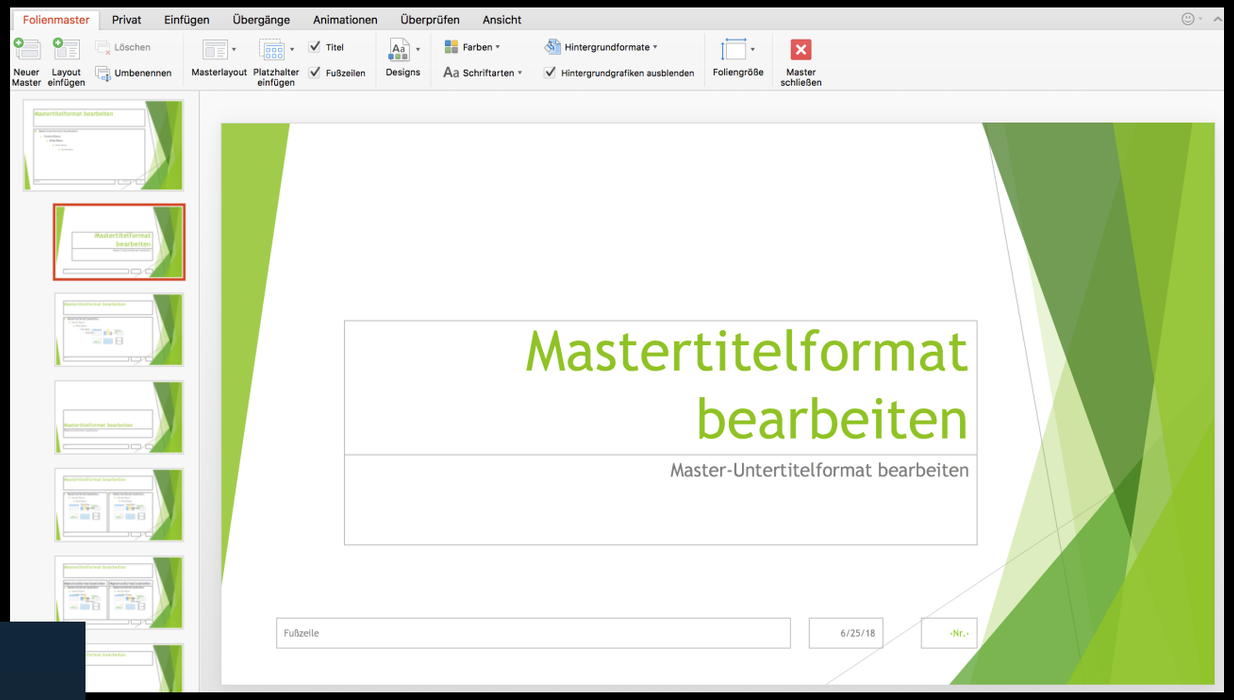
class KLASSENNAMEN {

Klassen Körper (class body)

}

Klasse (Template)

Jede Klasse kommt in ein eigenes File mit dem Klassennamen



sketch_180625c | Titelfolie

```
1  class Titelfolie{
2
3    String titleText;
4    String subtitleText;
5
6
7
8
9
10
11
12
13 }
```

