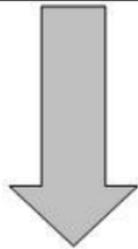




Benutzer



Anfrage an Webdatenbank

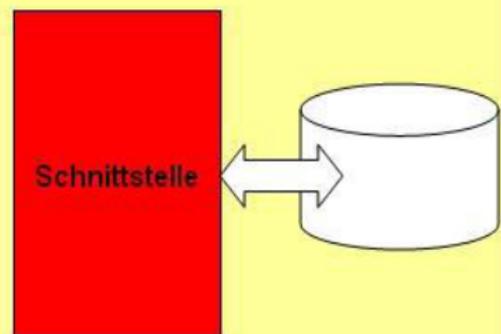
Aufbereitete Daten

# Informatiktag 2017 Wuppertal

**Arno Pasternak**

Fritz-Steinhoff-Gesamtschule Hagen  
TU Dortmund

## Webdatenbank



w.openpr.de/images/articles/b/5/b5366bbfeeee995471116185a41744bc\_g.jpg)

## Inhaltsverzeichnis

1

Didaktische Vorbemerkungen

- Informatisches Denken
- Kontextorientierung
- Das Konzept der Roten Fäden

2

Vorkenntnisse für den Kontext Webdatenbanken

- Programmiersprachen
- Datenbanken
- Netze
- HTML
- Betriebssysteme

3

Unterrichtsverlauf

- Netze
- HTML
- SQLite
- Webdatenbanken

Jeannette Wing 2006



**Computational thinking**  
is a fundamental skill for everyone,  
not just for computer scientists.

## C.T.: Abstraction and Automation [Wing, 2008]

### Two A's to C.T. Combined

- Computing is the **automation** of our **abstractions**
  - They give us the audacity and ability to scale.
- Computational thinking
  - choosing the right abstractions, etc.
  - choosing the right "computer" for the task

## Computational thinking is a fundamental skill for everyone, ...

### Computational thinking ...

- ... is thinking **recursively**
- ... **parallel processing**
- ... **interpreting** code as data and data as code.
- ...

### Computational thinking ...

- ... is **using abstraction and decomposition** when attacking a large complex task ....
- ... is choosing an **appropriate representation** for a problem ...
- ...

[Wing, 2006, S.33]

## Warum Kontextorientierung?

### Ziel

Wir wollen ja auch und gerade Informatik für **Nichtinformatiker** betreiben!

