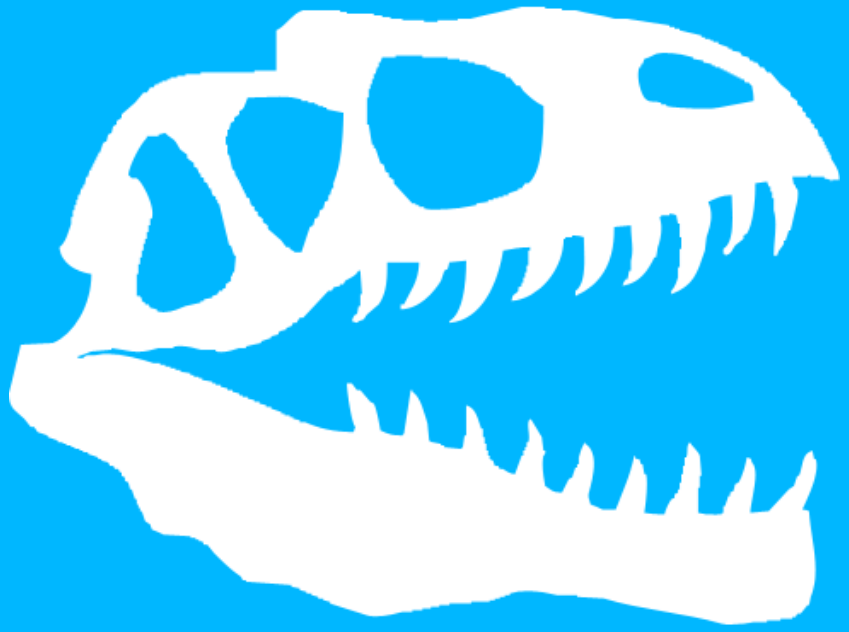
http://unicodesnowmanforyou.com/

JOHANNES HOFMEISTER



@PRO_CESSOR

Page

HTTP://CESSOR.DE

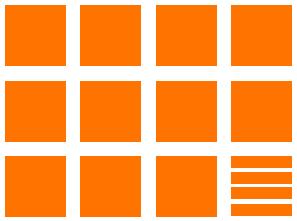
I ♥ CODE

don't you?

Ψ

INFORMATIK

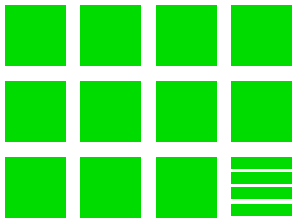
Keine Naturwissenschaft



Machine

SEPARATION

Domain



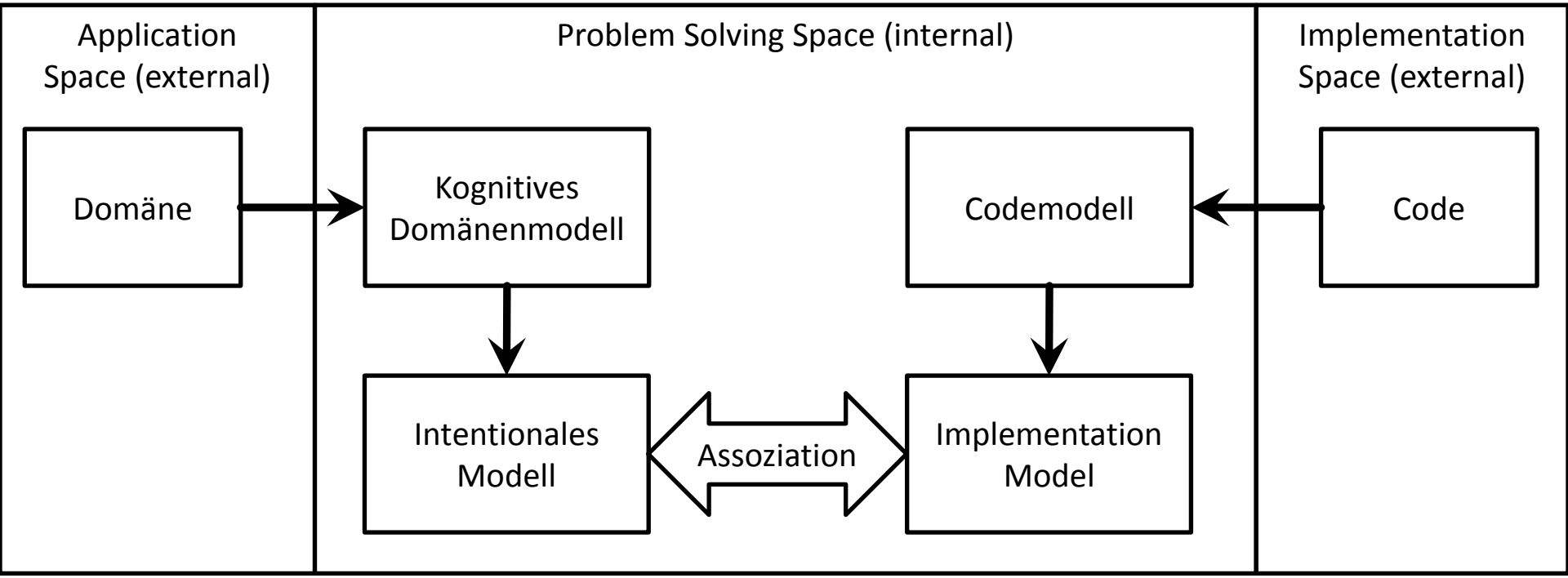
??

Code

Intention

INTENTION

Code


Application
Space (external)
Problem Solving Space (internal)
Implementation
Space (external)
Domäne
Kognitives
Domänenmodell
Codemodell
Code
Intentionales
Modell
Assoziation
Implementation
Model

Marius Schulz
@MariusSchulz

Following

@pro_cessor Ich bin ja dafür, dass du weaselhunter.com aufmachst und da deine entweaselten Renamings auflistest ;)
@asp_net @lanwin

View translation

Reply Retweet Favorite More

9:20 AM - 17 Apr 13

Reply to @MariusSchulz @asp_net @lanwin

Johannes @pro_cessor 17 Apr
@MariusSchulz Ich hab die Domain mal registriert ;)
@asp_net @lanwin
Details

Marius Schulz @MariusSchulz 17 Apr
@pro_cessor Haha, wie cool, hast du ja wirklich gemacht :P
@asp_net @lanwin
Details

http://weaselhunter.com

EMPATHIE

Von Person zu Person

EMPATHY IS THE CAPACITY TO THINK AND FEEL ONESELF INTO THE INNER LIFE OF ANOTHER PERSON

Heinz Kohut
Psychoanalyst

Simple First

Paradigm-Commitment

Abstraction Segregation

Domain Relationship

Listening and Learning

Languages

Binary Dependency

Domain Language

Shared Under-standing

Eloquence

Patterns aren't solutions

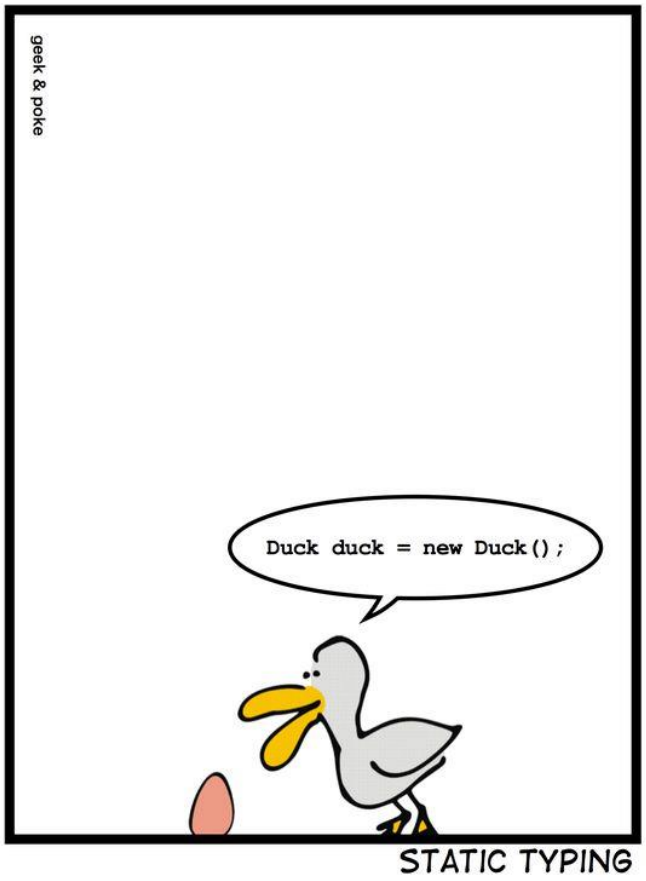
Weasel Word Removal

Simple First!

ERST MAL EINFACH

Richtig schwer!

SIMPLY EXPLAINED

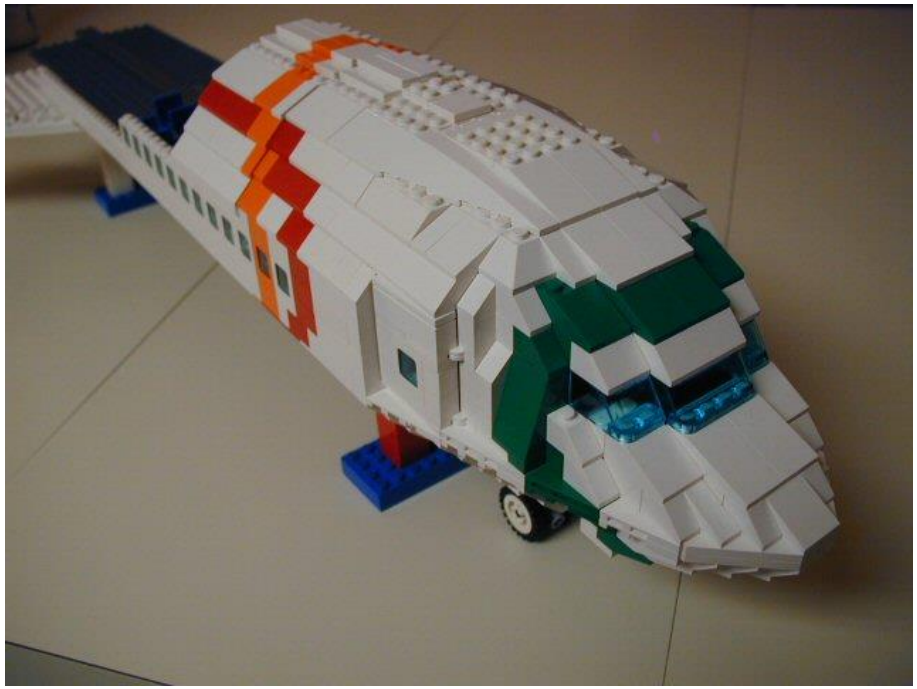


C#

```
var duck = new Duck();
```

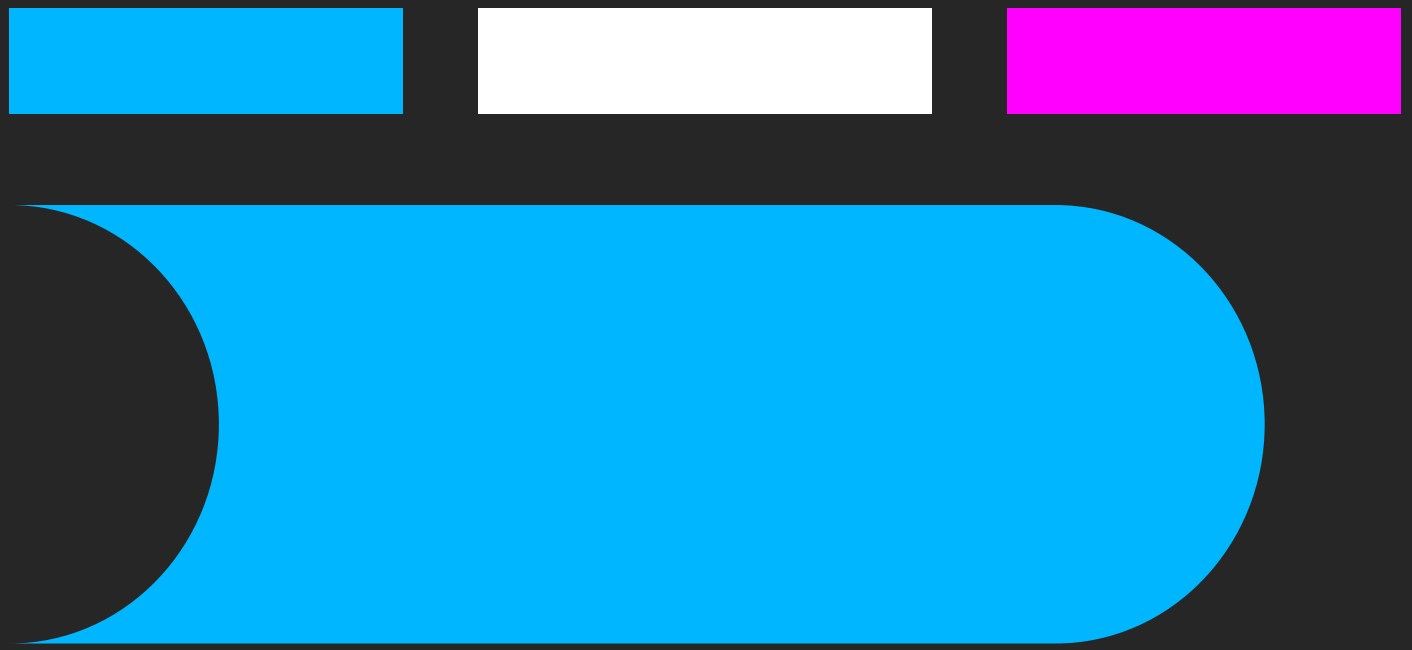
DON'T LET A STRANGER TOUCH YOUR PRIVATES

