

# Informationen zum Wahlpflichtbereich II

am Konrad-Adenauer-Gymnasium

# Wahlpflichtbereich II

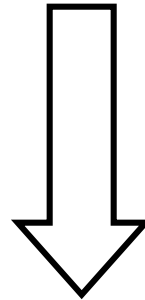
## „Differenzierung“

- zusätzliches Fächerangebot
- Unterricht in Kursen statt Klassen
- durchgehend in den Jahrgangsstufen 9/10
- man hat die **Wahl** zwischen mehreren Fächern
- dieses muss 2 Jahre belegt werden (**Pflicht**)
- größere Freiheiten in der inhaltlichen Gestaltung



## Grundgedanke:

Ich entscheide mich für ein Fach, das ich  
belegen möchte,  
weil ich es gut kann und/oder weil es mir  
Spaß macht.



Hinführung zur Autonomie in der Oberstufe:  
Wahlen zur Sekundarstufe II  
Kurssystem

# Fächerangebot an unserer Schule in der Differenzierung:

■ Informatik

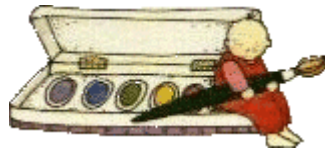


■ Biologie/Chemie



■ Spanisch

■ Kunst/Design



für **alle** Fächer gilt:

Unterricht in Kursen und Fachräumen

**Klassen 9 und 10:** 3 Stunde pro Woche

(i. d. R.: 1 Doppelstunde + 1 Einzelstunde)

**Ausnahme: Spanisch** 4 Stunden pro Woche

WPfII-Fächer gehören zur **Fächergruppe II**

also: kein „**Hauptfach**“ bei  
Versetzungsentscheidungen

d.h. kein Ausgleich für M, D, E, F, L

2 Klassenarbeiten pro Halbjahr

# Wahlverfahren



Durchführung der Wahl spätestens bis

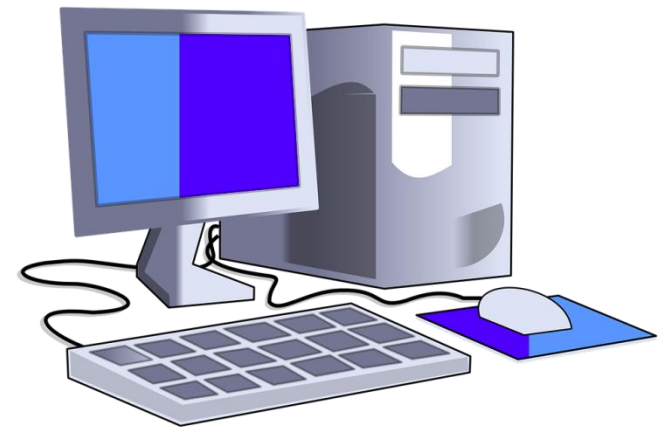
**02. Juni 2026**

- Angabe eines **Erstwunsches**
- zwingend: Angabe von zwei weiteren Wünschen

- Entscheidung über Wahlen und Kurszuteilung: Schule
- Wahl über Homepage
- bei zu vielen Interessenten für ein Fach **entscheidet das Los**
- Zweitwunsch kann fast immer erfüllt werden

# Informatik in der Mittelstufe

- ◆ auf Klasse 6 aufbauend
- ◆ Vermittlung von Medienkompetenz
- ◆ Theoretische Informatik
- ◆ erste Programmiererfahrungen
- ◆ variable thematische Ausgestaltung
- ◆ selbstständiges Erarbeiten von Themen



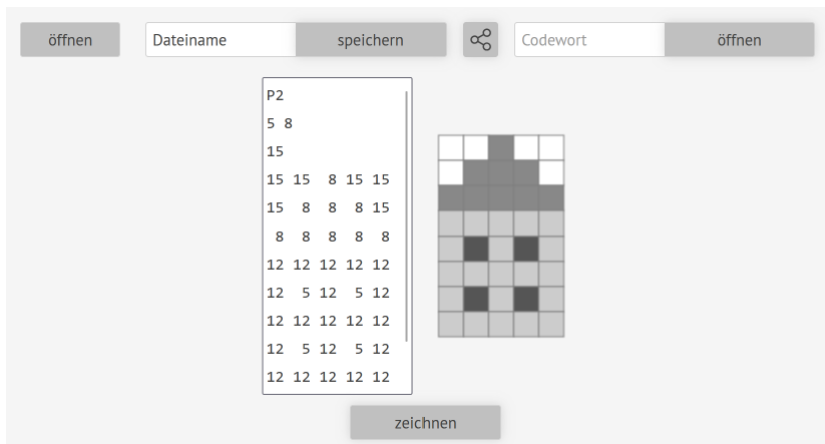
# Mein digitaler Fußabdruck

- ◆ Prinzipien des Datenschutzes
- ◆ personenbezogene Daten / Metadaten
- ◆ Umgang mit fremden und eigenen Daten
- ◆ Fallbeispiele aus dem Alltag



