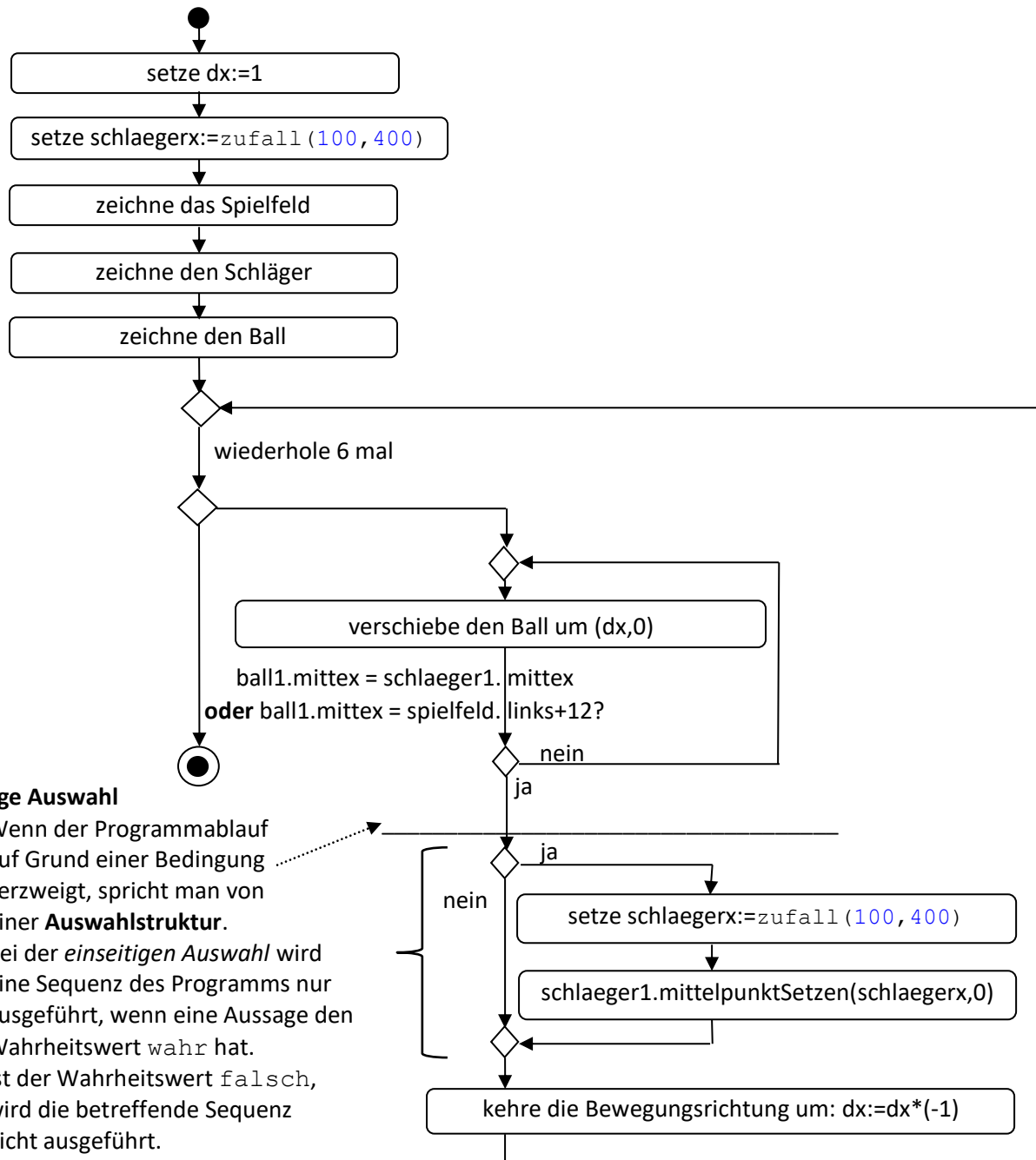




Die algorithmische Grundstruktur *Auswahl*

1. Im Programm *ballspiel3.eos* wird der Schläger nur einmal zu Beginn an eine zufällige Position gesetzt. Interessanter wäre es, wenn der Schläger immer wieder neu gesetzt würde, **wenn** der Ball links an der Bande abprallt.
 - Formuliere in dem Aktivitätsdiagramm unten die Bedingung dafür.