class KLASSENNAMEN {

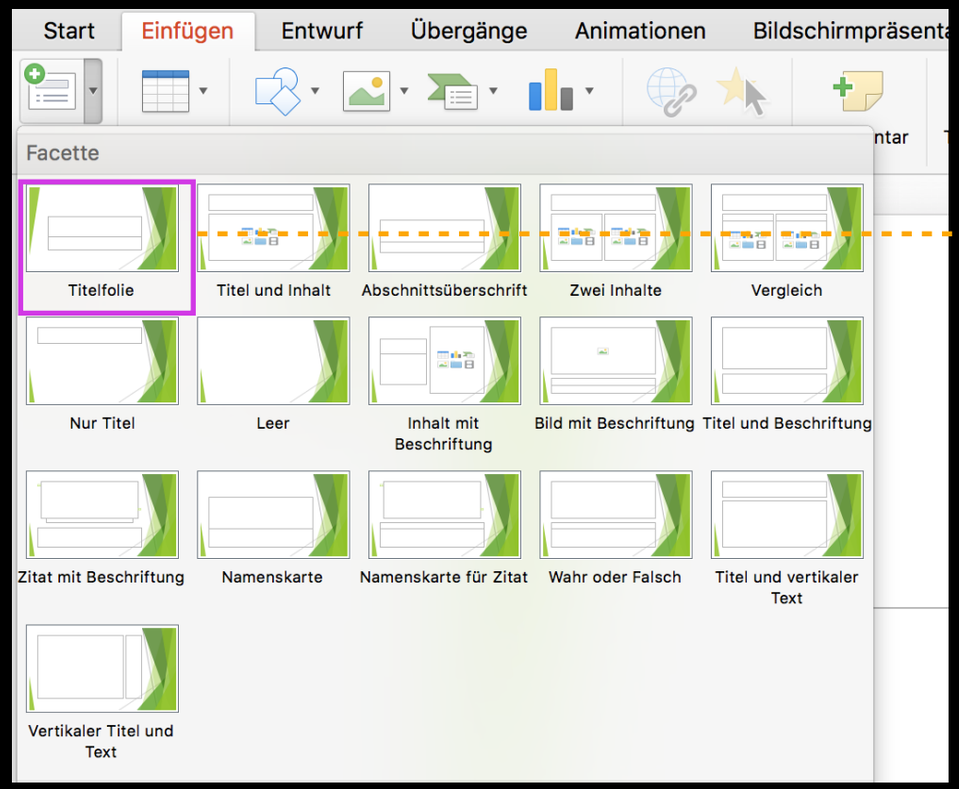
Attribute (class member): in Klasse gültige globale Variablen, speichern Zustand

Methoden: definieren Funktionalitäten

}

Objekt instanziieren (Objekt / Instanz erstellen)

Neuer Datentyp Titelfolie



Objekt folie1


```
1  void setup(){
2
3    Titelfolie folie1 = new Titelfolie();
4
5  }
```

Objekt **Methode ausführen**

PUNKT

verbindet Objektname mit Methodenname **ohne** Abstand

```
7    folie1.setSubtitleText("Fiona Nüesch");
```

Objektname Methodenname

Klasse mit Methode definieren

sketch_180625c | Titelfolie

```
1  class Titelfolie{
2
3    String titleText;
4    String subtitleText;
5
6    void setTitleText(String text){
7      titleText = text;
8    }
9
10   void setSubtitleText(String text){
11     subtitleText = text;
12   }
13 }
```

Ein Objekt mit Name folie1 erstellen

```
1  void setup(){
2
3    Titelfolie folie1 = new Titelfolie();
4
5  }
```

Methode aufrufen

```
7    folie1.setSubtitleText("Fiona Nu
```

Die Klasse definiert einen neuen Datentyp Titelfolie, welcher über die Methode setSubtitleText verfügt. Wenn ein Objekt dieses Datentyps instanziiert/erstellt wird kann auf diese Methode zugegriffen werden. Diese führt **auf diesem Objekt** die in der Klasse definierte Funktionalität aus.

Objekt **Methoden aufrufen**, die Zustand ändern


```
5    folie1.setTitleText("Brückenkurs Programmieren");
```

```
7    folie1.setSubtitleText("Fiona Nüesch");
```

Objekte instanziieren

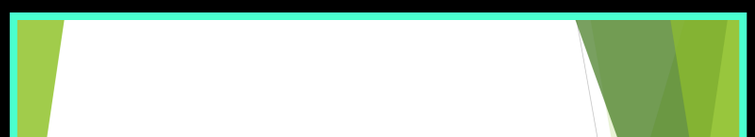
```
Titelfolie title1 = new Titelfolie();
Titelfolie title2 = new Titelfolie();
Titelfolie title3 = new Titelfolie();
```

Methoden aufrufe

```
title1.setTitleText("Brückenkurs Programmieren");
title1.setSubtitleText("Fiona Nüesch");
```


```
title2.setTitleText("Babedi Bubedi");
title2.setSubtitleText("WM ohne Italien");
```