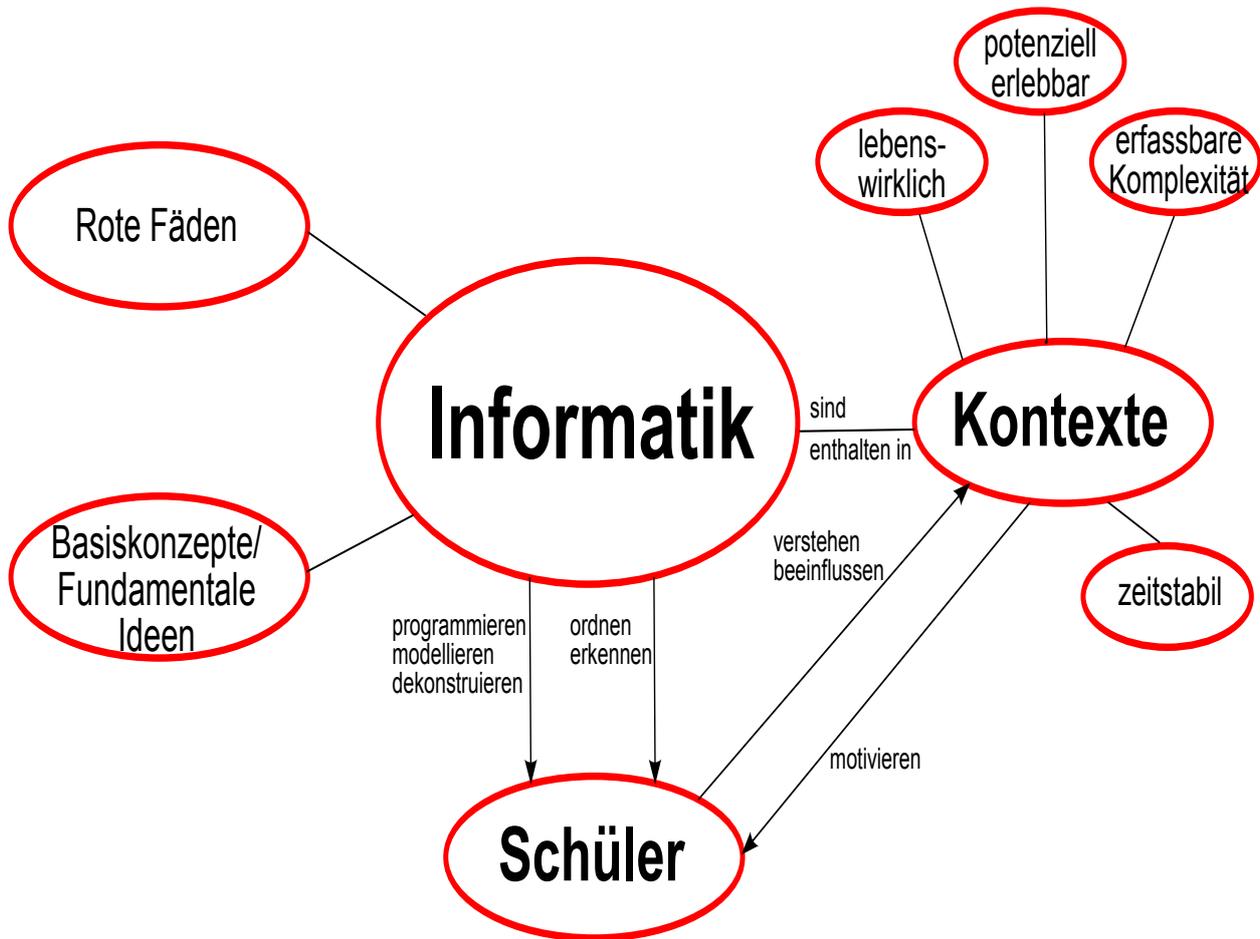
Es ist sinnvoll, nachvollziehbare Zusammenhänge aus der **Lebenswirklichkeit** aus Sicht des Schülers als Ausgangspunkt zu nehmen, also:

**kontextorientierte Informatik.**

### Das ist nichts Neues!

- Comenius [Lion, 1883]
- Wagenschein [Wagenschein, 1968], [Wagenschein, 1971]
- Bruner [Bruner, 1960]
- Aebli [Aebli, 1976], [Aebli, 1998]

...

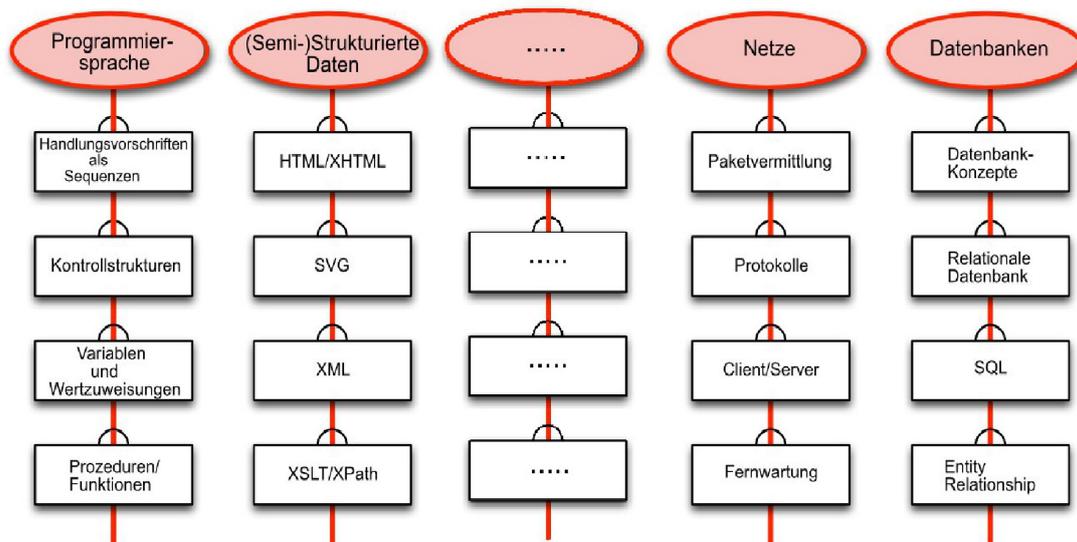


## Rote Fäden

### Definition **Roter Faden**

Ein **Roter Faden** ist eine Ansammlung unterrichtlicher Gegenstände, die den folgenden Kriterien genügt:

- Die unterrichtlichen Gegenstände lassen sich einem **gemeinsamen fachinhaltlichen** (strukturellen oder thematischen) **Zusammenhang** zuordnen. (Kohärenz)
- Der gemeinsame fachinhaltliche Zusammenhang ist zu jedem Zeitpunkt aus **Sicht der Schülerinnen und Schüler** erkennbar und nachvollziehbar. (Transparenz)
- Der gemeinsame fachinhaltliche Zusammenhang wird im Verlauf des Unterrichts **aus verschiedenen Blickwinkeln** oder in verschiedenen Kontexten dargestellt. (Varianz)
- Die Anordnung der unterrichtlichen Gegenstände durchzieht **mehrere Unterrichtseinheiten**. (Redundanz)

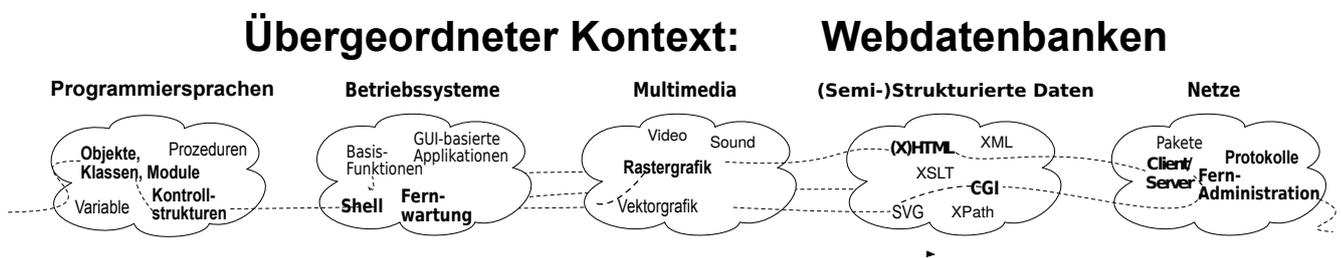


Beispielhafte **Rote Fäden** im Unterricht der Sekundarstufe I

Wichtig: Die Auswahl der (endlich vielen) Fäden obliegt dem Fachkollegen bei der Erstellung des Curriculums (siehe auch: [Pasternak and Vahrenhold, 2009]).

## Rote Fäden

Im Kontext werden die **Roten Fäden** miteinander verwoben.



*(Für den Kontext Webdatenbanken werden gegebenenfalls aus dem früheren Unterricht fehlende Einzelheiten im Kontext eingeführt.)*

siehe auch: [Pasternak and Vahrenhold, 2010]

## Notwendig

### „Übliche“ Kenntnisse über

- Variablen
- Prozedurkonzept
- Objekte, Klassen, Module
- Kontrollstrukturen
- ...

## Ergänzend

### Kenntnisse über

- Dateiverwaltung
- Prozesssteuerung
- ...

Es muss **dieselbe Programmiersprache** verwendet werden wie im sonst **üblichen Unterricht** (z.B. Java/Groovy).

## Notwendig

### „Übliche“ Kenntnisse über

- Relationale Datenbanken
- SQL
- speziell: Select-Anweisung
- ...

## Evtl. Ergänzend

### Kenntnisse über

- Eingabe per Script
- Ausgabe als HTML-Fragment
- ...

Es muss **dieselbe Datenbank** verwendet werden wie im sonst **üblichen Unterricht** (z.B. SQLite).

## Notwendig

### „Übliche“ Kenntnisse über

- Aufbau des Internets
- Client-Server-Prinzip
- Remote-Zugriff
- ...

## Evtl. Ergänzend

### Kenntnisse über

- Rolle von IP und Port
- http-Protokoll
- Aufruf von Servern bzw. Daemonen
- ...

Aktivitäten wie **Starten eines Servers** (z.B. thttpd) wurden auch **praktisch** durchgeführt bzw. können durchgeführt werden.