Einseitige Auswahl

- Wenn der Programmablauf auf Grund einer Bedingung verzweigt, spricht man von einer **Auswahlstruktur**. Bei der *einseitigen Auswahl* wird eine Sequenz des Programms nur ausgeführt, wenn eine Aussage den Wahrheitswert *wahr* hat. Ist der Wahrheitswert *falsch*, wird die betreffende Sequenz nicht ausgeführt.

Schreibweise in EOS:

```
wenn <Bedingung> dann
    <Anweisungen>
*wenn
```

- Optimierte die Programmversion *ballspiel3.eos*, indem der Schläger immer wieder neu gesetzt wird, wenn der Ball links an der Bande abprallt (*ballspiel4.eos*). (Vorlagedatei: *v06-ballspiel3.eos*)



2.6.1 Modellieren und Codieren von Algorithmen

Arbeitsblatt 07 Die algorithmische Grundstruktur *Auswahl*

2. Ein weiteres Beispiel für einen Zufallsversuch ist das Werfen einer Münze. Hier können zwei Ereignisse eintreten: „Kopf“ oder „Zahl“.
Das Entwickeln eines Programms, mit dem das Werfen einer Münze simuliert und Ergebnisse als Text („Kopf“ oder „Zahl“) ausgegeben werden, ist in EOS mit Hilfe der Klasse *Textfeld* möglich.

Um das Werfen einer Münze zu simulieren, wird eine Zufallszahl von 1 bis 2 erzeugt und in der Variablen *Wurf* gespeichert. Für die Zahl 1 soll der Text 'Kopf' angezeigt werden, für die Zahl 2 der Text 'Zahl'.

In die Variablen *WurfTextx* und *WurfTexty* werden die x- bzw. y-Koordinaten des Textfelds eingetragen.

- Ergänze das Aktivitätsdiagramm.

methode *wirfMuenze*

```
Wurf:=zufall(1,2)
```

```
wenn Wurf=1 dann
```

```
    Text_Wurf.zeileHinzufügen('Kopf')
```

```
*wenn
```

```
wenn Wurf=2 dann
```

```
    Text_Wurf.zeileHinzufügen('Zahl')
```

```
*wenn
```

ende

- Codiere die Aufgabenstellung in EOS.
Beachte die Objektdiagramme!
Speichere das Programm als *muenze1.eos*.

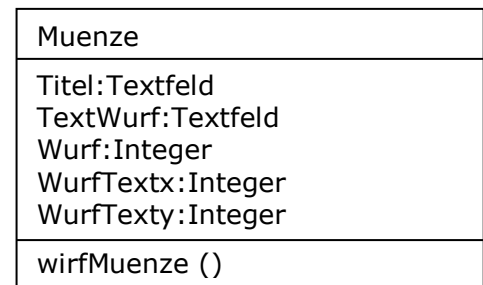
<u>Muenze1</u> :MUENZE
WurfTextx=-400 WurfTexty=130

<u>Titel</u> :TEXTFELD
Links=-400 Oben=180 Randfarbe=weiß Zeile1='Wurf einer Münze:'

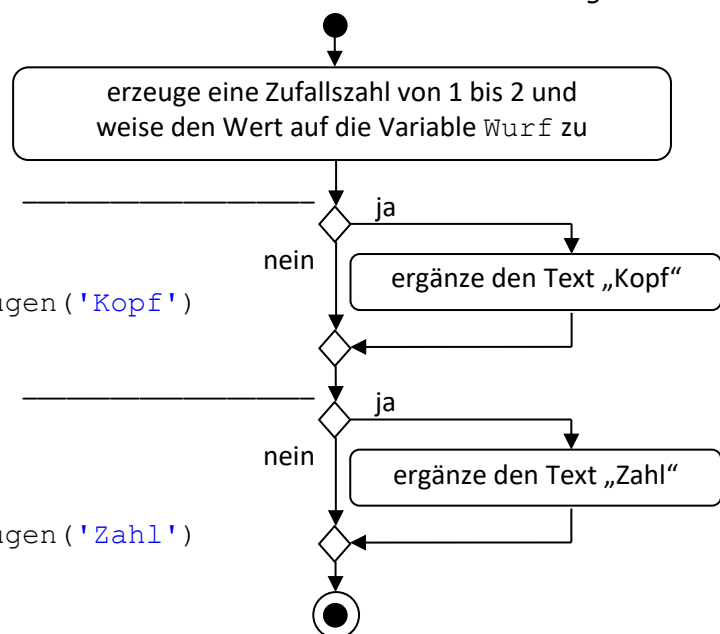
<u>f</u> :FENSTER
Links=24 Oben=88 Breite=850 Höhe=400 Hintergrundfarbe=weiß Gitter=aus