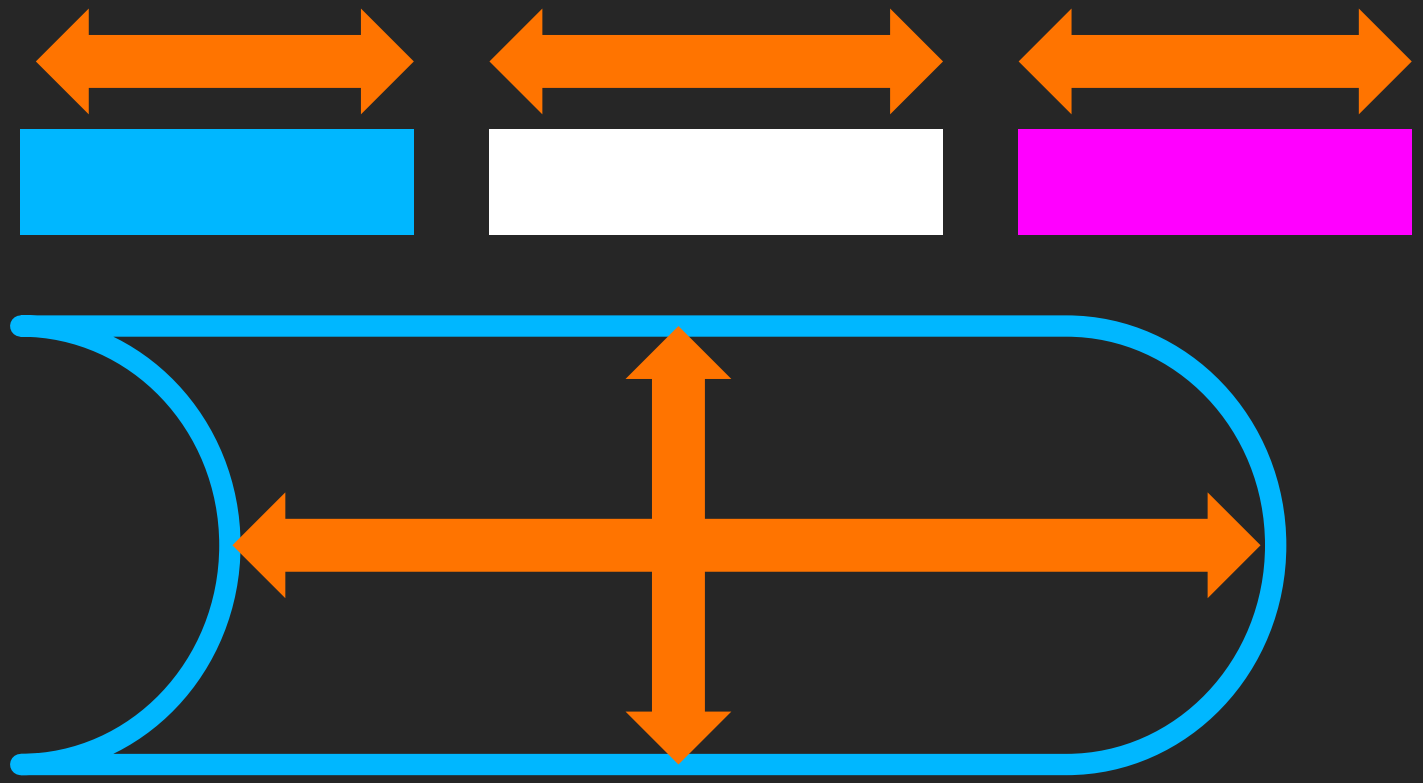
SIZE & REUSE

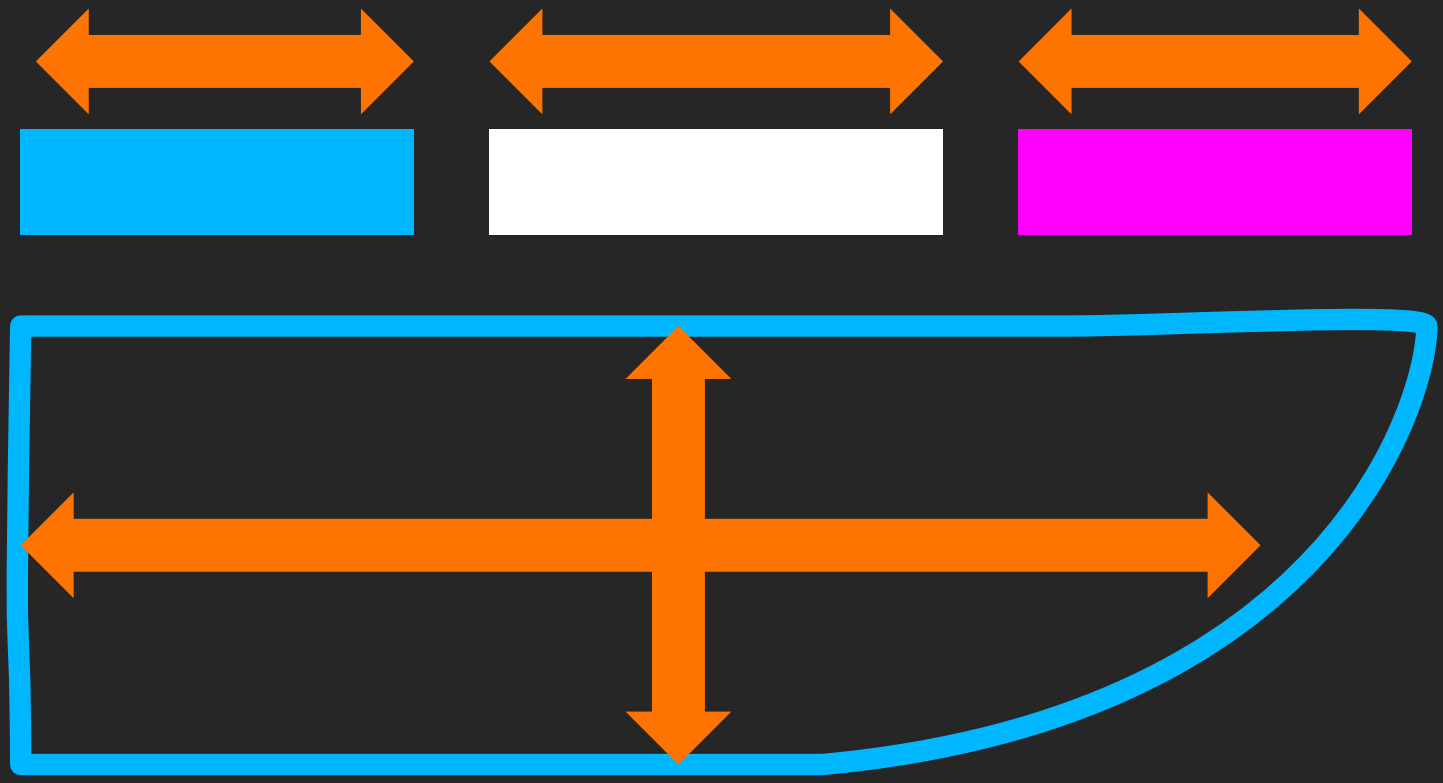


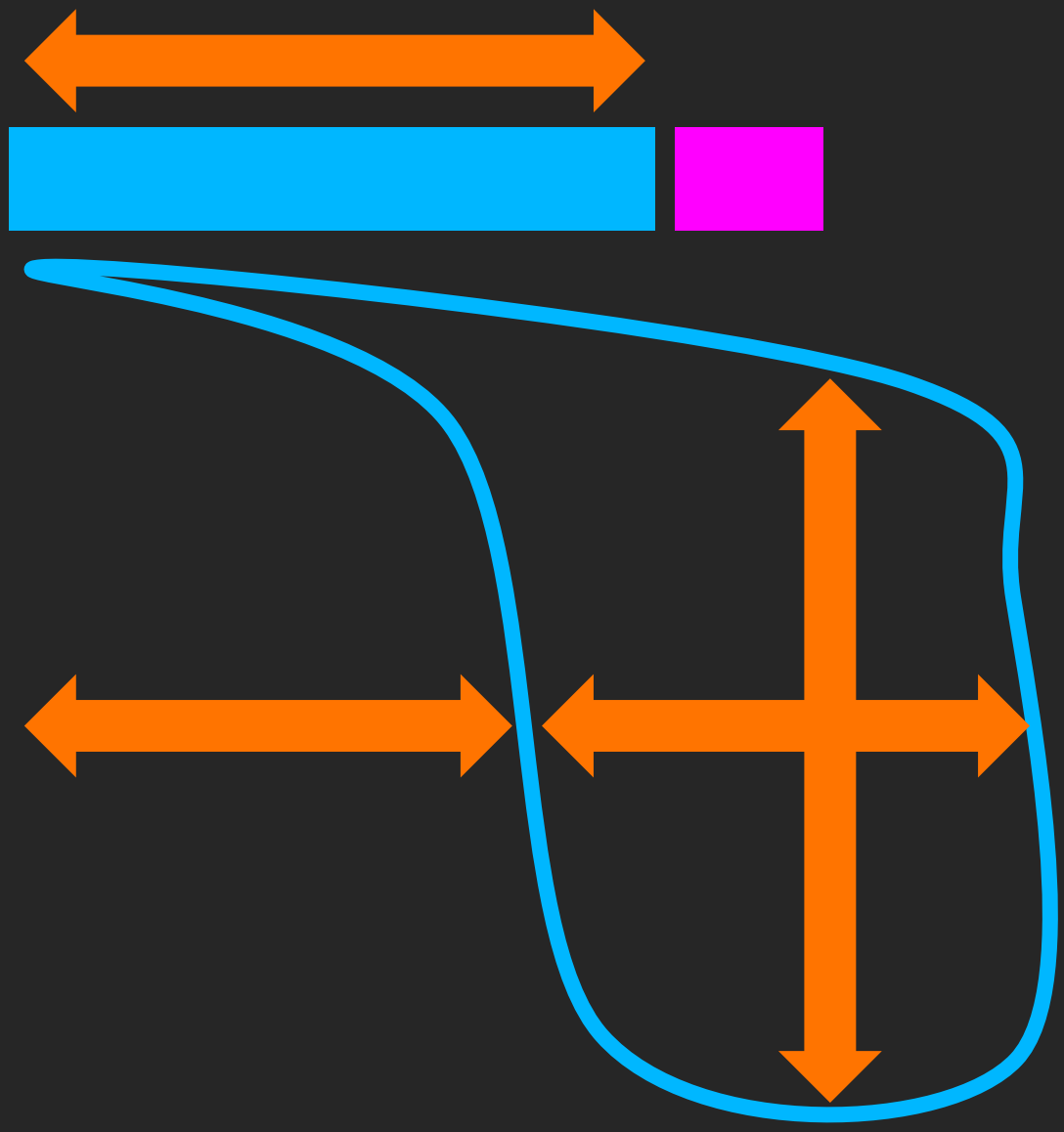
```
public void Execute(MainViewModel mainViewModel)
{
    mainViewModel.Done = () => Break(mainViewModel);
    mainViewModel.StartCounter();
}

public void Break(MainViewModel model)
{
    model.Color = 0x00DBFF.Rgb().Brush();
    model.TimeLeft = 5.Minutes();
    model.Done = () => Work(model);
    model.StartCounter();
}

public void Work(MainViewModel model)
{
    model.Color = 0x00DB00.Rgb().Brush();
    model.TimeLeft = 25.Minutes();
    model.Done = () => Break(model);
    model.StartCounter();
}
```

```
public void Execute(MainViewModel mainViewModel)
{
    mainViewModel.Done = () => Break(mainViewModel);
    mainViewModel.StartCounter();
}

public void Break(MainViewModel model)
{
    model.Color = 0x00DBFF.Rgb().Brush();
    model.TimeLeft = 5.Minutes();
    model.Done = () => Work(model);
    model.StartCounter();
}

public void Work(MainViewModel model)
{
    model.Color = 0x00DB00.Rgb().Brush();
    model.TimeLeft = 25.Minutes();
    model.Done = () => Break(model);
    model.StartCounter();
}
```









```
def filter(markup):
	return add_twitter_names(markup)

def add_twitter_names(markup):
	pattern = "@(\w+)"
	url = "https://twitter.com/"
	link = "<a href='%s\\1'>@\\1</a>" % url
	replacement = link
	return re.sub(pattern, replacement, markup)

def home(entries):
	markup = create_page(entries)
	markup = filter(markup)
	return markup

def main():
	entries = Entries()
	if len(argv) > 1:
		date = argv[1]
		entries = entries.written_on(date)
	print home(entries)
```

GUARD CLAUSES

```
if(someValue != null)
{
	doSomething();
	doSomethingElse(result);
	result = doSomeMoreSomething();
	return result;
}
return null;
```

```
if(someValue != null) { return null; }

doSomething();
doSomethingElse(result);
result = doSomeMoreSomething();
return result;
```

FAIL EARLY

CYCLOMATIC COMPLEXITY

NESTED IFS

NUMBER OF PATHS

2 ** 5 = 32

GUARD CLAUSES

2 * 5 = 10

CAN I HOLD THESE IN MEMORY?

IST DER CODE PERFORMANT VON MEINEM GEHIRN AUSFÜHRBAR?

1 2 2 4 9 0

12

24

90

12/24/90

Weihnachten!

12/24/90

CHUNKING

```
if(someValue != null)
{
	Lorem ipsum = dolor sit;
	amet = consectetuer(adipiscing, elit);
	laoreet = sed.diam(nonummy);
	nibh.euismod(tincidunt,ut);
	laoreet = dolore.magna ? aliquam : erat.volutpat;
}
else {
	Eodem.modo().typi(qui).nunc(nobis => {
		videntur.parum(clari).fiant();
		return sollemnes(in,futurum);
	});
	laoreet = sed.diam(nonummy);
	nibh.euismod(tincidunt,ut);
}
return null; // Bis hier her habe ich den Context vergessen.
```

CHUNKS

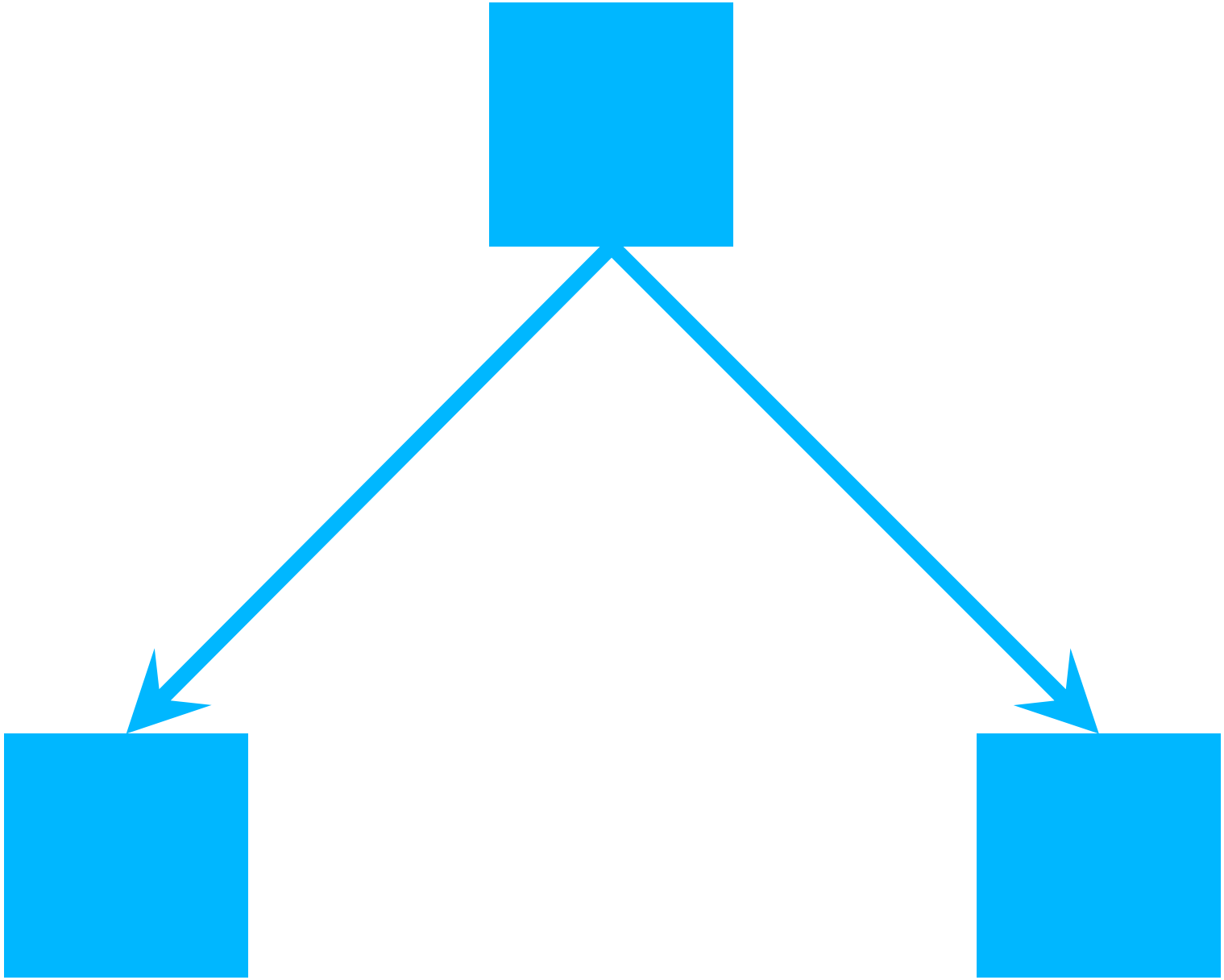
im Code?

```
public class Conference {
        DateTime Date { get; private set; }
        string Name { get; private set; }
}
```

```
var expenses = function (shop) {
        shop.TotalSocialCosts = totalSocialCosts(shop);
        var balance =
                shop.Advertising
                + (shop.Distributors * 500)
                + rent(shop.Location)
                + shop.Service
                + salaries(shop)
                + shop.TotalSocialCosts
                + storage(shop);
        debitAccount(shop,balance);
};
```

Ideal:

BINÄRE CHUNKS



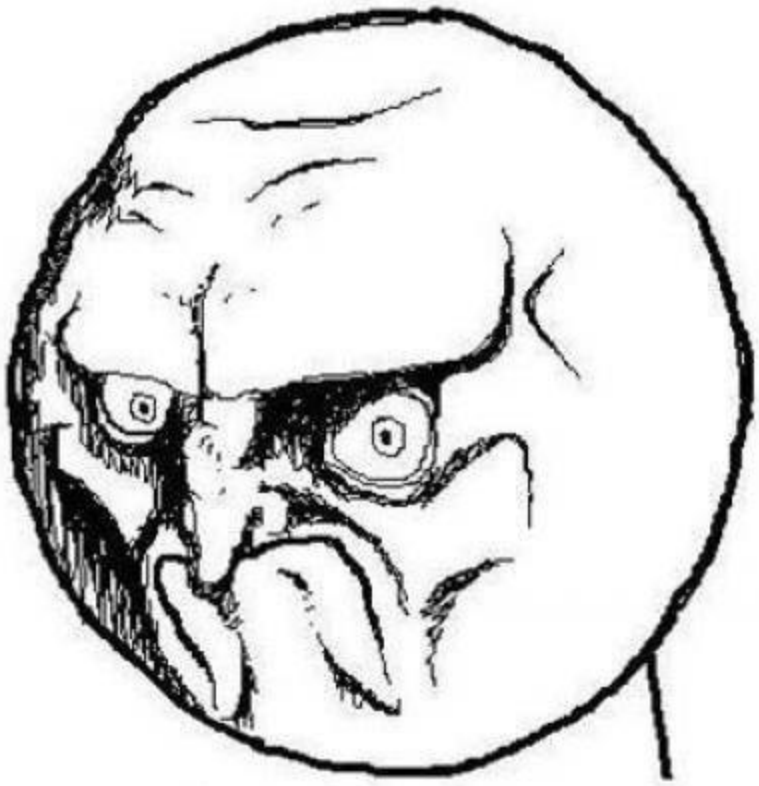
IF THE IMPLEMENTATION IS HARD TO EXPLAIN, IT'S A BAD IDEA.

python

Richtig

ZUHÖREN

und dabei lernen!