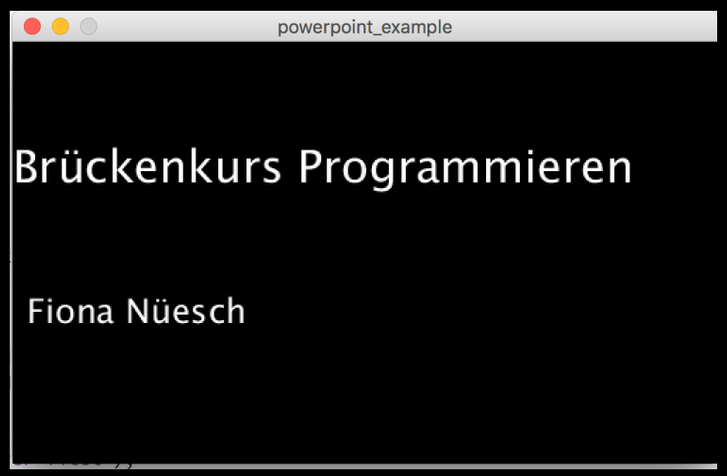

```
title3.setTitleText("anderer Titel");
title3.setSubtitleText("und anderer Subtitel");
```



Übung zusammen

- Wir erstellen die Klasse „Titelfolie“ aus den Folien
- Füge der Klasse eine Methode `void show()` an, die die Folie zeichnet. Tipp: textSize(size) und text(text, x, y)
- Erstelle in `setup()` eine Folie und setzte Titel und Subtitel. Zeichne diese Folie in `draw()`

Output einfach



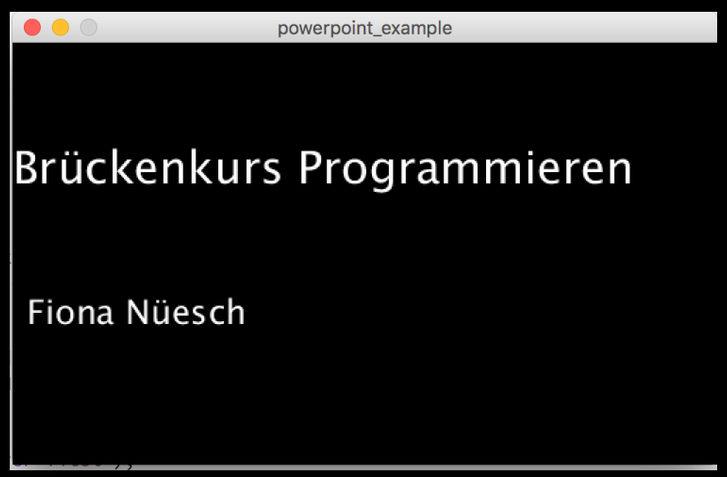
Output Zusatz

Tipp: https://processing.org/reference/image_.html

Übung zusammen

- Instanziiere 3 Folien. Zeichne jeweils die erste wenn der Benutzer ‚1‘ eingibt, die zweite wenn er ‚2‘ eingibt und die dritte wenn er ‚3‘ eingibt.

Output einfach



Output Zusatz

Tipp: https://processing.org/reference/image_.html

„ Das Lernen allein genügt nicht, sondern man muss auch die Gewöhnung hinzunehmen und dann die Übung..“

– EPIKTET

KONSTRUKTOR

- wird immer und nur nach **new** Statement gebraucht **um** ein **Objekt zu erzeugen/ instanziieren**
- wenn wir keinen selber schreiben wird immer der default constructor verwendet
- ist dazu da, den Grundzustand des Objektes zu setzen

Default Constructor

```
Titelfolie title1 = new Titelfolie();
```

Name der Klasse, keine Parameter

Konstruktor definieren

```
1  class Titelfolie{
2
3    // Klassen Attribute
4    String titleText;
5    String subtitleText;
6
7    // Konstruktor
8    Titelfolie(String title, String subtitle){
9      this.titleText = title;
10     this.subtitleText = subtitle;
11   }
12
13   Titelfolie(String title){
14     this.titleText = title;
15   }
16
17
18   // Klassen Funktionalitäten
19   void setTitleText(String text){
```

- wird wie eine Methode aber ohne Rückgabe definiert.
- Der Name entspricht immer dem Klassennamen.
- Pro Klasse können mehrere verschiedene definiert werden
- -> verschieden heisst sie müssen sich bei den Parametern unterscheiden

Konstruktor benutzen

```
folie1 = new Titelfolie("Babedi Bubedi", "WM ohne Italien");
```

wird wie eine Methode benutzt, nur das die Rückgabe immer ein neues Objekt vom genannten Typ ist.