<u>TextWurf</u> :TEXTFELD
Links=WurfTextx Oben=WurfTexty groesseAutomatischAnpassen=wahr

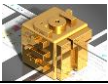
- Teste das Programm, indem du es mehrmals ausführst und die Ergebnisse kontrollierst.
Dabei müssen die Ereignisse in einer zufälligen Abfolge eintreten.



Klassendiagramm



Aktivitätsdiagramm *wirfMuenze*



2.6.1 Modellieren und Codieren von Algorithmen

Arbeitsblatt 07 Die algorithmische Grundstruktur *Auswahl*

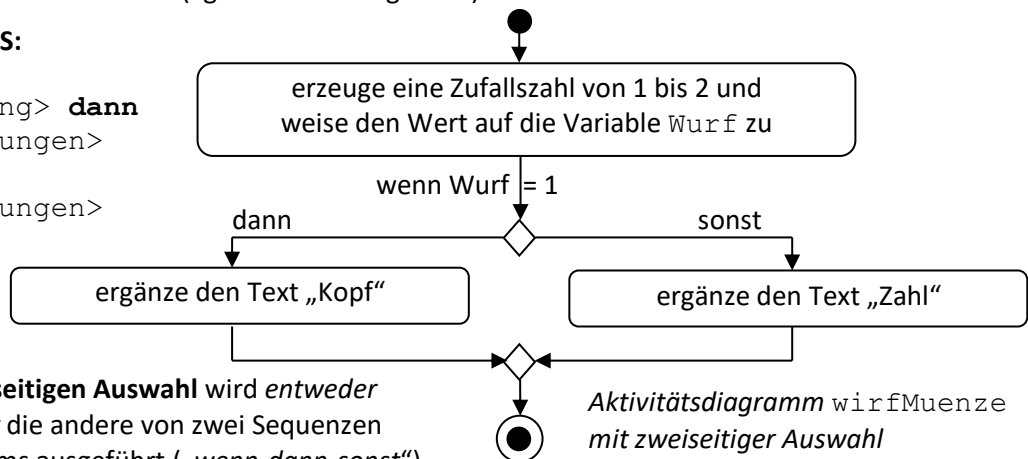
Zweiseitige Auswahl

Da hier genau zwei Möglichkeiten bestehen ('Kopf' oder 'Zahl'), kann die Auswahlstruktur auch vereinfacht werden: Wenn das Ergebnis nicht 'Kopf' ist, dann kann es ja nur 'Zahl' sein bzw.:

„Wenn 1 dann 'Kopf' sonst 'Zahl'.“ (vgl. Aktivitätsdiagramm)

Schreibweise in EOS:

```
...
wenn <Bedingung> dann
    <Anweisungen>
sonst
    <Anweisungen>
*wenn
...
```

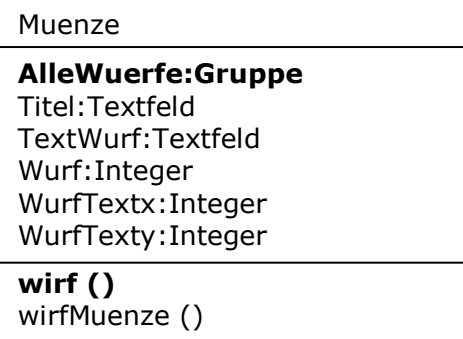


- Bei der **zweiseitigen Auswahl** wird *entweder* die eine *oder* die andere von zwei Sequenzen des Programms ausgeführt („wenn-dann-sonst“).