NO.

Zum Üben…
https://github.com/cessor/refuctoring

“You can call it beautiful code when the code also makes it look like the language was made for the problem”

WARD CUNNINGHAM.

Quicksort

```
public class
{
    public static int[] Sort(int[] array)
    {
        int[] a = new int[array.Length];
        array.CopyTo(a, 0);
        Sort(0, array.Length - 1, ref a);
        return a;
    }

    private static void Sort(int links, int rechts, ref int[] daten)
    {
        if (links >= rechts) return;
        int teiler = Divide(links, rechts, ref daten);
        Sort(links, teiler - 1, ref daten);
        Sort(teiler + 1, rechts, ref daten);
    }

    private static int Divide(int left, int right, ref int[] data)
    {
        int leftpos = left;
        int rightpos = right - 1;
        int pivot = data[right];

        do {

            while (data[leftpos] <= pivot && leftpos < right) leftpos++;
            while (data[rightpos] >= pivot && rightpos > left) rightpos--;
            if (leftpos >= rightpos) continue;
            Swap(leftpos, rightpos, ref data);

        } while (leftpos < rightpos);

        if (data[leftpos] > pivot) {
            Swap(leftpos, right, ref data);
        }
        return leftpos;
    }

    private static void Swap(int left, int right, ref int[] data)
    {
        int z = data[left];
        data[left] = data[right];
        data[right] = z;
    }
}
```

```
let rec quicksort (list:int list) =
    match list with
    | [] -> []
    | head::tail ->
        let smaller,larger = List.partition (fun y -> y <= head) tail
        quicksort smaller @ [head] @ quicksort larger
```

“Clean code reads like well-written prose”

GRADY BOOCH

Gesprochene,

NATÜRLICHE

Sprache

```
var date = new DateTime (2012, 4, 14, 16, 32, 18, 500);
```

```
var start = 14.April(2012).At(8.PM());

var end = 8.Hours().Later(start);
```

Trennen von Einzelheiten

ABSTRAKTION

in Intention und Maschine

Wenn du darüber sprichst...

DOMÄNENBEZUG

...ist es wahrscheinlich wichtig

WAR STORIES

Also ich hab da...

“Also ich hab da so ne Software”

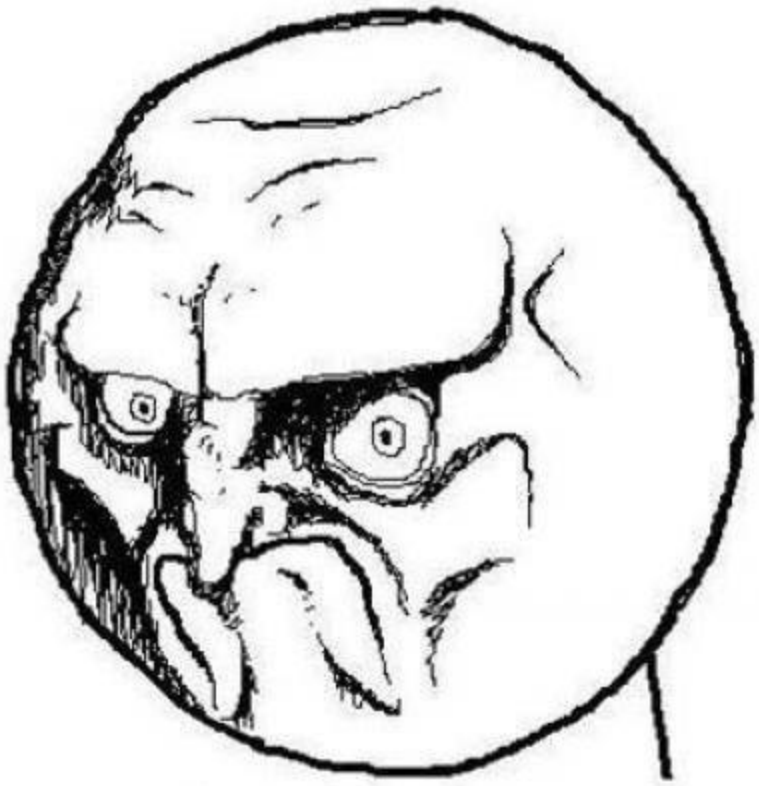
“Also ich hab da so ne Software, die bearbeitet so Bilder.”

“Also ich hab da so ne Software, die bearbeitet so Bilder. Aber ich hab da so zwei viel zu große Klassen.”

“Also ich hab da so ne Software, die bearbeitet so Bilder. Aber ich hab da viel zu große Klassen. Und ich reiche da so ein Bytearray rum.”

“Also ich hab da so ne Software, die bearbeitet so Bilder. Aber ich hab da viel zu große Klassen. Und ich reiche da so ein Bytearray rum. Und das wird dann von allen bearbeitet.”

“Also ich hab da so ne Software, die bearbeitet so Bilder. Aber ich hab da viel zu große Klassen. Und ich reiche da so ein Bytearray rum. Und das wird dann von allen bearbeitet. Das ist schon ziemlich kompliziert…”



NO.

PROGRAMMER

Y U NO OBJECT ORIENT?

BILDER
SIND KEINE BYTEARRAYS

EMAIL ADRESSEN
SIND KEINE STRINGS

POLYMORPHISMEN

TYPEN

NULLOBJEKTE

BIS HIER: EMPATHIC CODE.

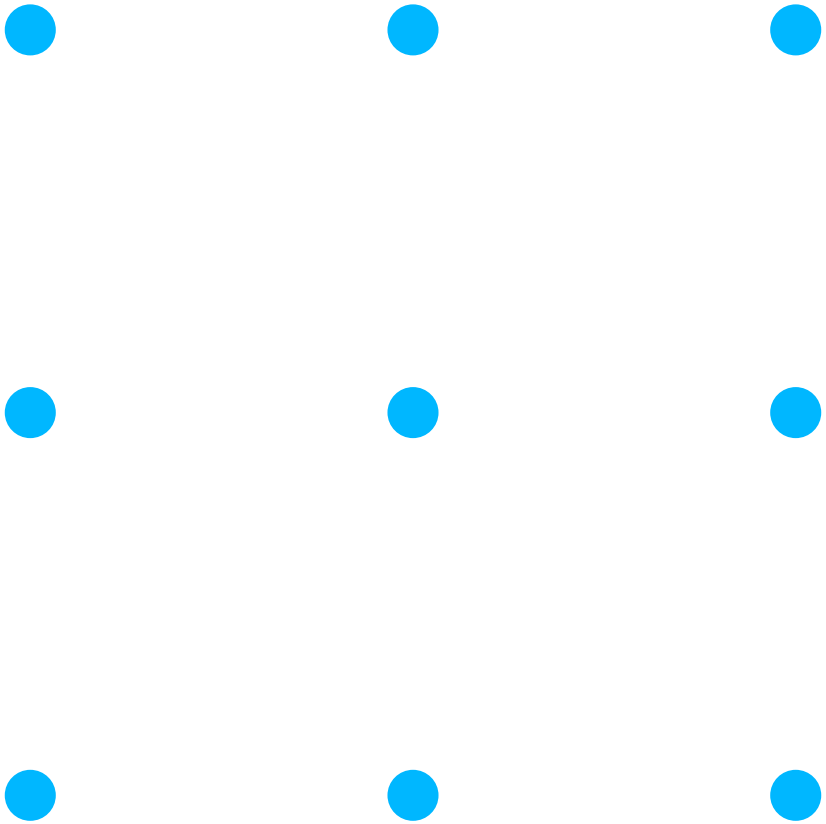
THEORIE.

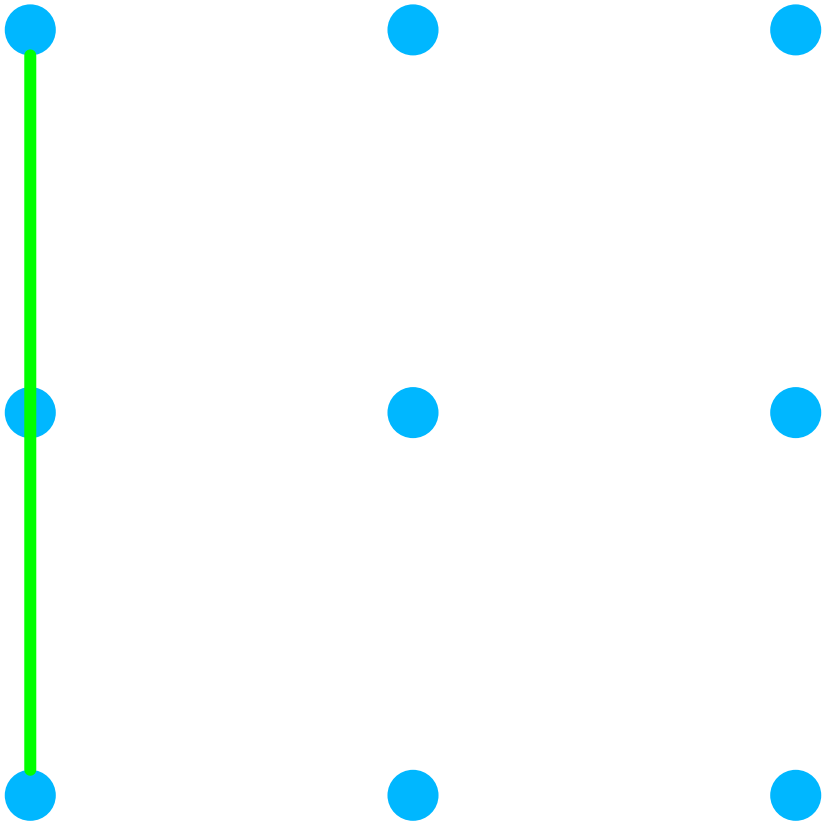
Ψ

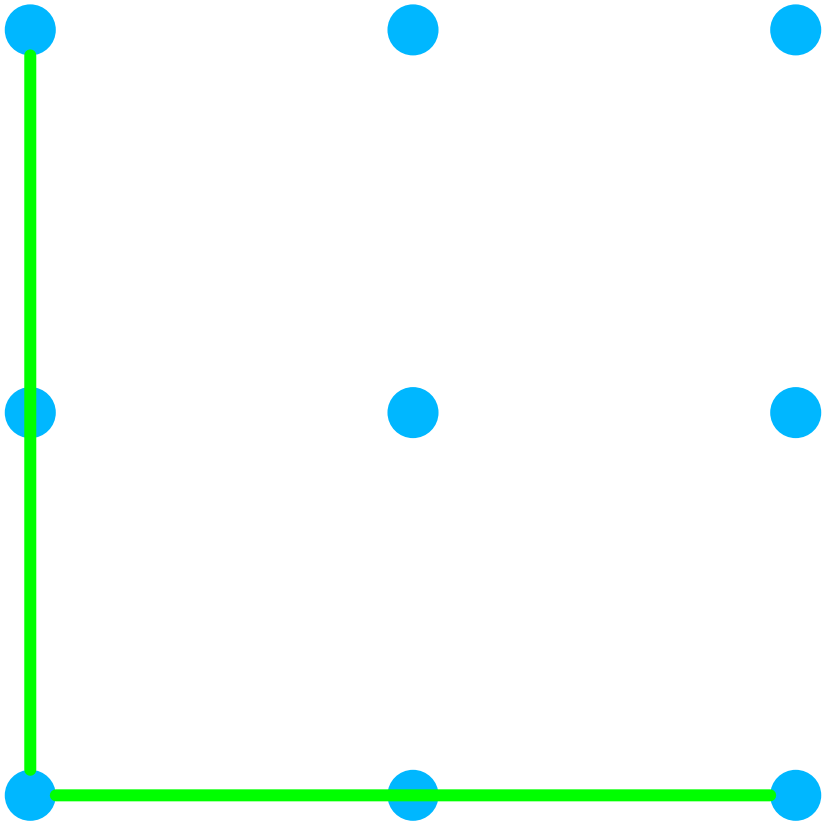
COMPLEX SYSTEMS

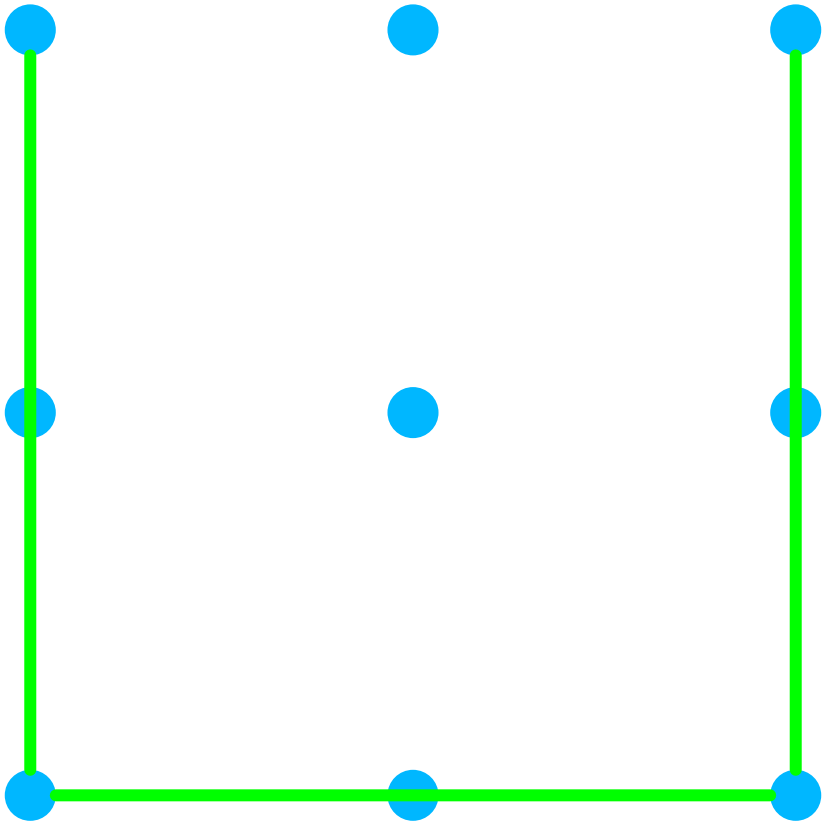
COMPLEX PROBLEM SOLVING

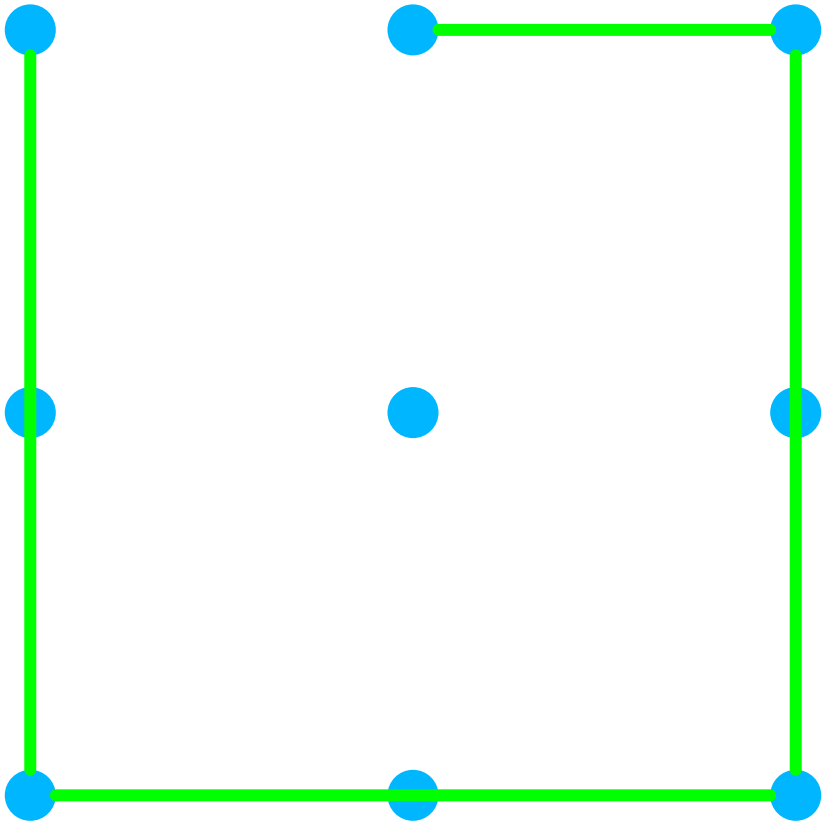
WHAT IS A PROBLEM?