sobald wir einen eigenen Konstruktor definiert haben, kann der default constructor nicht mehr verwendet werden.

this

```
3   // Klassen Attribute
4   String titleText;
5   String subtitleText;
6
7   // Konstruktor
8   Titelfolie(String title, String subtitle){
9     this.titleText = title;
10    this.subtitleText = subtitle;
11  }
```

- das **this** Statement steht für diese Objekt und hilft uns im Konstruktor die Klassen Attribute von den Konstruktor Parametern zu unterscheiden
- kann überall in der Klasse verwendet werden.

wäre auch so möglich

```
5   // Klassen Attribute
6   String titleText;
7   String subtitleText;
8
9
10  // Konstruktor
11  Titelfolie(String title, String subtitle){
12    titleText = title;
13    subtitleText = subtitle;
14  }
```

Problem

```
5   // Klassen Attribute
6   String titleText;
7   String subtitleText;
8
9   // Konstruktor
10  Titelfolie(String titleText, String subtitleText){
11    titleText = titleText;
12    subtitleText = subtitleText;
13  }
```

Problem gelöst

```
5   // Klassen Attribute
6   String titleText;
7   String subtitleText;
8
9   // Konstruktor
0   Titelfolie(String titleText, String subtitleText){
1     this.titleText = titleText;
2     this.subtitleText = subtitleText;
3   }
```

Übung 10 Konstruktoren

„ Work hard. Play Hard.“

–INTERNET

Mit einer Klasse definieren wir einen neuen Datentyp

```
Titelfolie title1 = new Titelfolie();
```

```
Titelfolie folie1 = new Titelfolie("Babedi Bubedi", "WM ohne Italien"
```

Das heisst wir können einen Array erstellen

```
Titelfolie[] folien = new Titelfolie[10];
```

Datentyp

```
folien[0] = new Titelfolie("Title", "Sub");
```

Konstruktor Aufruf -> neues Objekt

Und generell wie Datentypen benutzen die wir schon kennen

powerpoint_example | Presentation | Titelfolie

```
1  class Presentation{
2    Titelfolie title;
3
4    Presentation(String title, String subtitle){
5      this.title = new Titelfolie(title, subtitle)
6    }
7
8    Presentation(Titelfolie title){
9      this.title = title;
10   }
11
12   Titelfolie getTitle(){
13    return title;
14   }
15 }
```

