

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

Computational Thinking

Jeannette Wing 2006

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

Arno Pasternak Webdatenbanken (3/54)

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

CT&TC 9 Jeannette M. Wing

Arno Pasternak Webdatenbanken (3/54)

Computational Thinking

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]

Arno Pasternak Webdatenbanken (5/54)

Warum Kontextorientierung?

Ziel

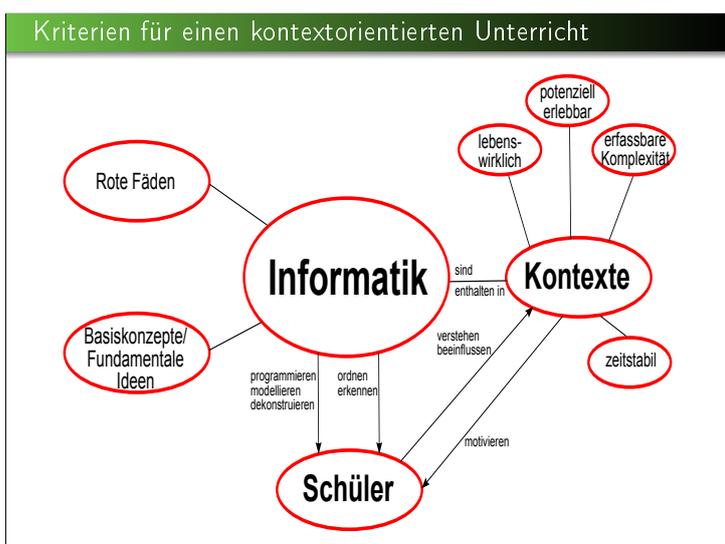
Wir wollen ja auch und gerade Informatik für **Nichtinformatiker** betreiben!

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

Das ist nichts Neues!

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

Arno Pasternak Webdatenbanken (6/54)



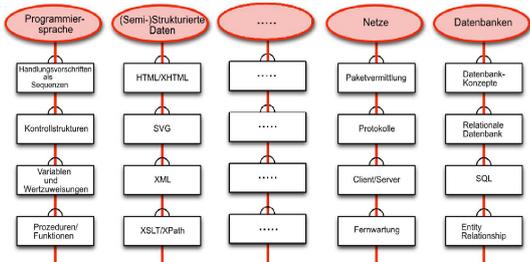
Rote Fäden

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)**

[Pasternak and Vahrenhold, 2009]



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]).

Programmiersprachen

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).

Netze

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. tthttpd) wurden auch **praktisch** durchgeführt bzw. können durchgeführt werden.

Betriebssysteme

Notwendig

„Übliche“ Kenntnisse über

- Arbeit mit der Konsole
- Fernkonsole
- ...

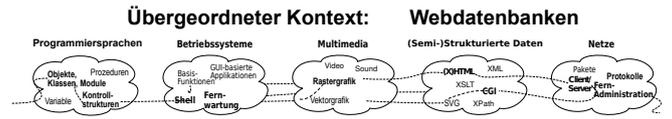
(Evtl.) Ergänzend

Kenntnisse über

- Parallele Prozesse
- Umgebungsvariablen
- ...

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

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]

Datenbanken

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).

HTML

Notwendig

„Übliche“ Kenntnisse über

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

(Evtl.) Ergänzend

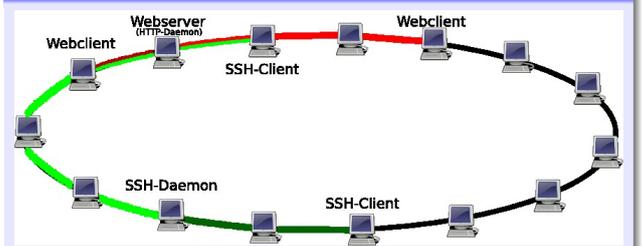
Kenntnisse über

- Formulare
- Statische und dynamische Seiten
- ...

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

Netze

Der Ring als Beispiel für eine Netzstruktur



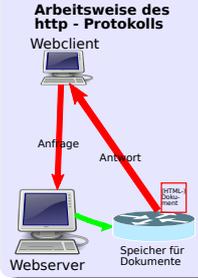
Die Arbeitsweise eines Netz ist **nicht trivial** und ohne Unterricht den Schülern **nicht bekannt**. Der (vereinfachte) Ring ist eine gute Erklärungsgrundlage. Die **Paketvermittlung** kann auch sehr gut gespielt werden.

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



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: `thttpd -d ./ -p 2000`

Erste dynamische HTML-Seite

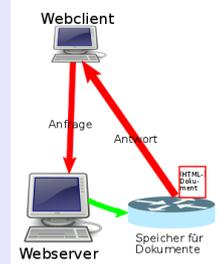
```
// PROGRAMM moin0 (in Groovy);
println("Content-type: text/html");
println();
println("<html>");
println("<head>");
println("<title>Guten Tag</title>");
println("</head>");
println("<body>");
println("Hallo allerseits!");
println("</body>");
println("</html>");
// ENDE
```

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

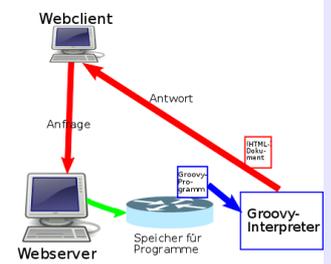
Start des http-Servers **jetzt**: `thttpd -d ./ -p 2000 -c /cgi/*`

Statische und dynamische Webseiten

Statische Seite



Dynamische Seite



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
```

Statische und dynamische Webseiten

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
```

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>
<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

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';

```

... 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
}

```




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.