## Notwendig

### „Übliche“ Kenntnisse über

- Aufbau einer HTML-Seite (Head, Body)
- Einfache Dokumentenstrukturmittel (Überschriften, ...)
- Unterschied: Aufruf lokal/remote
- ...

## (Evtl.) Ergänzend

### Kenntnisse über

- Formulare
- Statische und dynamische Seiten
- ...

**CSS, Javascript, Ajax** etc. sind in diesem Kontext **nicht nötig**, sie verwirren hier nur.

## Notwendig

„Übliche“ Kenntnisse über

- Arbeit mit der Konsole
- Fernkonsole
- ...

## (Evtl.) Ergänzend

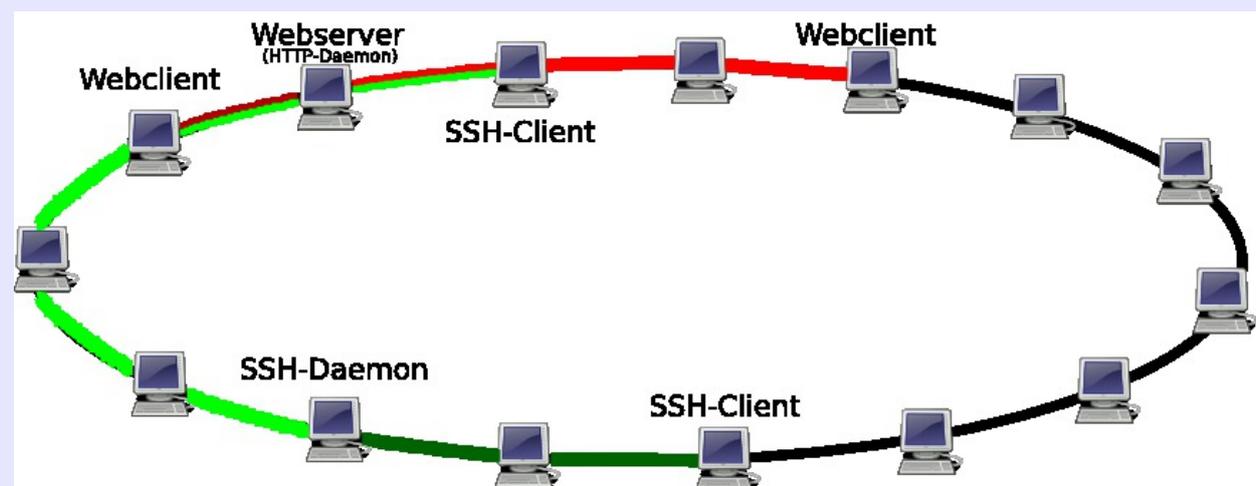
Kenntnisse über

- Parallele Prozesse
- Umgebungsvariablen
- ...

Es zeigt sich auch hier, dass es **sinnvoll** ist, **direkt auf der Konsole arbeiten** zu können.

# Netze

## Der Ring als Beispiel für eine Netzstruktur



Die Arbeitsweise eines Netz ist **nicht trivial** und **ohne Unterricht** den Schülern **nicht bekannt**.

Der (vereinfachte) **Ring** ist eine gute Erklärungsgrundlage.

Die **Paketvermittlung** kann auch sehr gut gespielt werden.

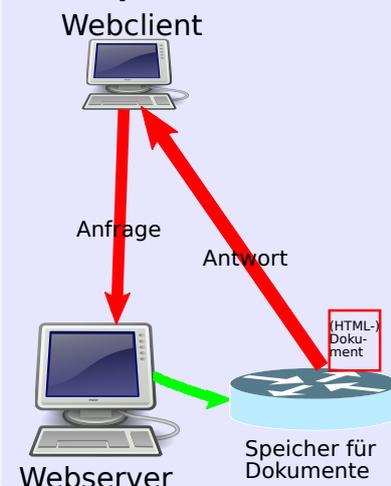
## Erste HTML-Seite

```
<html>

<head>
<title>Unsere erste Seite</title>
</head>

<body>
<h1>Unsere tolle erste Seite</h1>
Die Sch&uuml;ler waren faul.
</body>
</html>
```

## Statische Webseite

Arbeitsweise des  
http - Protokolls

Die **Rolle des Browsers** und der Unterschied zwischen Laden einer Datei auf dem **lokalen Computer** und per **http-Protokoll** muss klar werden. (Das ist nicht selbstverständlich.)