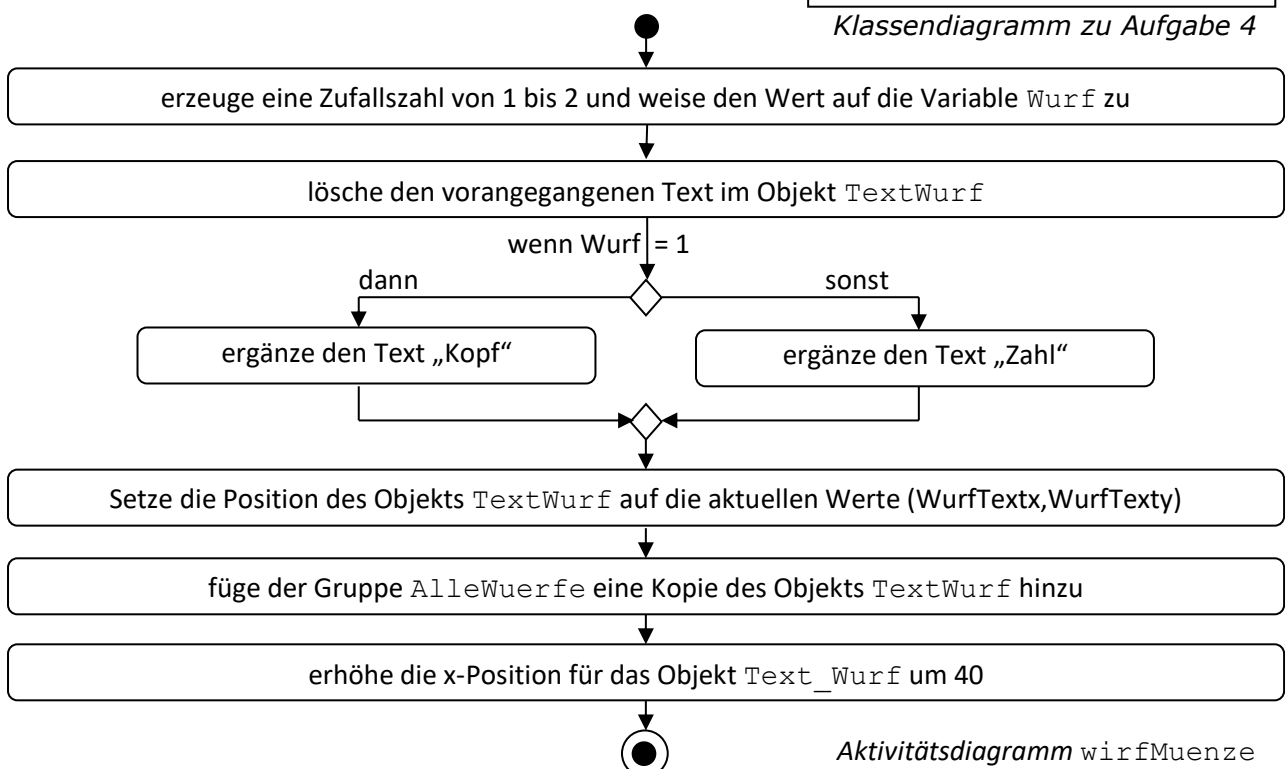
3. Ändere dein Programm *muenze1.eos* unter Verwendung der zweiseitigen Auswahl (*muenze2.eos*). (Vorlagedatei: v07-muenze1.eos)

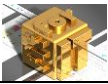
4. Ergänze dein EOS-Programm: Es sollen 20 Würfe ausgeführt werden, so dass eine Zeile des Fensters angefüllt wird. Erstelle dafür die Methode *wirf()* (*muenze3.eos*).