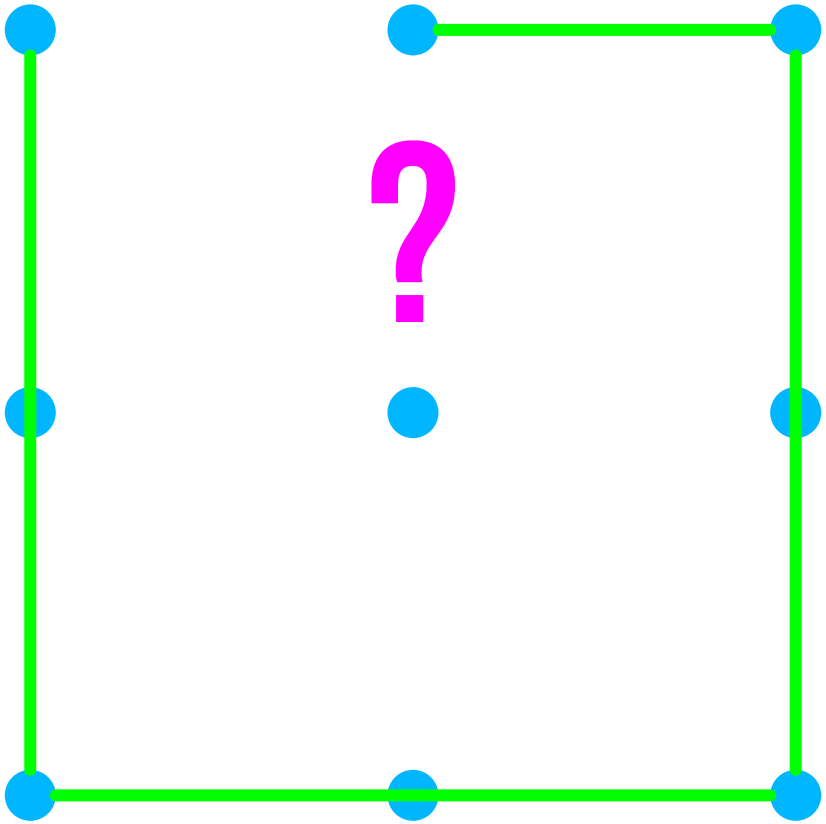


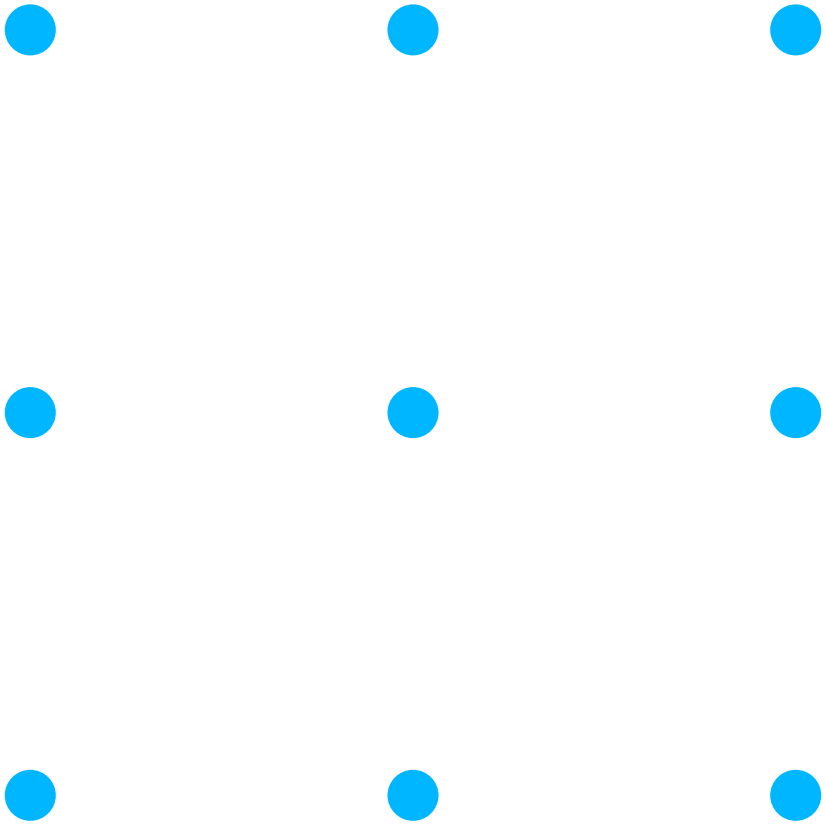


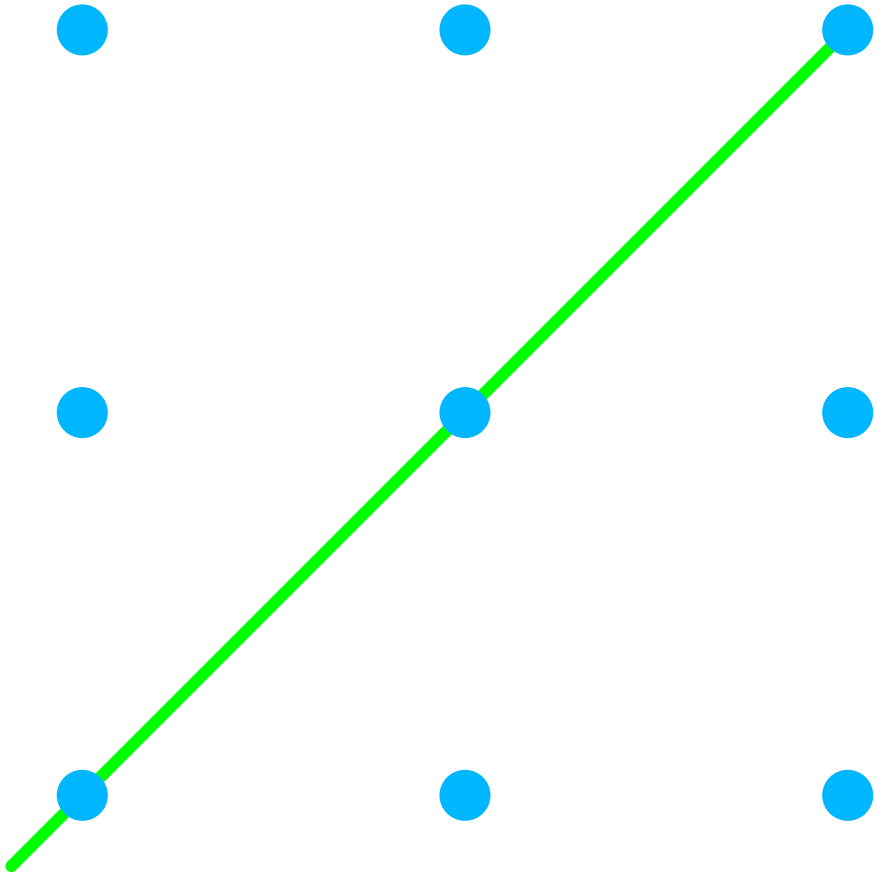


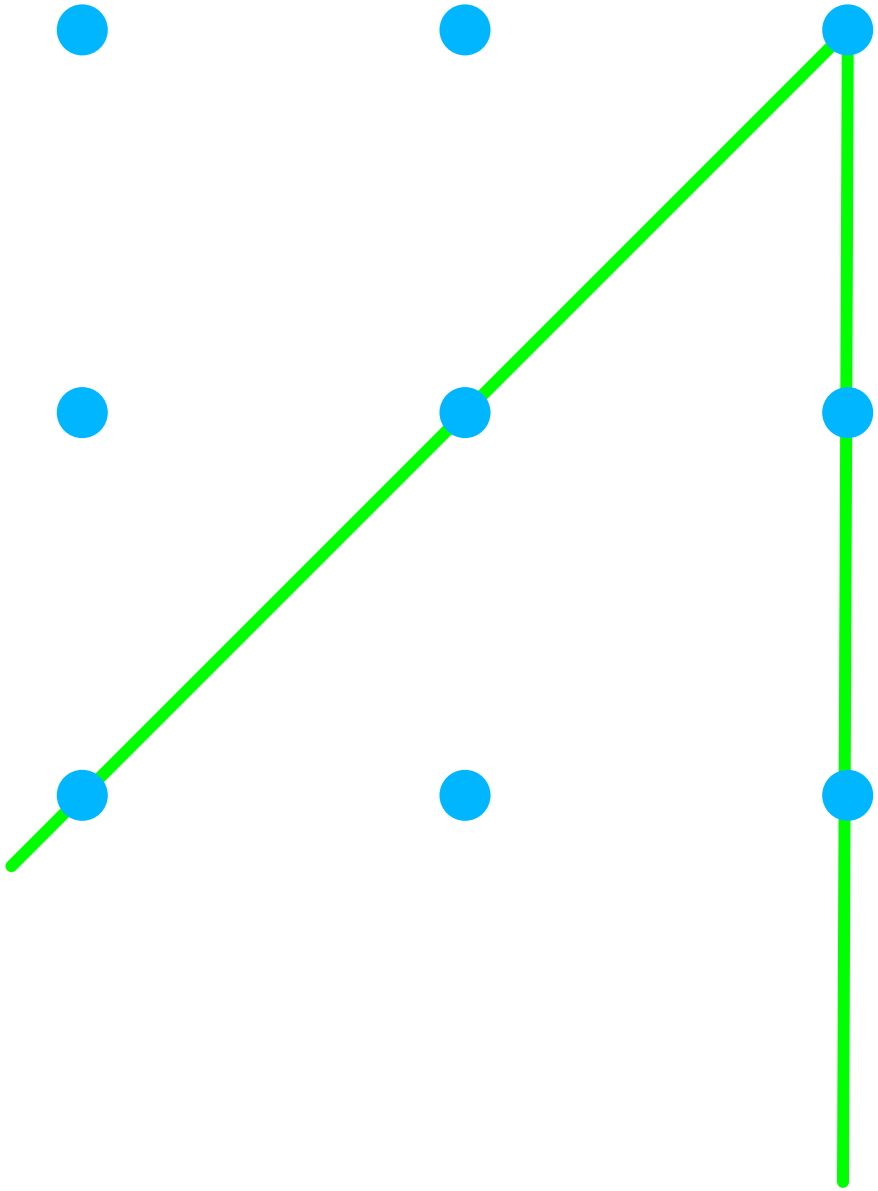


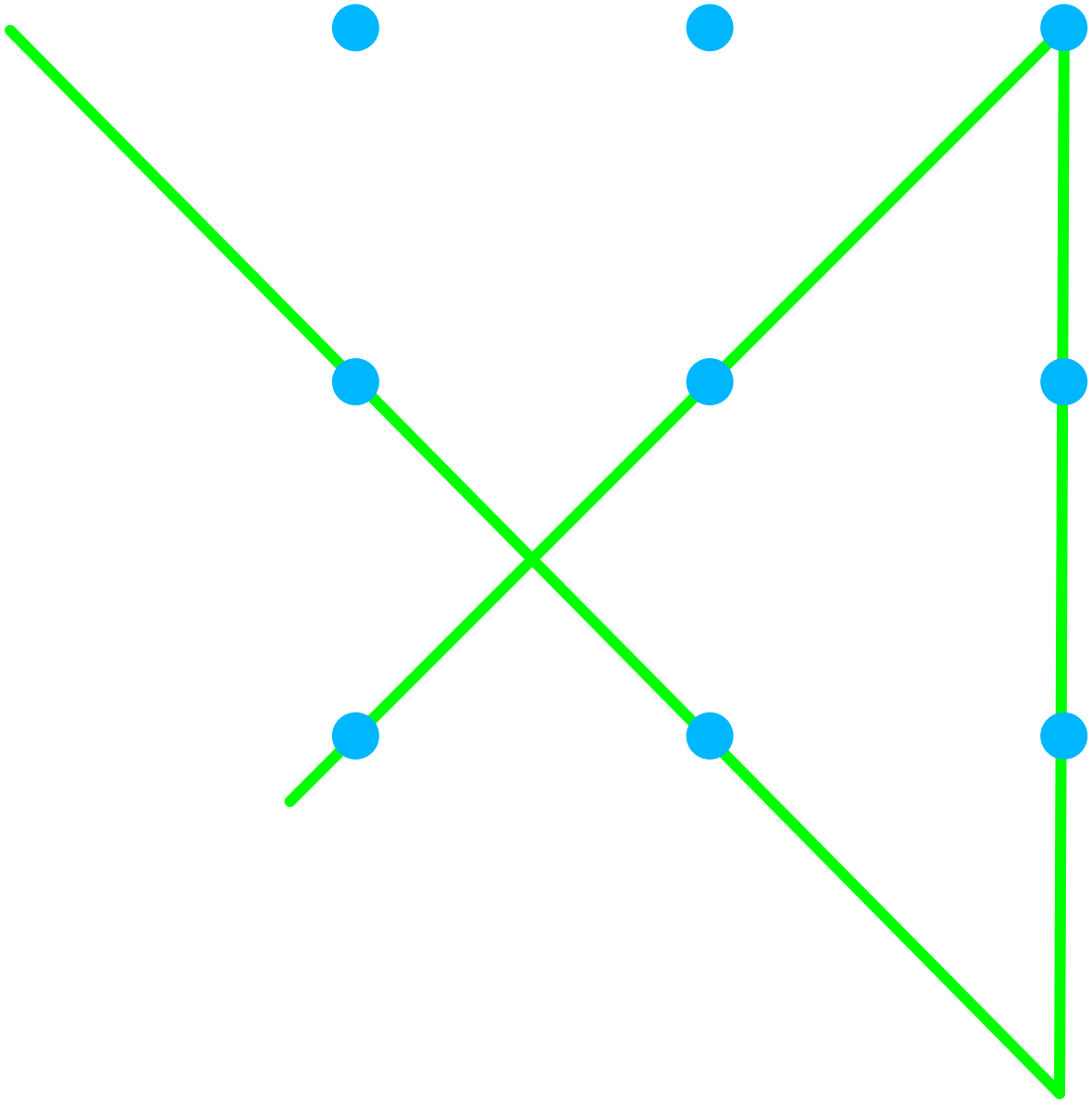


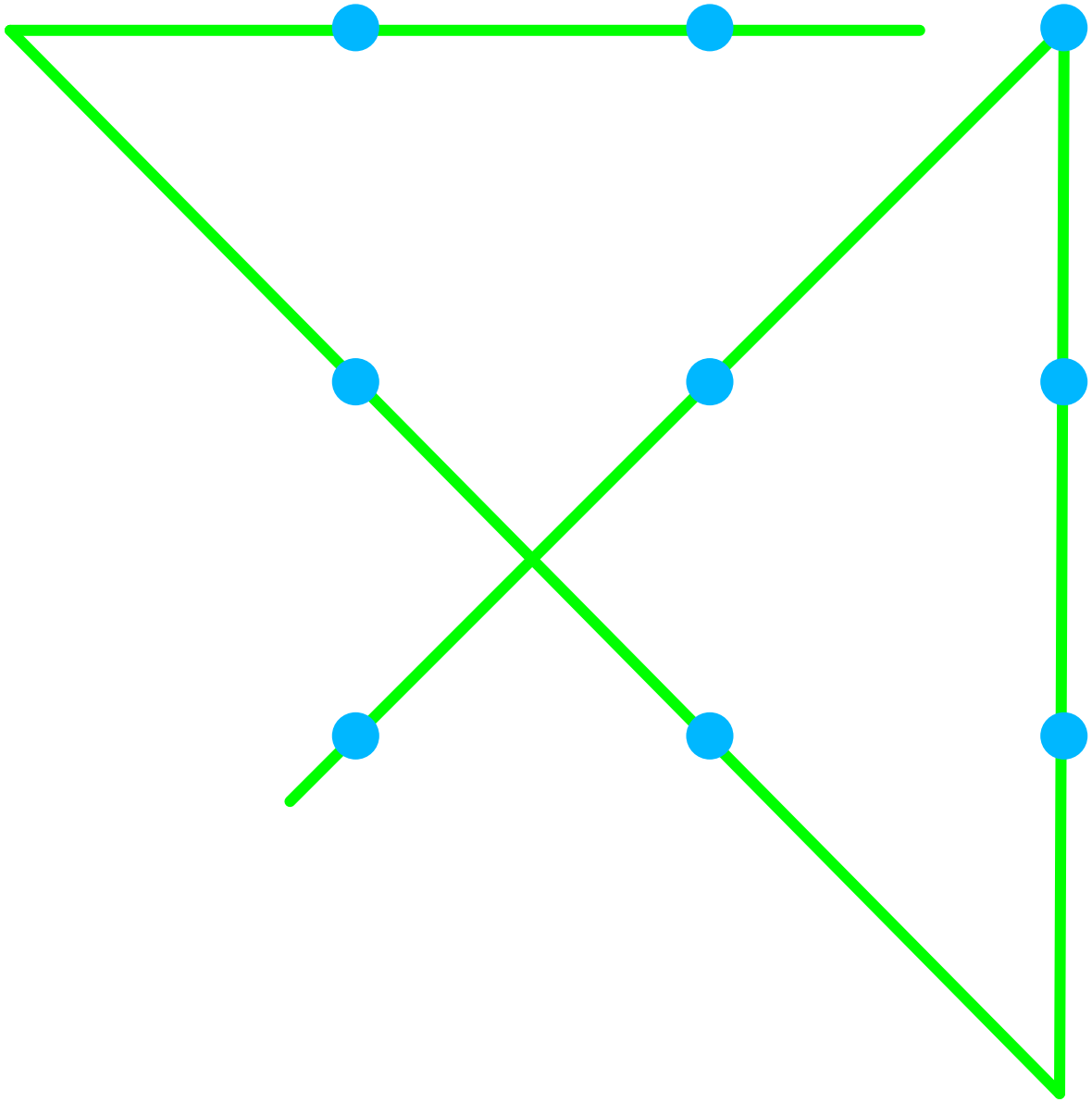


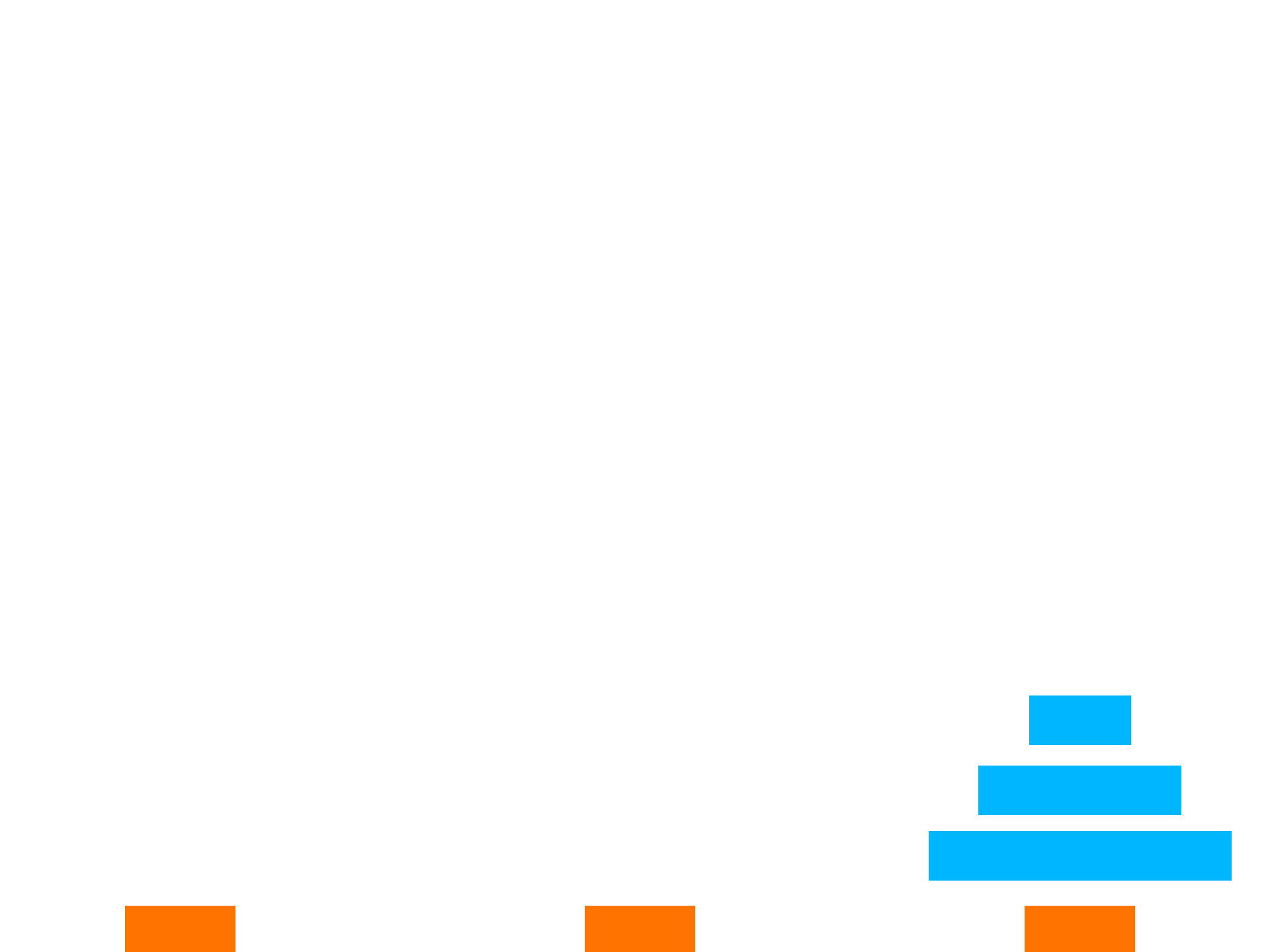












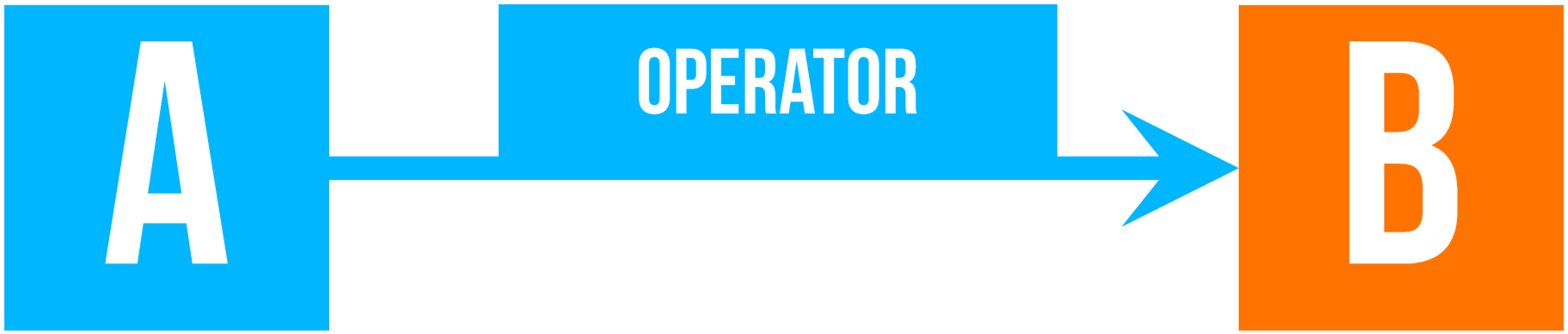