# Daten und ihre Codierung

- ◆ rechnen in Stellenwertsystemen
- ◆ Codierung in Binärcode
- ◆ Arten der Informationscodierung
- ◆ Pixelgrafiken



# HTML und CSS

- ◆ Grundkenntnisse im Webdesign
- ◆ Erstellung eigener Webseiten
- ◆ Projektarbeit
- ◆ evtl. Erweiterung um PHP



```
h1 { color: white;
background: orange;
border: 1px solid black;
padding: 0 0 0 0;
font-weight: bold;
}
/* begin: seaside-theme */

body {
background-color:white;
color:black;
font-family:Arial,sans-serif;
margin: 0 4px 0 0;
border: 12px solid;
}
```

# Tabellenkalkulation

- ◆ Erweiterung des Grundwissens
- ◆ Tabellen erstellen
- ◆ Formatierungen gezielt einsetzen
- ◆ Nutzen von Zellfunktionen
- ◆ Diagramme erstellen / auswerten
- ◆ Bedingungen / Verweise

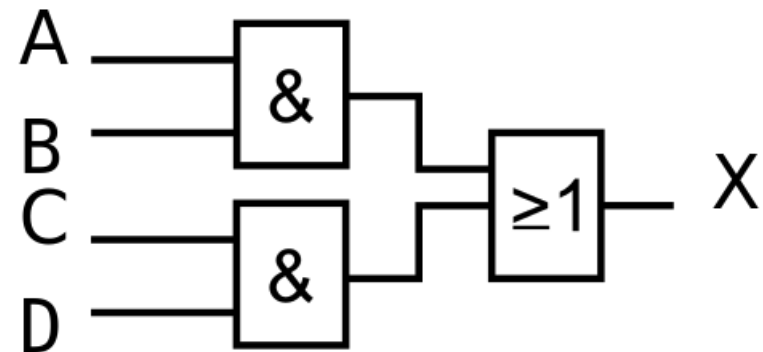
	A	B	C	D
1	<b>Date</b>	<b>Income</b>	<b>Expenses</b>	<b>Profit</b>
2	2005-12-17	235 €	128 €	107 €
3	2005-12-18	311 €	124 €	187 €
4	2005-12-19	457 €	466 €	-9 €
5	2005-12-20	232 €	132 €	100 €
6	2005-12-21	122 €	134 €	-12 €
7	2005-12-22	128 €	223 €	-95 €
8	2005-12-23	432 €	218 €	214 €
9	2005-12-24	256 €	121 €	135 €
10		<b>2.173 €</b>	<b>1.546 €</b>	<b>627 €</b>
11				
12	<b>Avg. Profit</b>	<b>=AVERAGE(D2:D9)</b>		

# Technische Informatik: Schaltalgebra

- ◆ Mathematik / Logik
- ◆ Modellierung von Rechnerarchitektur
- ◆ Logische Gatter / Wahrheitstabellen
- ◆ Optimierung von Schaltnetzen
- ◆ KV-Diagramme

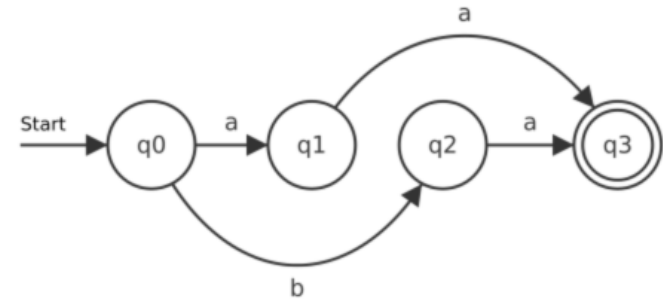
A	B	$(A \rightarrow B)$
0	0	1
0	1	1
1	0	0
1	1	1

	A	$\bar{A}$	B	$\bar{B}$
A	1	0	1	0
$\bar{A}$	0	1	1	0
B	1	0	1	0
$\bar{B}$	0	1	1	0

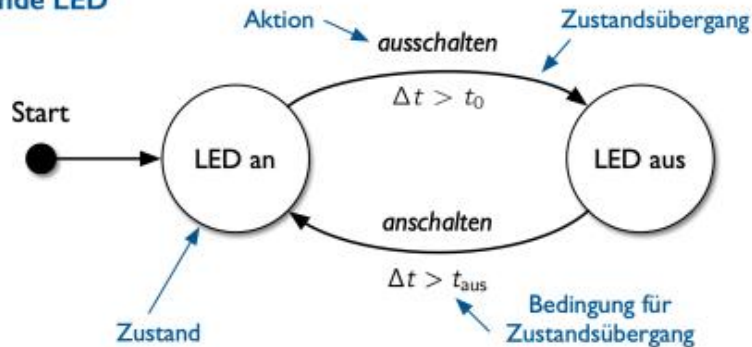


# Endliche Automaten

- ◆ theoretische Informatik
- ◆ Automaten als Modelle für Code
- ◆ Zustandsübergangsdiagramme
- ◆ Informatik mit “Stift und Papier”