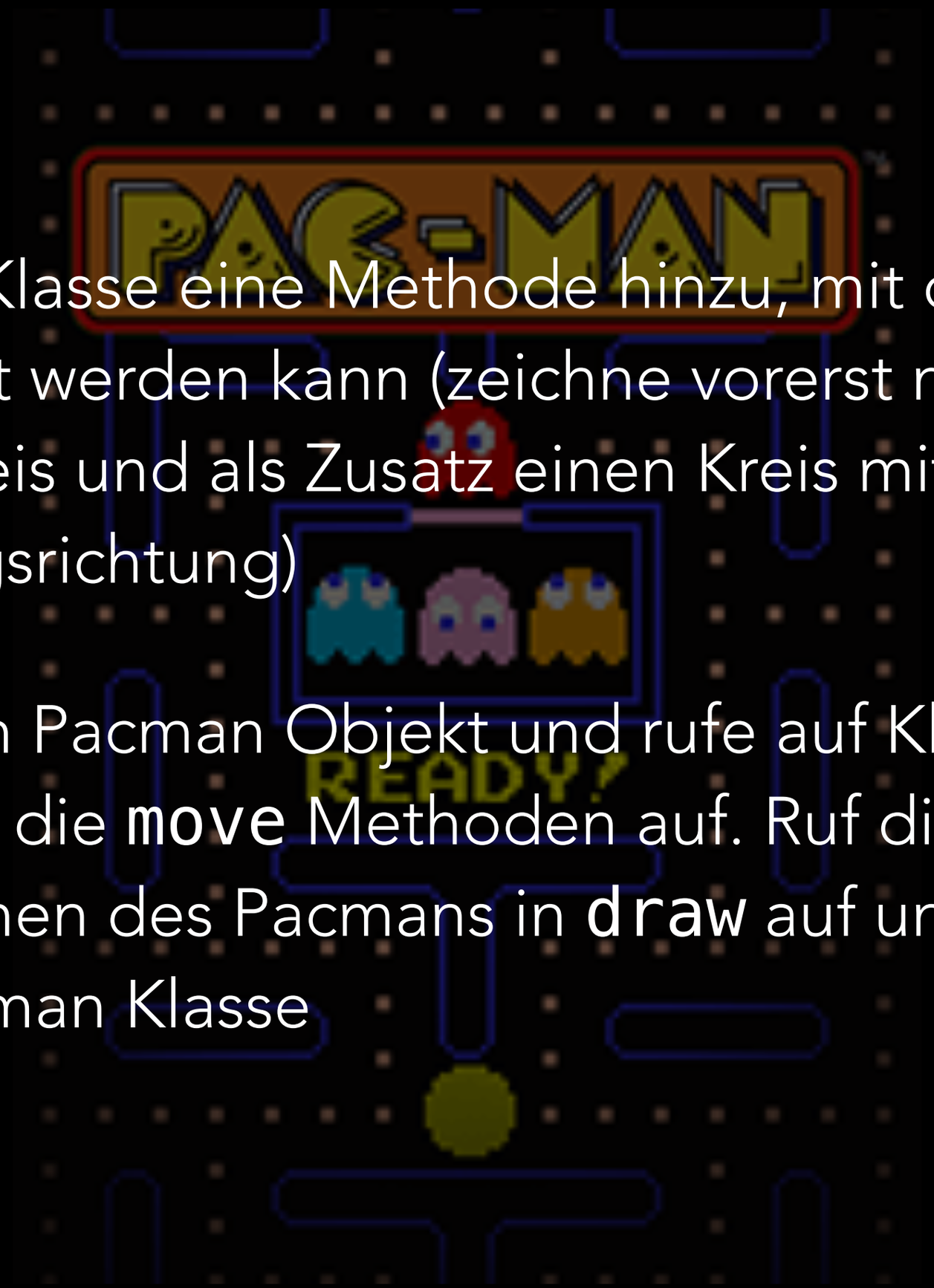
Konstruktor Aufruf -> ein Objekt wird erstellt

Übung 11 Klassen benutzen

„Oft schiessen trifft das Ziel.“

– SPRICHWORT

- Erstelle eine Klasse `Pacman` die deine Pacman Figur repräsentieren soll.
- mit dem Attribut `Point coordinates` für die Position
- mit den Methoden: `moveLeft()`, `moveRight()`, `moveUp()` und `moveDown()` die die Figur in diese Richtungen bewegen sollen



- Füge der Klasse eine Methode hinzu, mit der die Figur gezeichnet werden kann (zeichne vorerst nur einen gelben Kreis und als Zusatz einen Kreis mit Öffnung in Bewegungsrichtung)

- Erstelle ein Pacman Objekt und rufe auf Klick der Pfeiltasten die `move` Methoden auf. Ruf die Methode zum zeichnen des Pacmans in `draw` auf und teste so deine Pacman Klasse

- Erstelle eine Klasse `Obstacle` welche die „Wände“ auf dem Spielfeld repräsentieren soll.
- Erstelle einen Konstruktor mit dem man die Position, Länge und Orientierung des Obstacles initialisieren kann. (Die Breite soll immer gleich sein)
- Erstelle eine Methode, die ein Obstacle als Rechteck mit rotem Rand zeichnet.
- Erstelle eine Methode `isTouched(float x, float y)` welche true zurück gibt wenn die x,y Koordinaten das Obstacle berühren oder sogar drüber sind und false wenn nicht

- Erzeuge 10 Obstacle Objekte mit zufälliger Position und Länge

- Prüfe bei jedem Pfeilklick des Benutzers nun ob du die Pacman Figur in dieses Richtung bewegen kannst oder ob sie da eine Wand berühren würde. Dann soll sie sich nicht bewegen. (bzw. wieder zurück bewegen) Ergänze dafür die Pacman Klasse mit einer Methode `Point getPosition()` und `float getSize();`
  - Tipp: wenn man nach oben fährt muss geprüft werden ob der obere Rand von Pacman das Rechteck berührt, wenn man nach rechts geht ob der rechte Rand berührt…

- Teste dein Programm & überprüfe deinen Code. Wo könntest du Variablen gebrauchen und wo eine Methode schreiben um Wiederholungen zu vermeiden?