ABSTRACT...

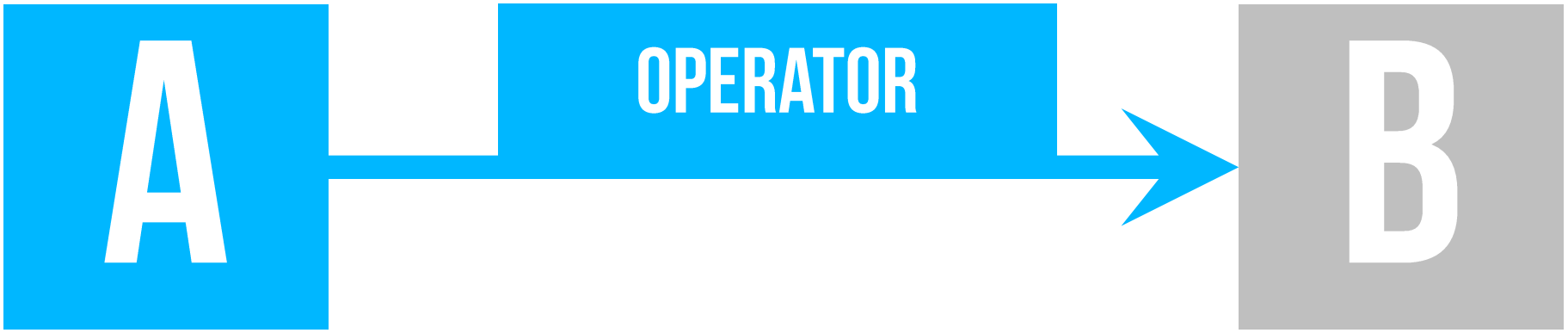

A


A


B


A
B


A
OPERATOR
B

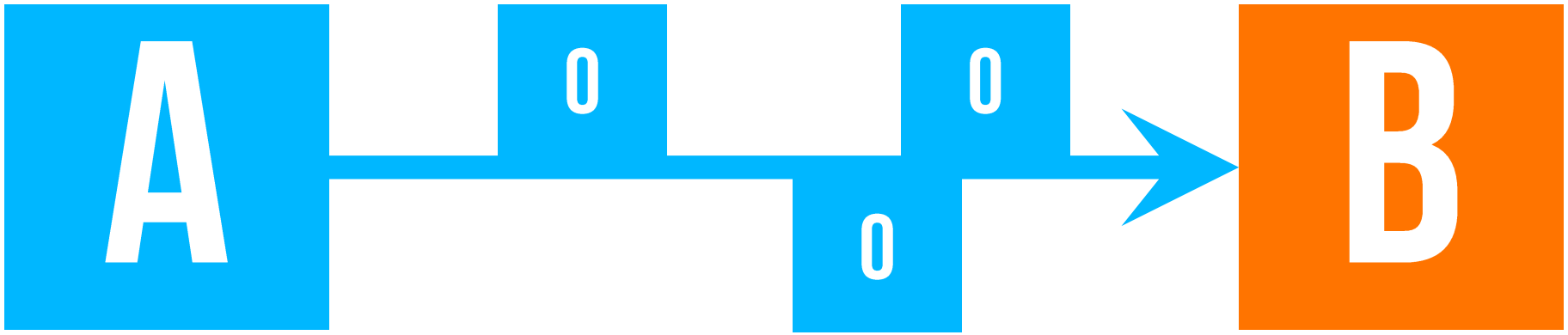

A
OPERATOR
B


A
UNBEKANNT
B

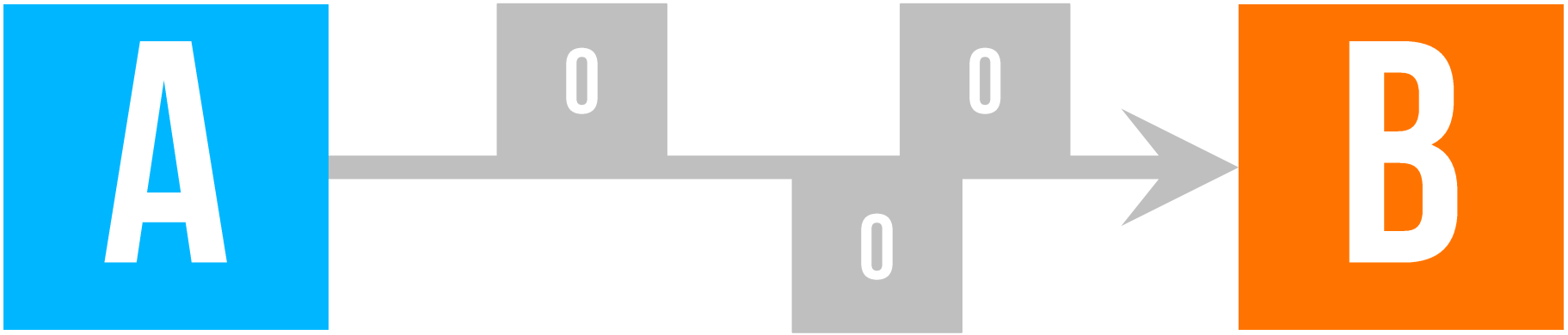

A
UNBEKANNT
B

Dietrich Dörner

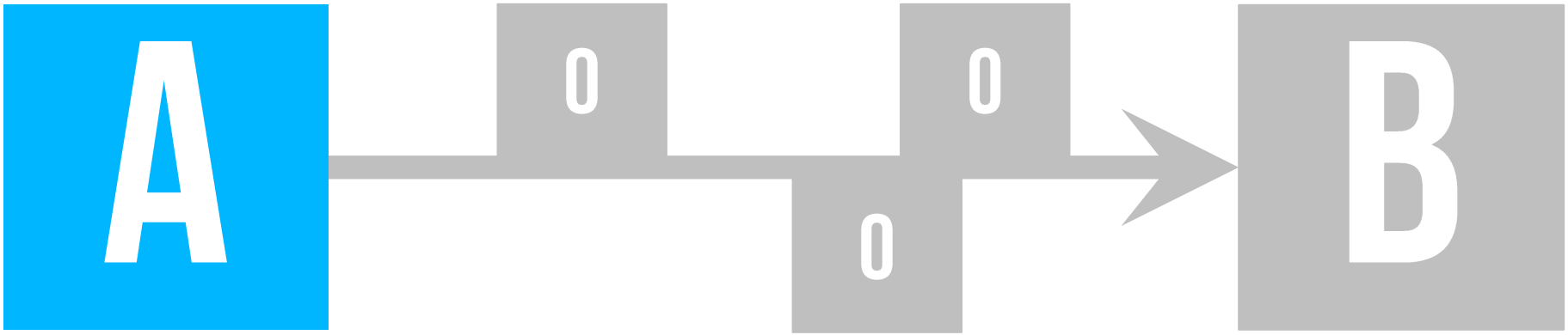
INTERPOLATION



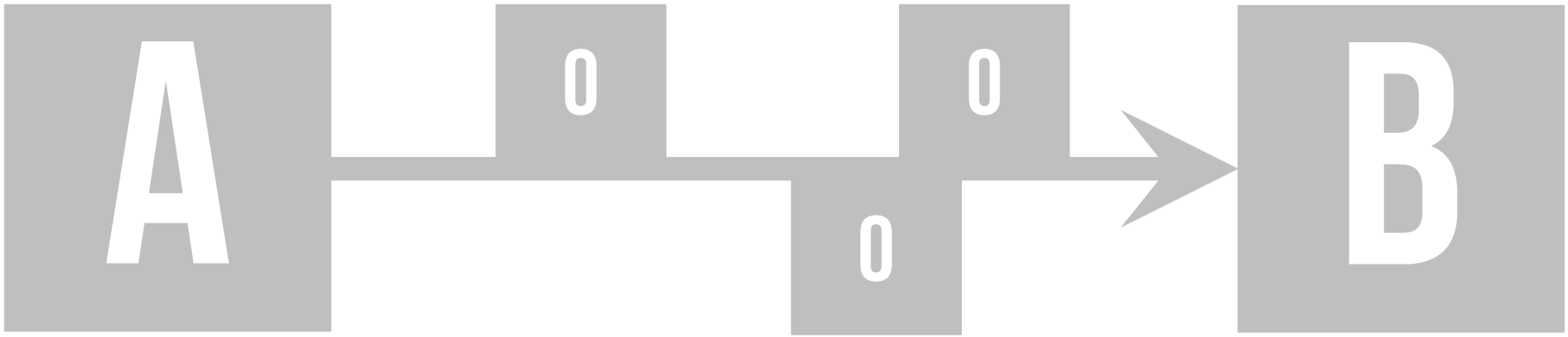
SYNTHESE



DIALEKTISCH



??