**Start** des (eigenen) http-Servers: `tthttpd -d ./ -p 2000`

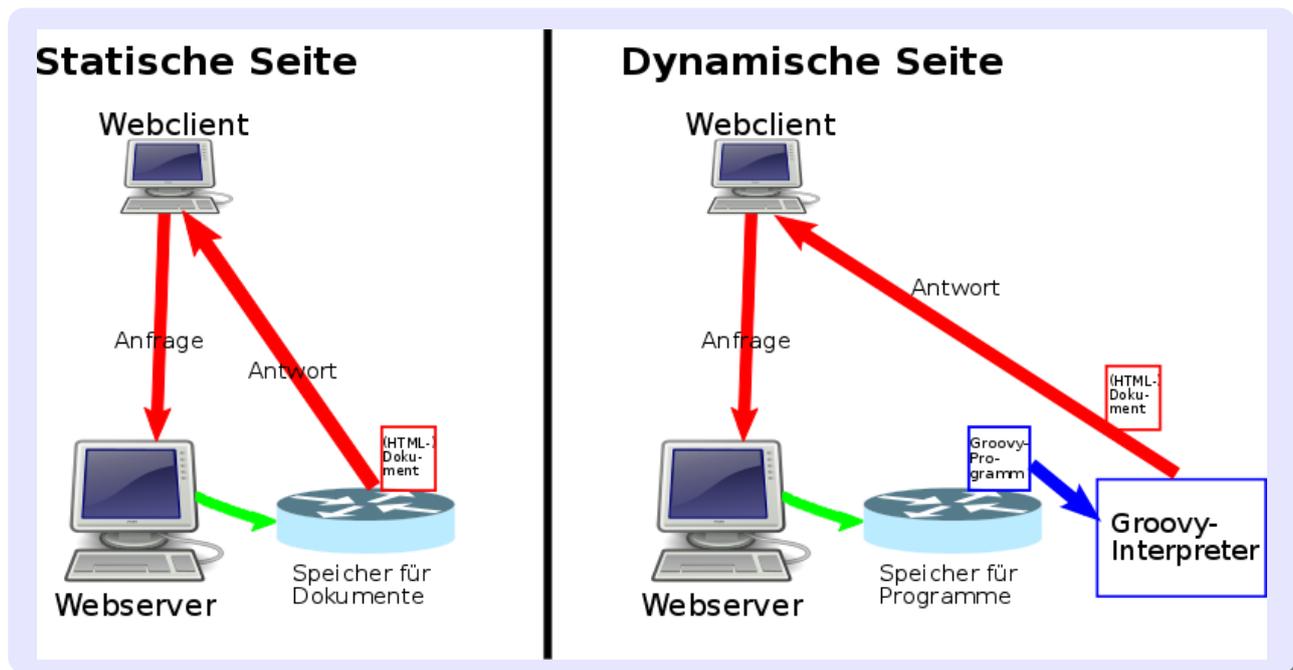
## HTML dynamisch

## Erste dynamische HTML-Seite

```
// PROGRAMM moin0 (in Groovy);
println("Content-type: text/html");
println();
println("<html>");
println("<head>");
println("<title>Guten Tag</title>");
println("</head>");
println("<body>");
println("Hallo allerseits!");
println("</body>");
println("</html>");
// ENDE
```

Statt dieses **Groovy-Programmes** hätte auch ein Programm in **Java** mit derselben Funktionalität geschrieben werden können.

**Start** des http-Servers **jetzt**: `tthttpd -d ./ -p 2000 -c /cgi/*`



## Wrapper

### Probleme

- Der **Webserver** kann nur **Maschinenprogramme** direkt ausführen!
- Der Webserver kann allerdings auch **Bash-Scriptprogramme** starten.
- **Groovy** bzw. **Java** benötigen eine **bestimmte** Systemumgebung.