Hinweis: Das Textfeld *TextWurf* kann wiederholt verwendet werden, wenn der jeweilige Wurf nicht direkt angezeigt wird, sondern in eine Gruppe, z. B. *AlleWuerfe*, kopiert wird:



Klassendiagramm zu Aufgabe 4



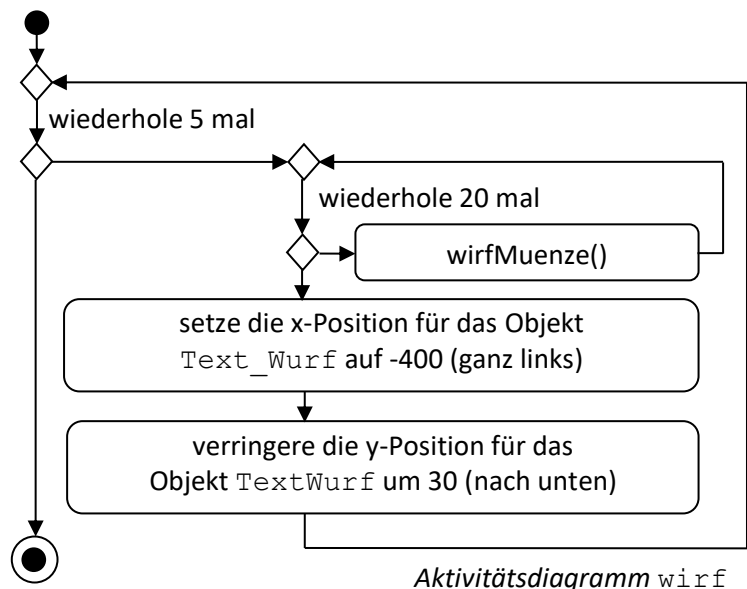


2.6.1 Modellieren und Codieren von Algorithmen

Arbeitsblatt 07 Die algorithmische Grundstruktur *Auswahl*

5. Die Münze soll 100-mal geworfen werden, wofür 5 Zeilen à 20 Würfeln benötigt werden. Ergänze das EOS-Programm nach dem Aktivitätsdiagramm zur Methode `wirf()` (muenze4.eos). (Vorlagedatei: v08-muenze3.eos)

Hinweis: Hier sind mehrere Zeilen erforderlich, um die Ergebnisse anzeigen zu können. Das kann z. B. mit einer geschachtelten Wiederholung umgesetzt werden.



6. Das vorige Programm soll so erweitert werden, dass die Summen der Ereignisse mitgezählt und am Ende in Textfeldern angezeigt werden (muenze5.eos). Dafür werden zwei Zählvariablen benötigt, z. B. `SummeKopf` und `SummeZahl`. Zur Anzeige der Ergebnisse soll eine neue Methode eingeführt werden, z. B. `gibErgebnisAus()`. Speichere das Programm unter `muenze05.eos`.

Zeichenfenster

Zufallsversuch mit einer Münze:

Kopf Kopf Kopf Kopf Zahl Zahl Zahl Zahl Zahl Kopf Zahl Kopf Kopf Zahl Zahl Kopf Zahl Kopf Kopf Kopf
Kopf Kopf Kopf Kopf Kopf Kopf Zahl Kopf Zahl Zahl Zahl Kopf Kopf Zahl Zahl Zahl Kopf Zahl Zahl Zahl
Kopf Kopf Zahl Zahl Kopf Kopf Zahl Zahl Zahl Zahl Kopf Zahl Kopf Kopf Zahl Kopf Zahl Kopf Kopf Kopf
Zahl Kopf Zahl Zahl Zahl Kopf Kopf Zahl Kopf Kopf Kopf Zahl Kopf Kopf Kopf Zahl Kopf Kopf Zahl Kopf
Kopf Kopf Kopf Zahl Kopf Zahl Zahl Zahl Zahl Kopf Kopf Zahl Zahl Kopf Kopf Kopf Kopf Kopf Kopf Kopf

Die Anzahl der Würfe war: 100
Die Anzahl der Ereignisse Kopf beträgt: 57
Die Anzahl der Ereignisse Zahl beträgt: 43

Muenze
AlleWuerfe:Gruppe
Titel:Textfeld
TextWurf:Textfeld
TextAnzahl:Textfeld
TextSummeKopf:Textfeld
TextSummeZahl:Textfeld
Wurf:Integer
SummeKopf:Integer
SummeZahl:Integer
WurfTextx:Integer
WurfTexty:Integer
wirf ()
wirfMuenze ()
gibErgebnisAus ()

Klassendiagramm zu Aufgabe 6