A
B

Joachim Funke

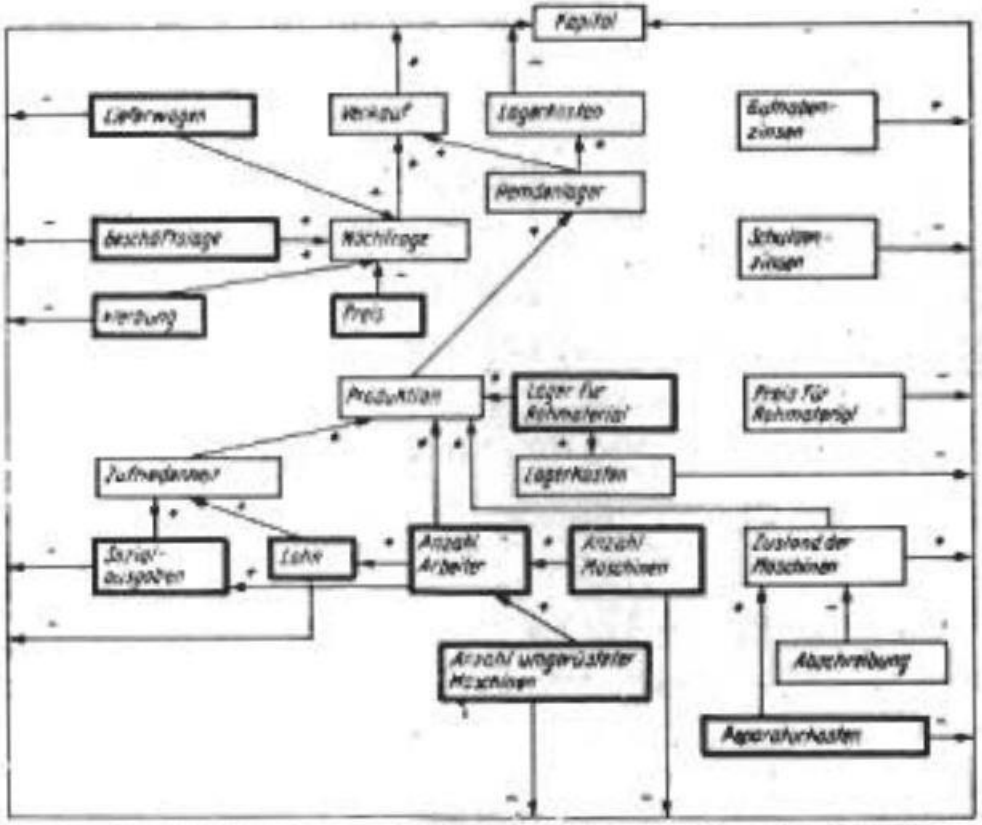
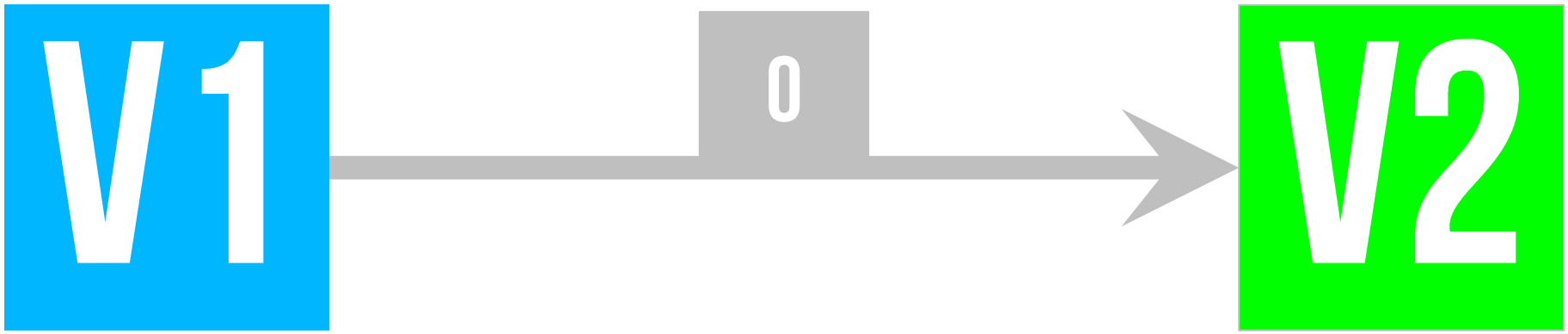
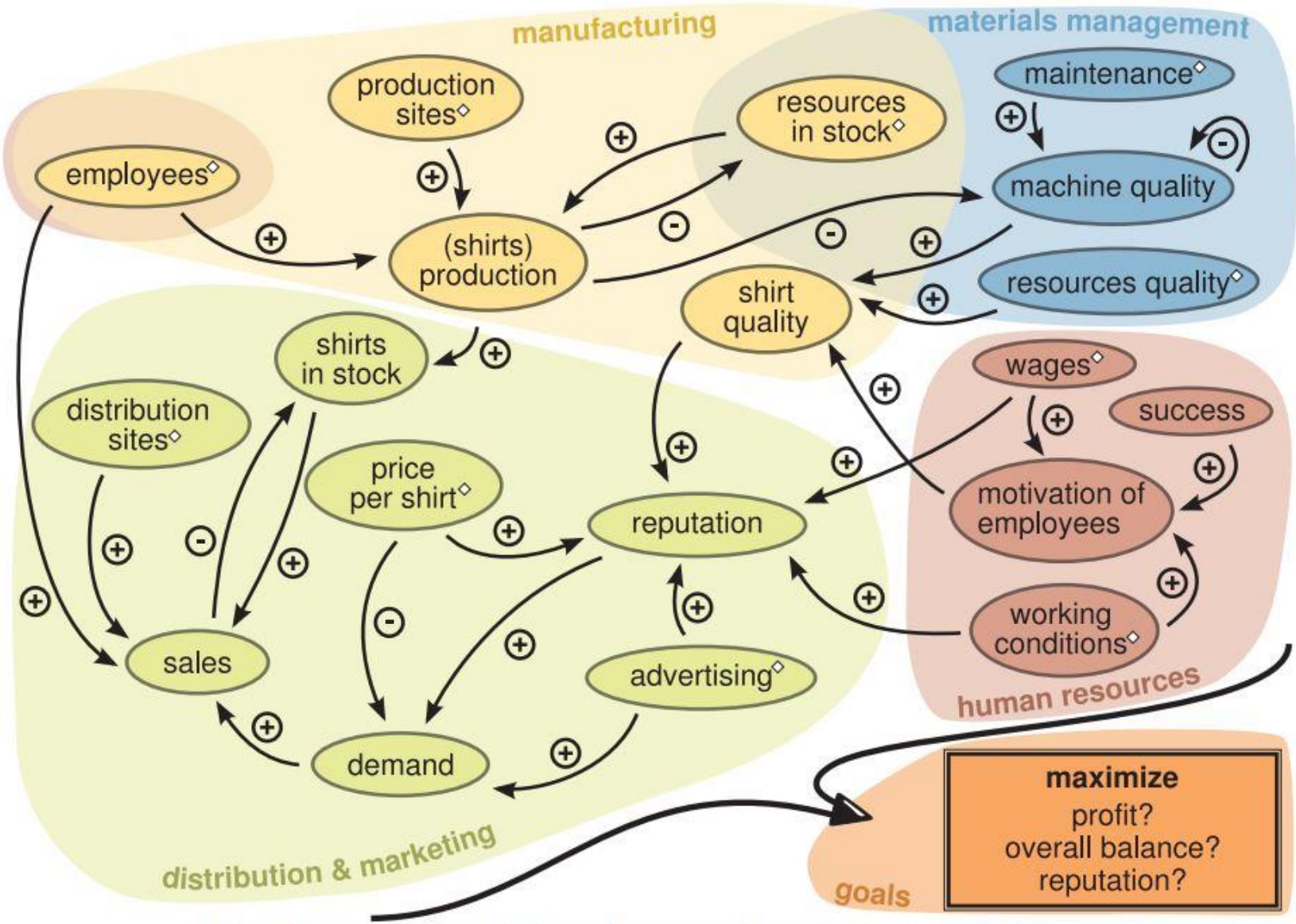


Abb. 1: Die Variablen der Schneiderwerkstatt und ihre Verknüpfungen. Die dick umrandeten Variablen sind direkt beeinflußbar (N = 11); + = positive Korrelation, d.h. z.B. steigt der Lohn, so steigt auch die Zufriedenheit; – = negative Korrelation, d.h. z.B. je mehr Maschinen gekauft sind, desto niedriger ist zunächst das Kapital (aus: Putz-Osterloh 1981, S.83).

Funke, J. (1983) Einige Bemerkungen zu Problemen der Problemlöseforschung: Oder, ist Testintelligenz doch ein Prädiktor? Diagnostica 1983, S. 383 -302

SYSTEME





Tailorshop *concept with dependencies and positive/negative influences.*
Diamonds show participants' control possibilities.

http://www.psychologie.uni-heidelberg.de/ae/allg/mitarb/jf/Engelhart_etal%20SIOPT_2011%20tailor_poster.pdf

Tabelle 2:
Vernetzung der Variablen in der Schneiderwerkstatt*

Nr. der Gleichung	Variable	Beziehung
(1)	ZA	ZA = Min (ZM, (0.5 + ((LO – 850)/550) + SM/800))
(2)	PM	PM = (Min (N1, A1) * (MA + ζ* 4 – 2) + Min (N2, A2)* (MA*2 + ζ*6 – 3)) * $\sqrt{ZA}$
(3)	PA	PA = Min (PM, RL)
(4)	HL	HL = HL + PA – VH
(5)	RL	RL = RL – PA
(6a)	NA_1	$NA_1 = NA_2/2 + 280*1.25*e^{(-(PH^2/4250))}$
(6b)	NA_2	NA_2 = Min (WE/5, NM) + LW*100 $NA_2 = NA_2 + NA_2*g + ζ*100 - 50$ wobei g = 0 für GL = Vorort = 0.1 für GL = Cityrand = 0.2 für GL = City
(7)	VH	VH = Min (HL, NA_1)
(8)	RP	RP = 2 + ζ*6.5
(9)	MA	MA = Min ((MA – 0.1*MA + (RS/(A1+A2))*0.017), MM)
(10a)	KA	KA = KA – SM*(N1+N2) – PA*1 – RL*.5 – HL*1 + VH*PH – WE – LW*500 – GL*2000 – RS – (N1+N2)*LO
(10b)	KA	KA = KA + KA*z, wobei z = 0.0025, wenn KA $\geq$ 0 = 0.0066, wenn KA $<$ 0

* Das Zeichen ζ steht für eine Zufallszahl im Bereich zwischen Null und Eins, erzeugt von einem rechnerinternen Pseudozufallszahlengenerator.

Funke, J. (1983) Einige Bemerkungen zu Problemen der Problemlöseforschung: Oder, ist Testintelligenz doch ein Prädiktor? Diagnostica 1983, S. 383 -302

Tabelle 1:
Liste der Variablen im Programm TAILOR-Shop

Kürzel	Bezeichnung	Anfangswert
ZA	aktuelle Zufriedenheit	0.981
PA	aktuelle Produktion	403.932
PM	mögliche Produktion	403.932
KL	Hemden im Lager	80.716
RL	Rohmaterial im Lager	16.068
NA	Nachfrage aktuell	766.636
VH	verkaufte Hemden	407.216
PH	Preis pro Hemd	52.000
RS	Reparatur- und Servicekosten	1200.000
MA	aktuelle Maschinenkapazität	47.040
WE	Werbe-Kosten	2800.000
LO	Lohnkosten pro Arbeiter	1080.000
SM	Sozialausgaben pro Arbeiter	50.000
N1	Anzahl Arbeiter an 50er	8
N2	Anzahl Arbeiter an 100er	0
A1	Anzahl 50er-Maschinen	10
A2	Anzahl 100er-Maschinen	0
LW	Anzahl Lieferwagen	1
GL	Geschäftslage (0 = Vorort, 0.5 = Cityrand, 1 = City)	0.5
KA	Kapital	15774.659
RP	Rohmaterial-Preis	3.994
ND	Anzahl regierter Monate	0

Funke, J. (1983) Einige Bemerkungen zu Problemen der Problemlöseforschung: Oder, ist Testintelligenz doch ein Prädiktor? Diagnostica 1983, S. 383 -302

TI-BASIC

GWBASIC

FLASH
ACTION SCRIPT 2

ZIEL:
JAVASCRIPT

CODE

Lieber Johannes,

Zeig den Leuten doch mal ein bisschen Code!

- Ich, vor knapp 13 Stunden

HIER BIST DU RICHTIG!