### Lösung: Wrapper

```
#!/bin/sh
```

```
export JAVA_HOME=/usr/lib64/java  
cd /home/arno/dokument/...../wuppertal_2017/web/cgi/  
groovy ./moin0a.groovy
```

# AUSPROBIEREN!!

(Bash-) Terminal-Sitzung:

bash

## HTML dynamisch

### Die erste echte dynamische HTML-Seite

```
// PROGRAMM moin1
int rotwert
String rot

rotwert = Math.random()*256
println rotwert
rot = Integer.toHexString(rotwert)
println rot

println ("Content-type: text/html");
println ();
println ("<html>");
println ("<head>");
println ("<title>Guten Tag</title>");
println ("</head>");
println ("<body bgcolor=#"+rot+"0000>");
println ("Hallo allerseits!");
println ("</body>");
println ("</html>");
// ENDE
```

# AUSPROBIEREN!!

(Bash-) Terminal-Sitzung:

bash

## HTML dynamisch mit 'Fernbedienung'

```
// PROGRAMM farbseite1
String parameterzeile
String farbe
int stelle

parameterzeile = System.getenv("QUERY_STRING")
println ("Parameterzeile: "+parameterzeile)

farbe=""; stelle=6;
while (stelle < parameterzeile.length()){
    buchstabe = parameterzeile.charAt(stelle);
    farbe = farbe + buchstabe;
    stelle = stelle + 1;
}
println ("Farbe ist: "+farbe)

println ("Content-type: text/html");
println ();
println("<html>");
println("<head>");
println("<title>Guten Tag</title>");
println("</head>");
println("<body bgcolor=#"+farbe+">");
println("Hallo allerseits!");
println("</body>");
println("</html>");
// ENDE
```

# AUSPROBIEREN!!

(Bash-) Terminal-Sitzung:

bash

## HTML Formular

```

<html>
<head>
<title>Seite mit CGI-Groovy-Aufruf</title>
</head>
<body>
<h1>Farbseite – Groovy-Version</h1>
<h2>Es kann der Farbton als hexadezimale Zahl
mit sechs Ziffern eingegeben werden.</h2>
<hr>

<form method="get" action="/cgi/starten_farbseite1.sh">
  RGB-Farbe: <input name="farbe" type="text "
              "maxlength="6"><br>
  <input type="submit" value="Anfragen">
</form>

</body>
</html>

```

# AUSPROBIEREN!!

(Bash-) Terminal-Sitzung:

```
bash
```

## Datenbanken mit SQLite

### DB im Unterricht der FSG

- **Datenbanken mit SQLite**  
werden in zwei Unterrichtseinheiten in Jg. 11 und Jg. 12 unterrichtet.
- **Webdatenbanken**  
folgen als Wiederholung und Ergänzung in Jg. 13.

### SQLite austesten

(Bash-) Terminal-Sitzung:

```
bash
```

## Ein Skript für SQLite

```
.open adressen1a.db
.output antwort.sql
.header on
.mode html

select * from adressentabelle where stadt='Hagen';
```

# Datenbanken mit SQLite

## ... und die Antwort von SQLite

```
<TR><TH>Nr.</TH>
<TH>Name</TH>
<TH>Vorname</TH>
<TH>Strasse</TH>
<TH>Plz</TH>
<TH>Stadt</TH>
</TR>
<TR><TD>2</TD>
<TD>Mader</TD>
<TD>Gerd</TD>
<TD>Bügelweg 5</TD>
<TD>58099</TD>
<TD>Hagen</TD>
</TR>
<TR><TD>3</TD>
<TD>Abel</TD>
<TD>Herrmann</TD>
<TD>Mühlenstr.</TD>
<TD>58097</TD>
<TD>Hagen</TD>
</TR>
...
```

## Erforderliche Schritte

```
//Programm Datenbank2b
String stadt

stadt = parameterzeile_auslesen()
anfrage_schreiben(stadt)
sqlite_aufruf_schreiben()
sqlite_aufrufen()
dokument_ausgeben()
```

## Schritt 1

```
String parameterzeile_auslesen() {
    String parameterzeile, stadt
    int stelle
    char buchstabe

    parameterzeile = System.getenv("QUERY_STRING")
    println ("Parameterzeile: "+parameterzeile)

    stadt=""
    stelle=6

    //Die Parameterzeile wird 'zerlegt':
    while (stelle < parameterzeile.length()) {
        buchstabe = parameterzeile.charAt(stelle)
        stadt = stadt + buchstabe
        stelle = stelle + 1
    }
    return stadt
}
```