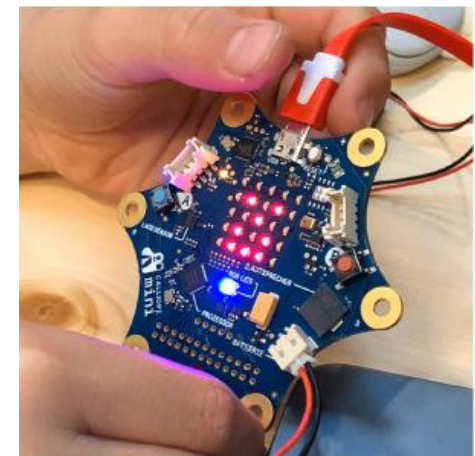
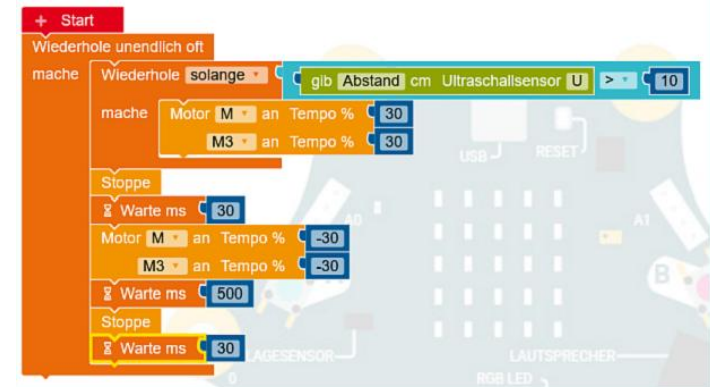
## Blinkende LED



- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{a, b\}$
- $\delta$  wird dargestellt durch die Pfeile, die von einem Zustand zum nächsten führen.
- $E = \{q_3\}$
- $s = q_0$

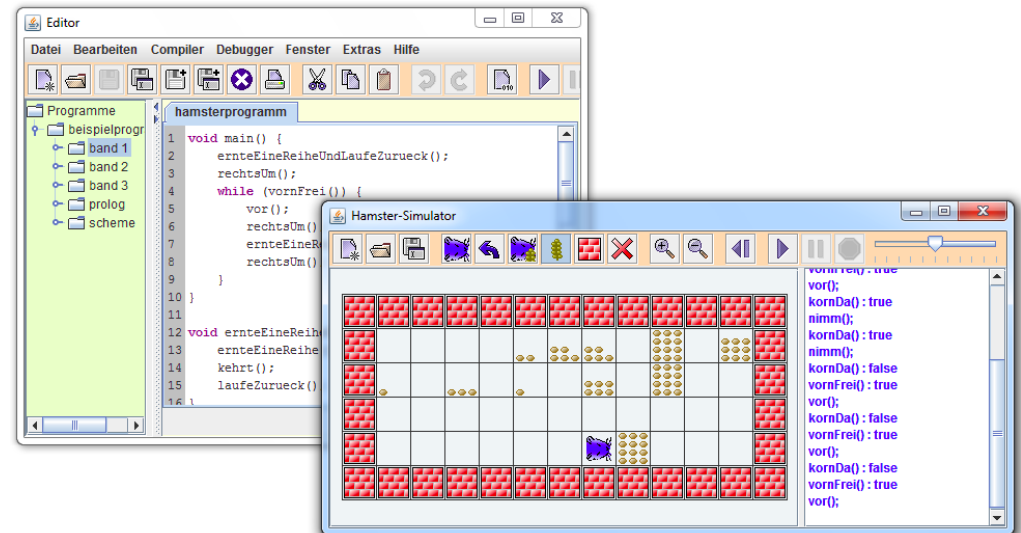
# Roboterprogrammierung

- ◆ Aufbau von Steuerungen
- ◆ Aktoren / Sensoren
- ◆ Calliope mit Callibot
- ◆ Programmierung auf Grundlage der Automatentheorie



# Java Hamster

- ◆ Imperative / Funktionale Programmierung
- ◆ Lösen einfacher bis mittelschwerer Programmieraufgaben
- ◆ gegebene Implementierungen untersuchen
- ◆ Code kommentieren
- ◆ Syntax/Strukturen verstehen



# optional: App-Inventor, Python

- ◆ stärkerer Anwendungsbezug
- ◆ Apps für eigene Handys
- ◆ Graphische Oberflächen (GUI)
- ◆ Programmierung einfacher Spiele
- ◆ einf. Anwendungen (Taschenrechner)