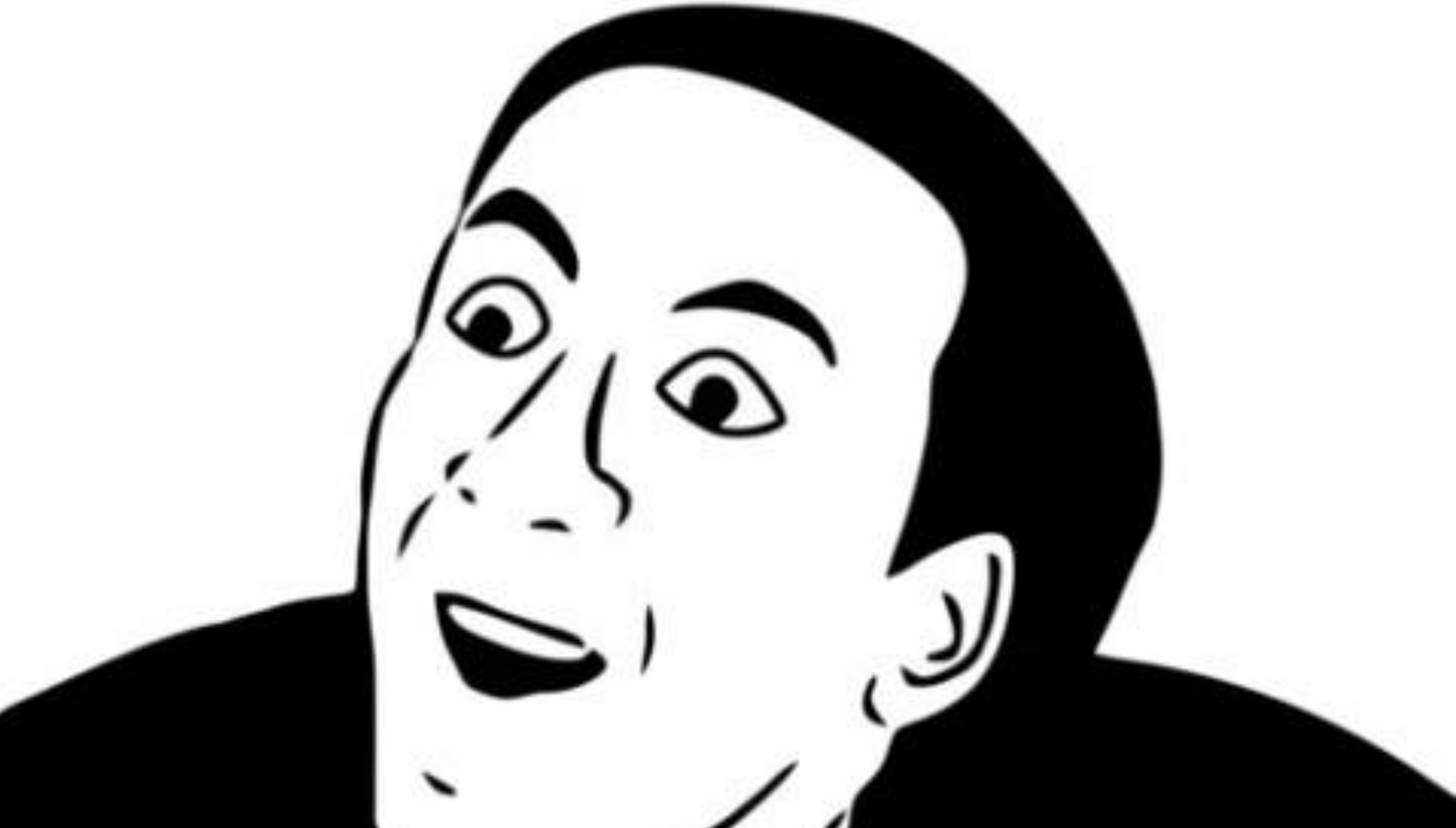
Alle

KOMMENTARE

entfernen!

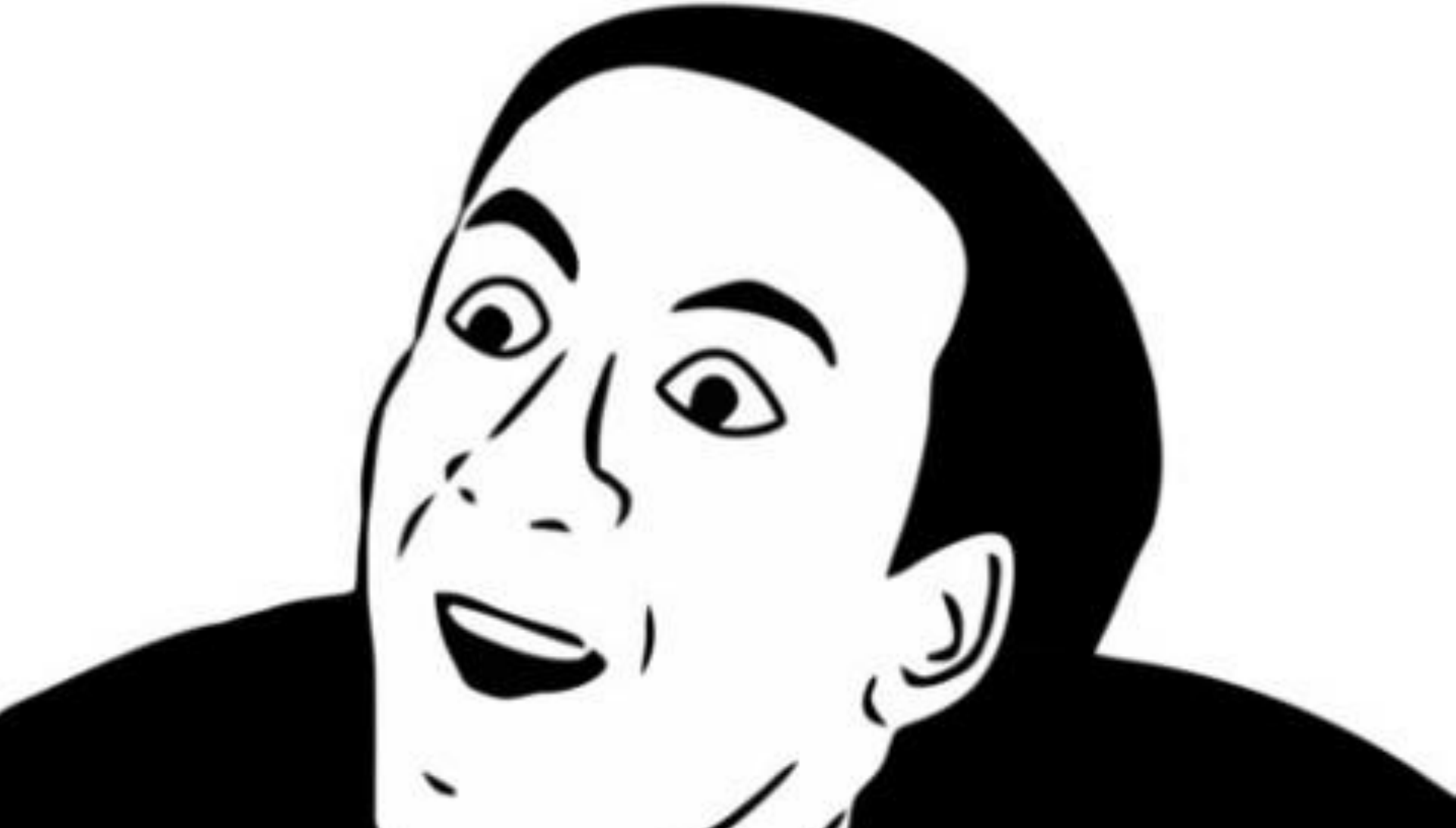
```
1  namespace StupidComments
2  {
3      public class Customer
4      {
5          /// <summary>
6          /// Gets or sets the Customer's Name
7          /// </summary>
8          public string Name { get; set; }
9      }
10 }
11
```

YOU DON'T SAY?



```
/* ------------------------------------------------ */
/* ------------------------------------------------ */
/* ------------------------------------------------ */
//
// main function
//
int main(int argc, char **argv) {
    // initialize gtk and set up gtkbuilder
    // for UI import from xml
    gtk_init ( &argc, &argv );
}
```

YOU DON'T SAY?



```
void printOwning(double amount)
{
      printBanner();

      // print details
      Console.WriteLine(“name” + _name);
      Console.WriteLine(“amount” + amount);
}
```

```
void printOwning(double amount)
{
      printBanner();
      printDetails(amount);
}

void printDetails(double amount)
{
      Console.WriteLine(“name” + _name);
      Console.WriteLine(“amount” + amount);
}
```

```
//50er Maschinen
//Maschinen kaufen
if (shop.Machines50 > lastRound.Machines50) {
    shop.Account -= (10000 * (shop.Machines50 -
lastRound.Machines50));
}

//Maschinen verkaufen
else if (shop.Machines50 < lastRound.Machines50) {
    var damage = shop.Capacity / shop.MaximalCapacity;
    shop.Account += (damage * 8000 *
(lastRound.Machines50shop.Machines50));
}
```

```
var maschinen_kaufen = function (shop, lastRound) {
       shop.Account -= (10000 * (shop.Machines50 -
lastRound.Machines50]));
}

var maschinen_verkaufen = function (shop) {
       var damage = shop.Capacity / shop.MaximalCapacity;
     shop.Account += (damage * 8000 * (lastRound.Machines50]
                             - shop.Machines50));
};

if (shop.Machines50 > lastRound.Machines50]) {
     maschinen_kaufen(shop, lastRound);
}
else if (shop.Machines50 < lastRound.Machines50]) {
     maschinen_verkaufen(shop);
}
```

```
var calculateFormulas = function(input, lastRound, month) {
        var shop = deepCopy(input);
        Procurement(shop, constants.MaximumCapacity);
        Purchase(shop);
        Manufacturing(shop, constants, month);
        Expenses(shop);
        Retail(shop);
        Advertising(shop, constants, month.Demand());
        Banking(shop, constants);
        Reporting(shop, month.month, month.MaterialPrice());
        return shop;
};
```

RENAME METHOD

```
public class CustomerRepository
{
    IEnumerable<Customer>
    GetCustomersByYearOfBirth(DateTime yearOfBirth)
    {

    }
}
```

```
public class CustomerRepository
{
    IEnumerable<Customer>
    GetByYearOfBirth(DateTime yearOfBirth)
    {

    }
}
```

```
public class Customers
{
    IEnumerable<Customer>
    GetByYearOfBirth(DateTime yearOfBirth)
    {

    }
}
```

```
public class Customers
{
    IEnumerable<Customer>
    BornIn(DateTime yearOfBirth)
    {

    }
}
```

```
public interface ICustomerRepository
{
      IEnumerable <Customer> BornIn(DateTime yearOfBirth);
}

public class Customers : ICustomerRepository
{
      /*…*/
}
```

```
public class Customers : ICustomerRepository
{
        /*…*/
}
```

```
public interface IFindCustomers
{
     IEnumerable <Customer> BornIn(DateTime yearOfBirth);
}
```

```
public interface IFindCustomers
{
     IEnumerable <Customer> BornIn(DateTime year);
}
```

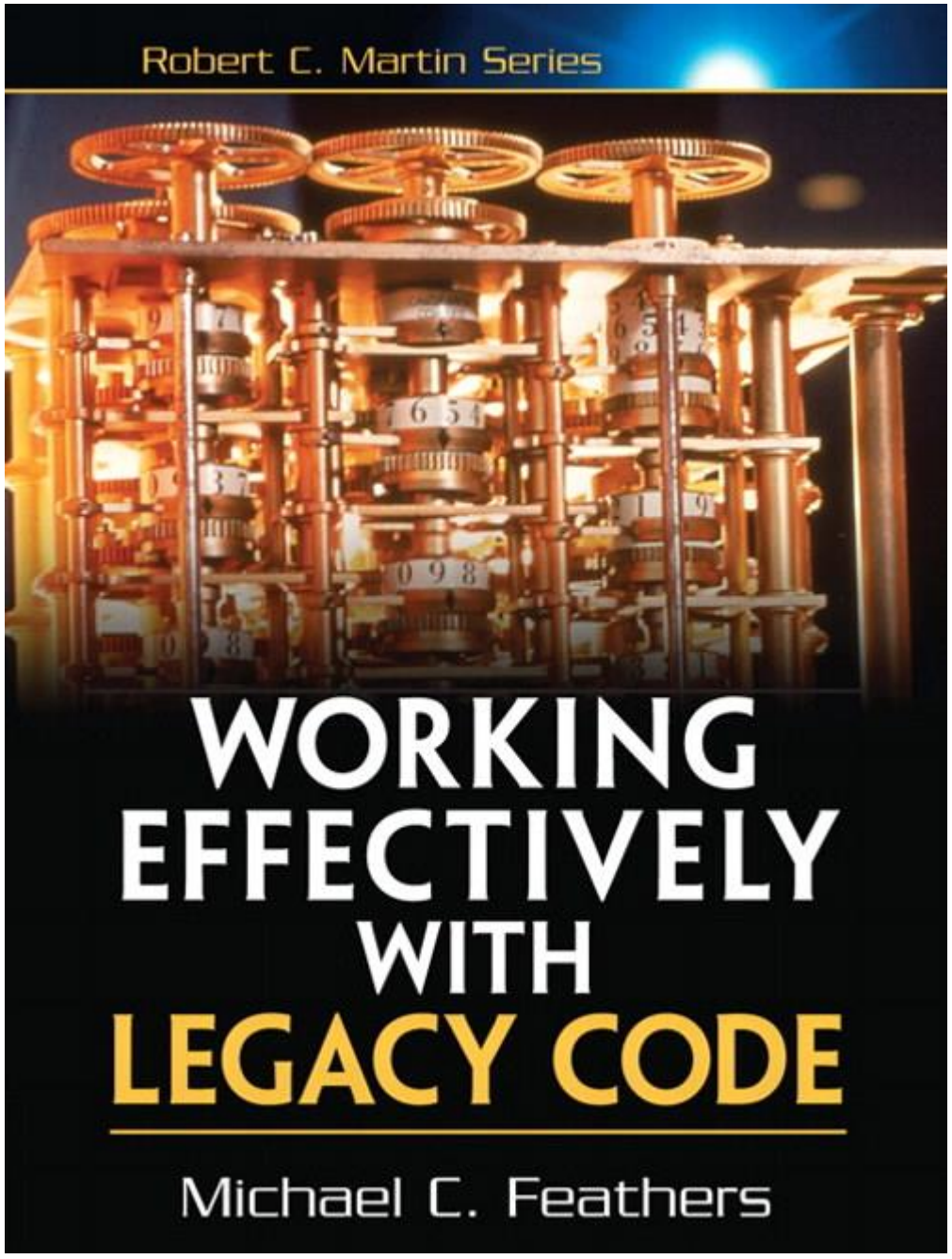
```
public interface IFindCustomers
{
    IEnumerable <Customer> BornIn(Year year);
}
```

R#: F2

Working effectively with

LEGACY CODE

by Michael C. Feathers


Robert C. Martin Series
WORKING EFFECTIVELY WITH LEGACY CODE
Michael C. Feathers

There are many powerful refactorings, but Rename Class is the most powerful. It changes the way people see code and lets them notice possibilities that they might not have considered before.

There are many powerful refactorings, but Rename Class is the most powerful. It changes the way people see code and lets them notice possibilities that they might not have considered before.

There are many powerful refactorings, but **Rename Class is the most powerful**. It changes the way people see code and lets them notice possibilities that they might not have considered before.

There are many powerful refactorings, but Rename Class is the most powerful. It changes the way people see code and lets them notice possibilities that they might not have considered before.

EXTRACT METHOD

EXTRACT CLASS

Psychology!

CHUNKING!