## Schritt 2

```
void anfrage_schreiben(String stadt) {
    //Die SQL-Anfrage wird in eine Textdatei geschrieben:
    FileWriter anfragedatei
    anfragedatei = new FileWriter("./anfrage.sql")

    anfragedatei.write(".open adressen1a.db\n")
    anfragedatei.write(".output antwort.sql\n")
    anfragedatei.write(".header on\n")
    anfragedatei.write(".mode html\n")
    anfragedatei.write("select * from adressentabelle\n")
    anfragedatei.write("where stadt='"+stadt+"' ;\n")
    anfragedatei.close()
    println ("Anfragedatei fertig geschrieben")
}
```

## Schritt 3

```
void sqlite_aufruf_schreiben() {
    //Der SQLite-Aufruf wird in eine Datei geschrieben:
    FileWriter aufrufdatei
    aufrufdatei = new FileWriter("aufruf.sql")

    aufrufdatei.write("rm fertig.txt\n")
    aufrufdatei.write("/usr/bin/sqlite3 < anfrage.sql\n")
    aufrufdatei.write("touch fertig.txt\n")
    aufrufdatei.write("sleep 1\n")
    aufrufdatei.write("rm fertig.txt\n")
    aufrufdatei.close()
    println ("Aufrufdatei fertig geschrieben")
}
```

## Schritt 4

```

void sqlite_aufrufen() {
    //Es wird der SQLite-Aufruf durchgeführt:
    File datei

    new ProcessBuilder(
        "/bin/sh", "aufruf.sql" ).start()
    println ("SQLITE wurde ausgeführt")

    //Es wird gewartet,
    //bis die Aufrufdatei ausgeführt wurde:
    datei = new File("./fertig.txt")
    while (datei.exists()==false) {
        println("ich warte")
    }
    println ("Genug gewartet!")
}

```

## Schritt 5

```

void dokument_ausgeben() {
    FileReader antwortdatei
    def zeichen = new char[1]
    int ende

    //Es wird das Ausgabedokument in die Ausgabe geschrieben:
    println("Content-type: text/html")
    println()
    println("<html>")
    println("<head>")
    println("<title>Datenbankbeispiel 2</title>")
    println("</head>")
    println("<body>")
    println("<h1>Datenbank Beispiel</h1>")

    println("<table border='1'>")
    //Die Antworttabelle von SQLite wird in das Dokument eingefügt:
    antwortdatei = new FileReader("./antwort.sql")
    ende = 0
    while (ende != -1) {
        ende = antwortdatei.read(zeichen)
        print((char)zeichen[0])
    }
    //Das Restdokument wird geschrieben:
    println("</table>")

    println("</body>")
    println("</html>")
}

```

## Formular-HTML-Seite

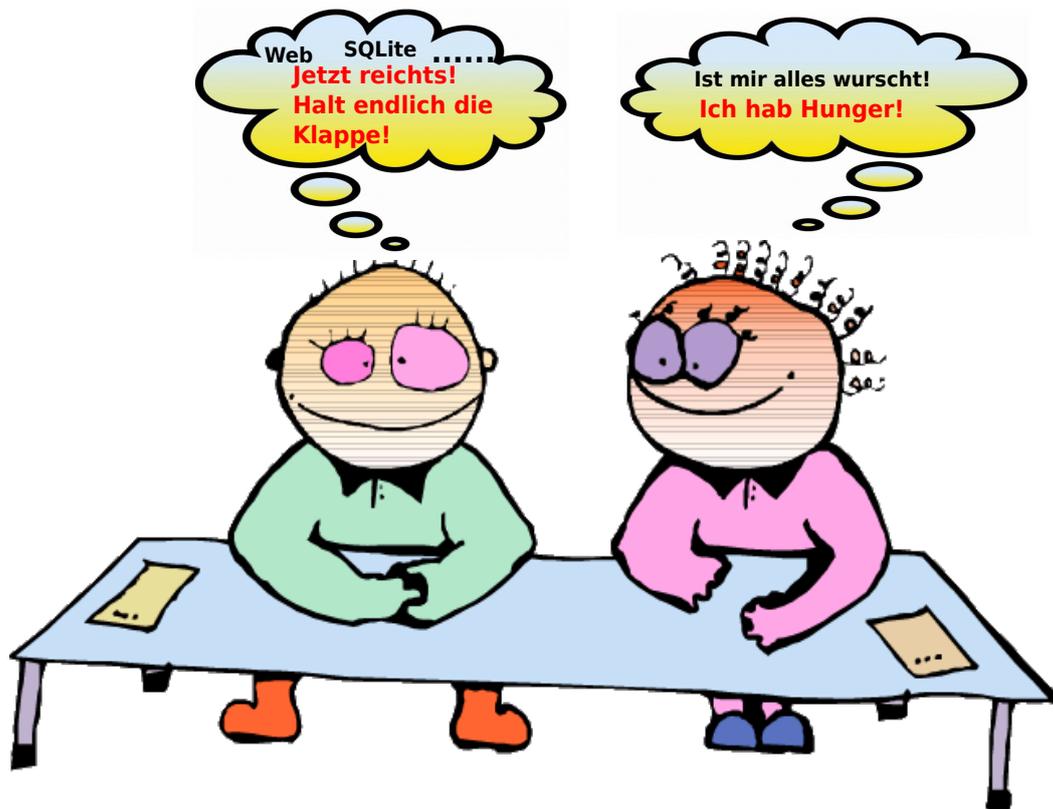
```
<html>
<head>
<title>Seite mit Groovy-Aufruf unserer Datenbank</title>
</head>
<body>
<h1>Datenbank Beispiel 2gb</h1>
<h2>Es wird der Inhalt der Datenbank
    f&uuml;r eine Stadt ausgegeben.</h2>

<form method="get"
    action="/cgi/starten_datenkbank2gb.sh">
  Stadt: <input name="stadt" type="text"
    size="15" maxlength="30"><br>
  <input type="submit" value="Anfragen">
</form>
</body>
</html>
```

# AUSPROBIEREN!!

(Bash-) Terminal-Sitzung:

```
bash
```



Bildquellen:  
[http://www.sks-tierpark.bildung-lsa.de/1\\_unsere\\_schule/schueler/cliparts/zwei\\_tisch.gif](http://www.sks-tierpark.bildung-lsa.de/1_unsere_schule/schueler/cliparts/zwei_tisch.gif), 23.3.2017  
<http://www.publicdomainpictures.net/pictures/70000/nahled/thought-cloud.jpg>, 23.3.2017



Aebli, H. (1976).

*Grundformen des Lehrens: Eine allgemeine Didaktik auf Kognitionspsychologischer Grundlage.*  
 Pädagogik - Einführung. Klett, 9. stark erweiterte und umgearbeitete auflage edition.



Aebli, H. (1998).

*Zwölf Grundformen des Lehrens: Eine Allgemeine Didaktik auf psychologischer Grundlage. Medien und Inhalte didaktischer Kommunikation, der Lernzyklus.*  
 Klett-Cotta, 10. auflage edition.



Bruner, J. S. (1960).

The process of education.  
 Harvard University Press.



Lion, D. C. T. (1883).

*Johann Amos Comenius \* Pädagogische Schriften.*  
 Bibliothek pädagogischer Klassiker. Hermann Beyer und Söhne, Langensalza.



Pasternak, A. and Vahrenhold, J. (2009).

Rote fäden und kontextorientierung im informatikunterricht.  
 In Peters, I.-R., editor, *Informatische Bildung in Theorie und Praxis*, pages 45–56, Berlin. LOG IN Verlag.



Pasternak, A. and Vahrenhold, J. (2010).

Braided teaching in secondary cs education: Contexts, continuity, and the role of programming.  
 In Cortina, T. and Walker, E., editors, *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE 2010)*, pages 204–208. ACM, ACM Press.



Wagenschein, M. (1968).

*Verstehen lehren.*  
 Beltz Taschenbuch. Beltz.



Wagenschein, M. (1971).

*Die pädagogische Dimension der Physik.*  
 Grundthemen der pädagogischen Praxis. Westermann, Braunschweig.



Wing, J. M. (2006).

**Computational thinking.**

*Communications of the ACM*, 49(3):33–35.



Wing, J. M. (2008).

**Computational thinking and thinking about computing.**

*Philosophical Transactions of The Royal Society*, (366):3717–3725.