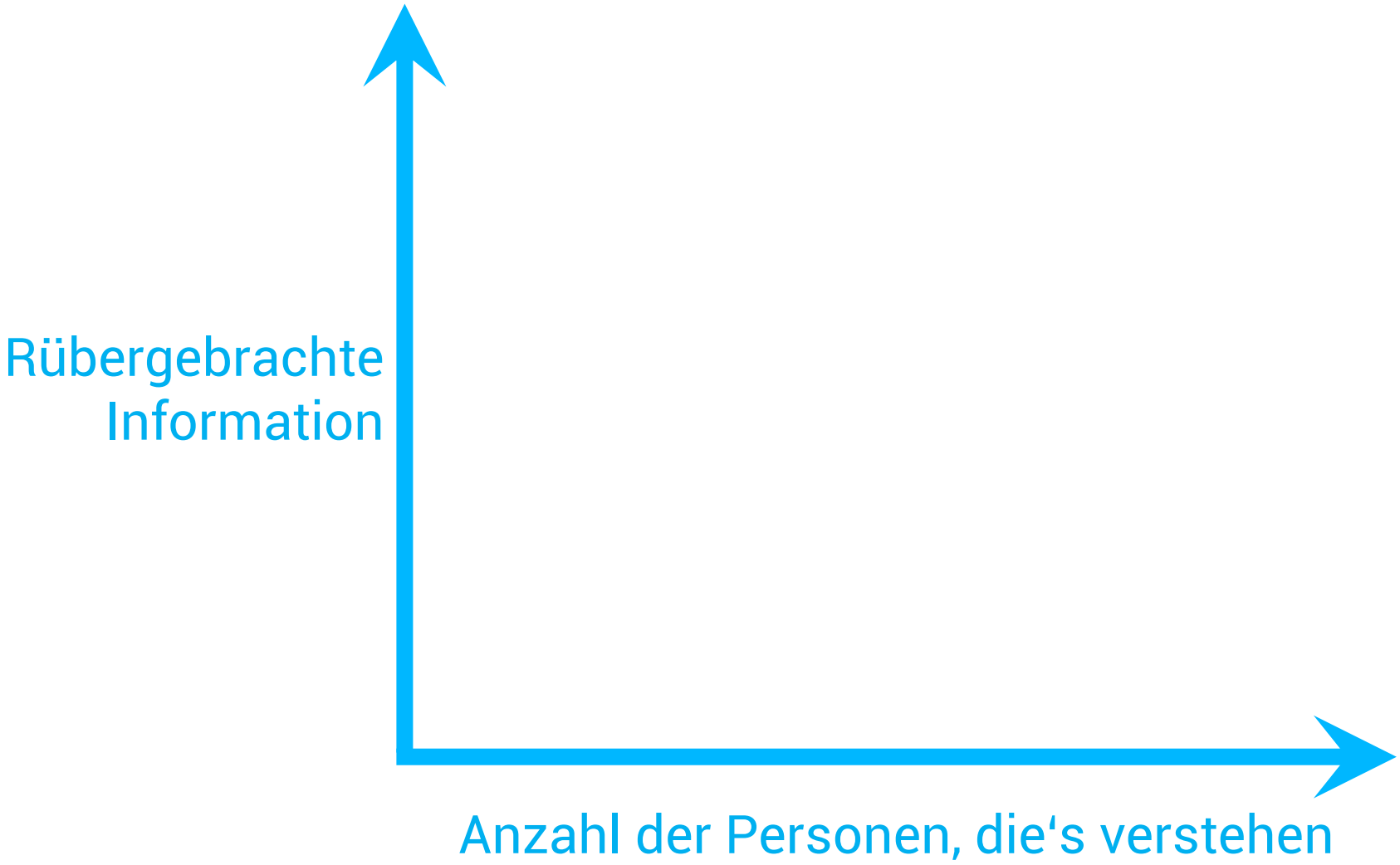
*F*** yeah!*

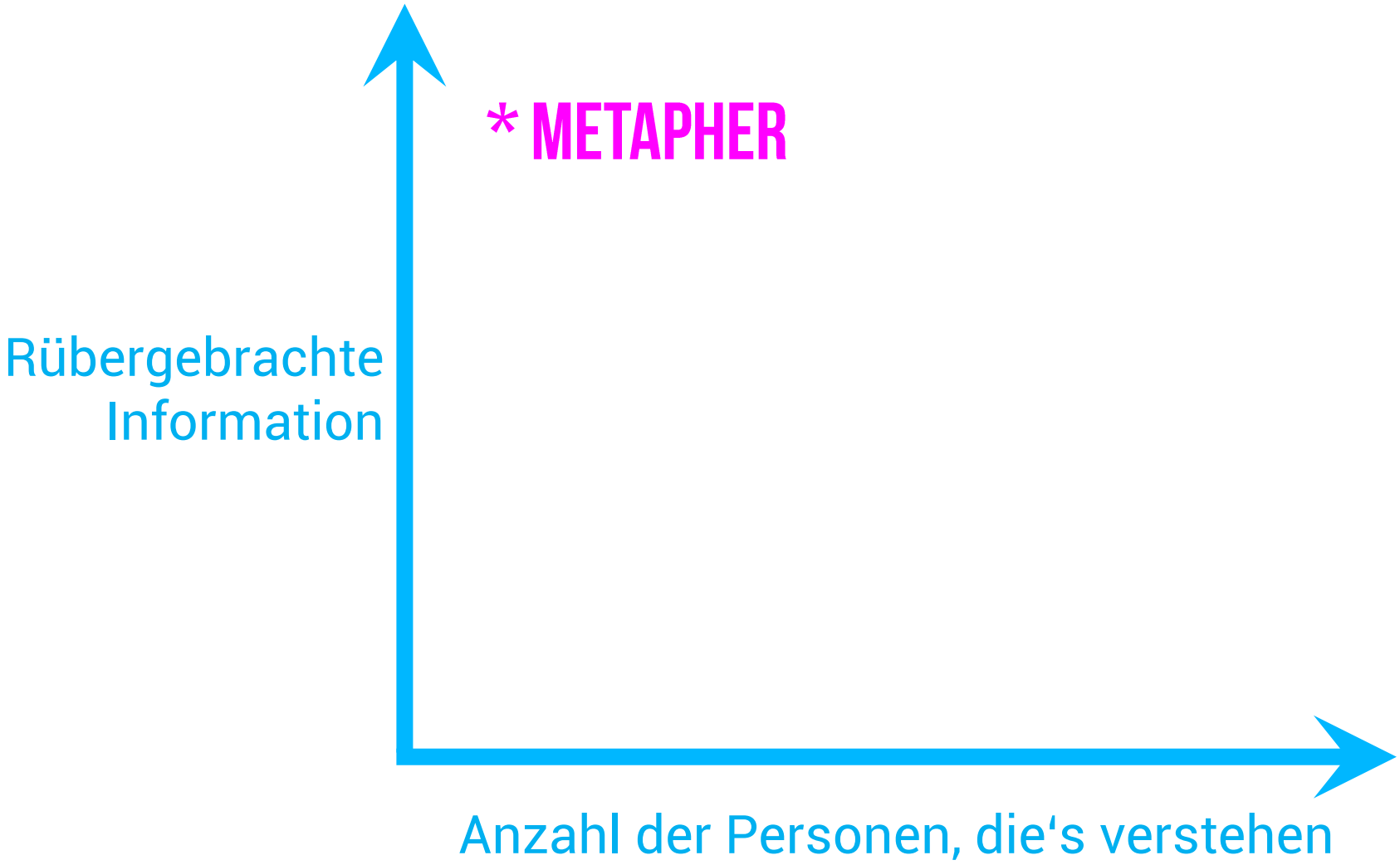
REPLACE CONDITIONAL WITH POLYMORPHISM

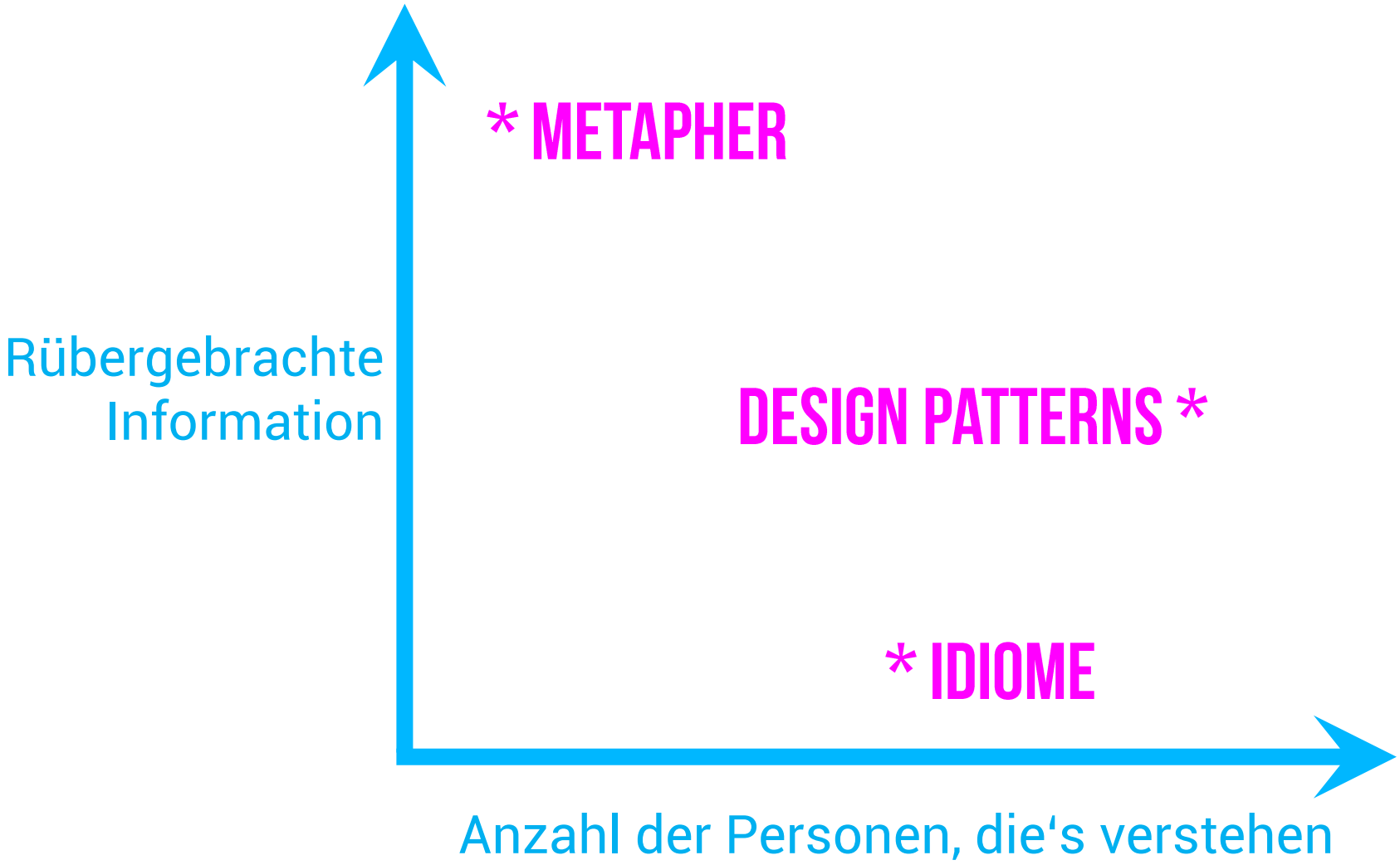
CODE...

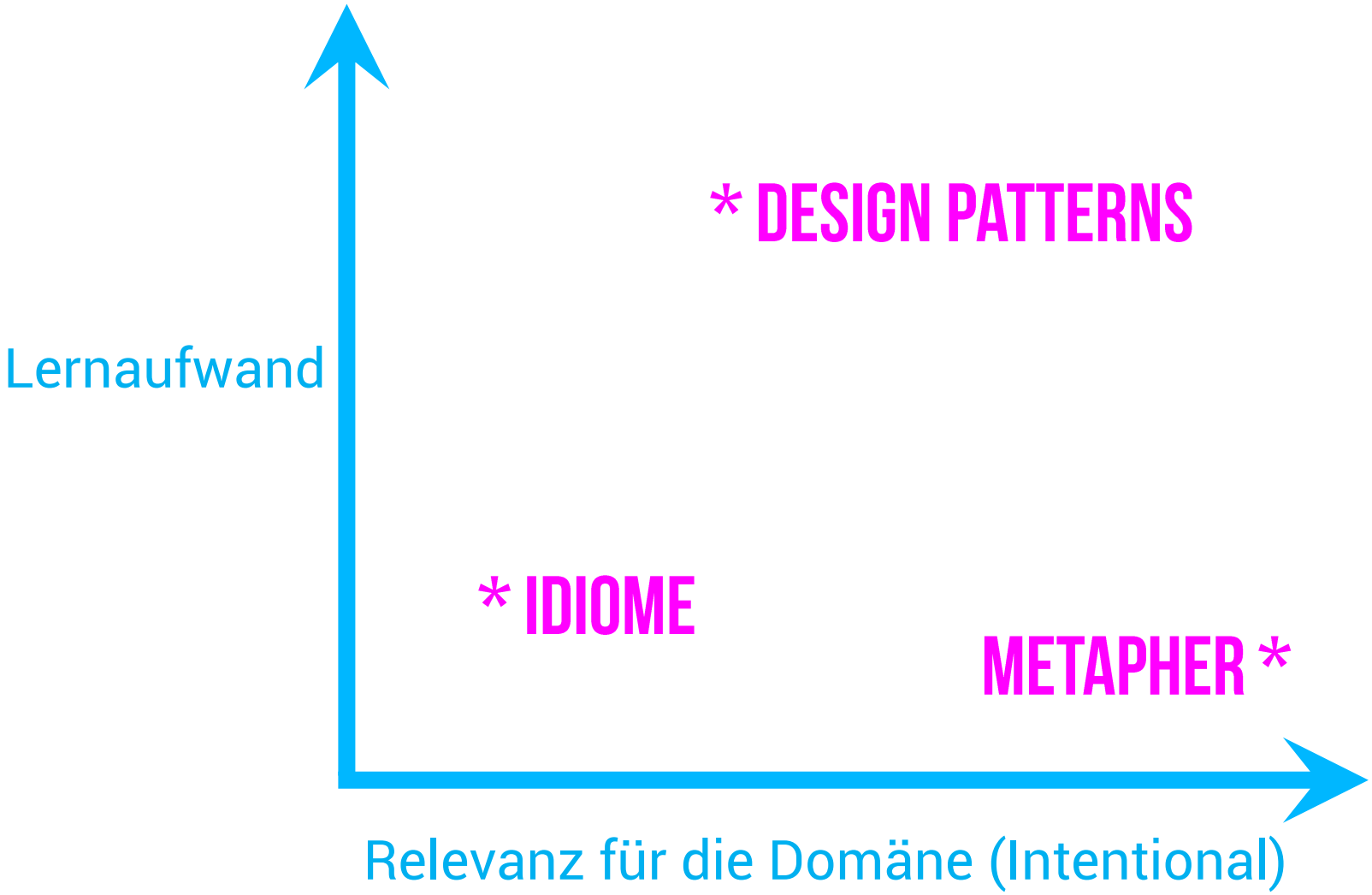
Visual Studio

SYSTEMISCHE METAPHER


Rübergebrachte Information
Anzahl der Personen, die‘s verstehen


* METAPHER
Rübergebrachte Information
Anzahl der Personen, die‘s verstehen


* METAPHER
Rübergebrachte Information
DESIGN PATTERNS *
* IDIOME
Anzahl der Personen, die‘s verstehen


* DESIGN PATTERNS
Lernaufwand
* IDIOME
METAPHER *
Relevanz für die Domäne (Intentional)

DAS IST WIE WENN...

CODE...

Signature Survey & Intent.js

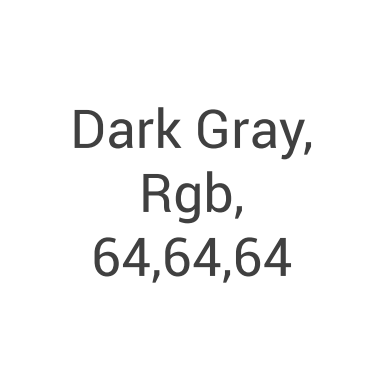

Blue
Rgb,
0,183,255


Orange,
Rgb,
255,116,0


Green,
Rgb
0,219,0


Magenta,
#FF00FF


Light Gray,
Rgb,
191,191,191


Dark Gray,
Rgb,
64,64,64

The blind men and the elephant
http://en.wikisource.org/wiki/The_poems_of_John_Godfrey_Saxe/The_Blind_Men_and_the_Elephant

Elephant
http://inquiry111westminster.wikispaces.com/Blind%20men%20and%20an%20elephant

Inspired by and using the fonts suggested at
http://www.labnol.org/software/tutorials/advice-select-best-fonts-for-powerpoint-presentation-slides/3355/

Duck Duck Duck
http://geekandpoke.typepad.com/geekandpoke/2012/03/static-typing.html

Rapist
http://rasmussenanders.blogspot.de/2011/03/catholic-priests-raping-nuns.html

Bundeswehr
http://www.bmlv.gv.at/download_archiv/photos/inlandseinsatz/images/hochwasser_august_26.jpg

Complaints
http://wayne.usschesapeake.org/wp-content/uploads/2011/06/Shout.png

Apologies
http://www.5lovelanguages.com/learn-the-languages/the-five-languages-of-apology/

Signature Survey
http://c2.com/doc/SignatureSurvey/