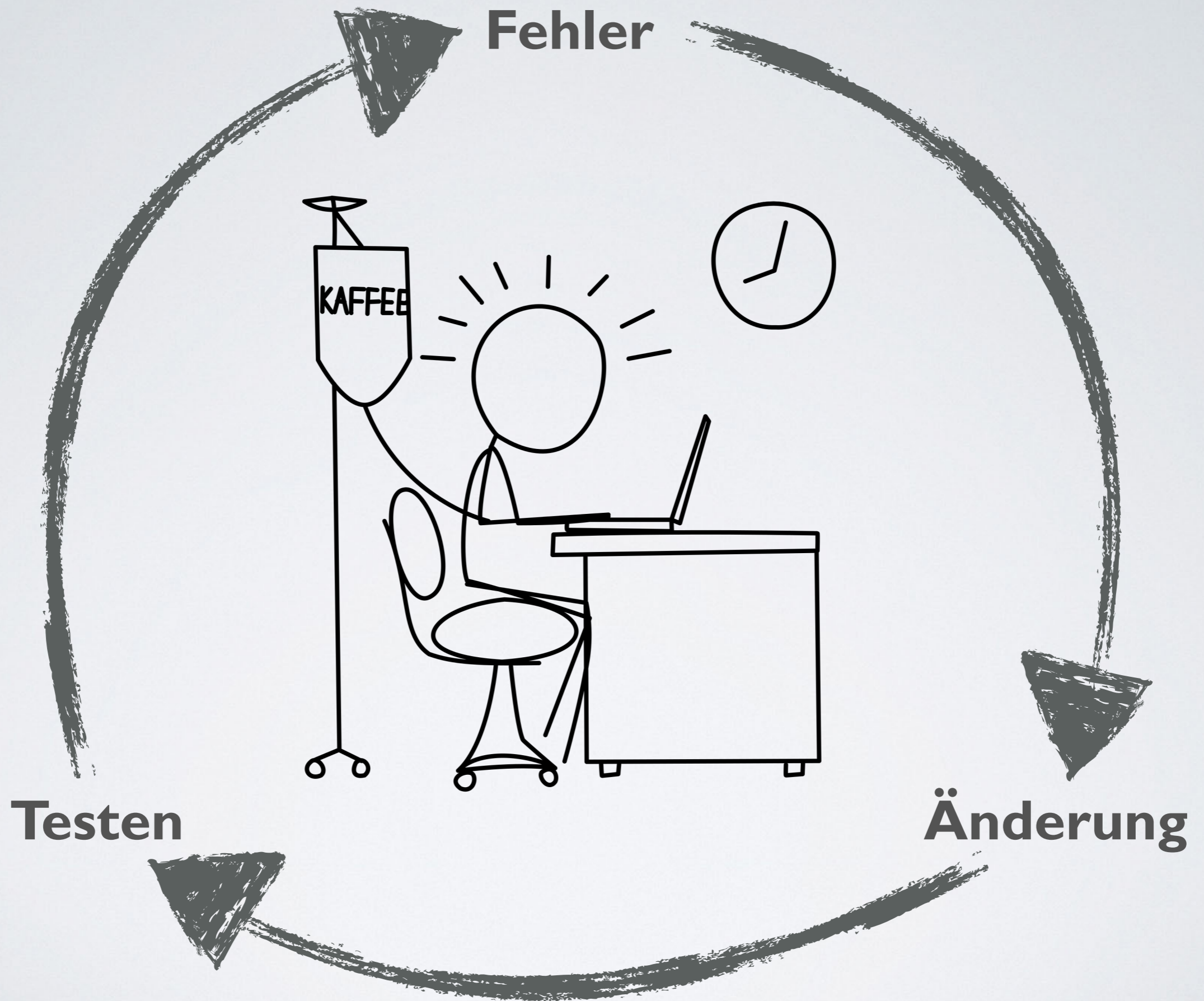


WENN AFFEN TESTEN

DAS ENDE DER BANANENSOFTWARE!

Dr. Jeremias Rößler



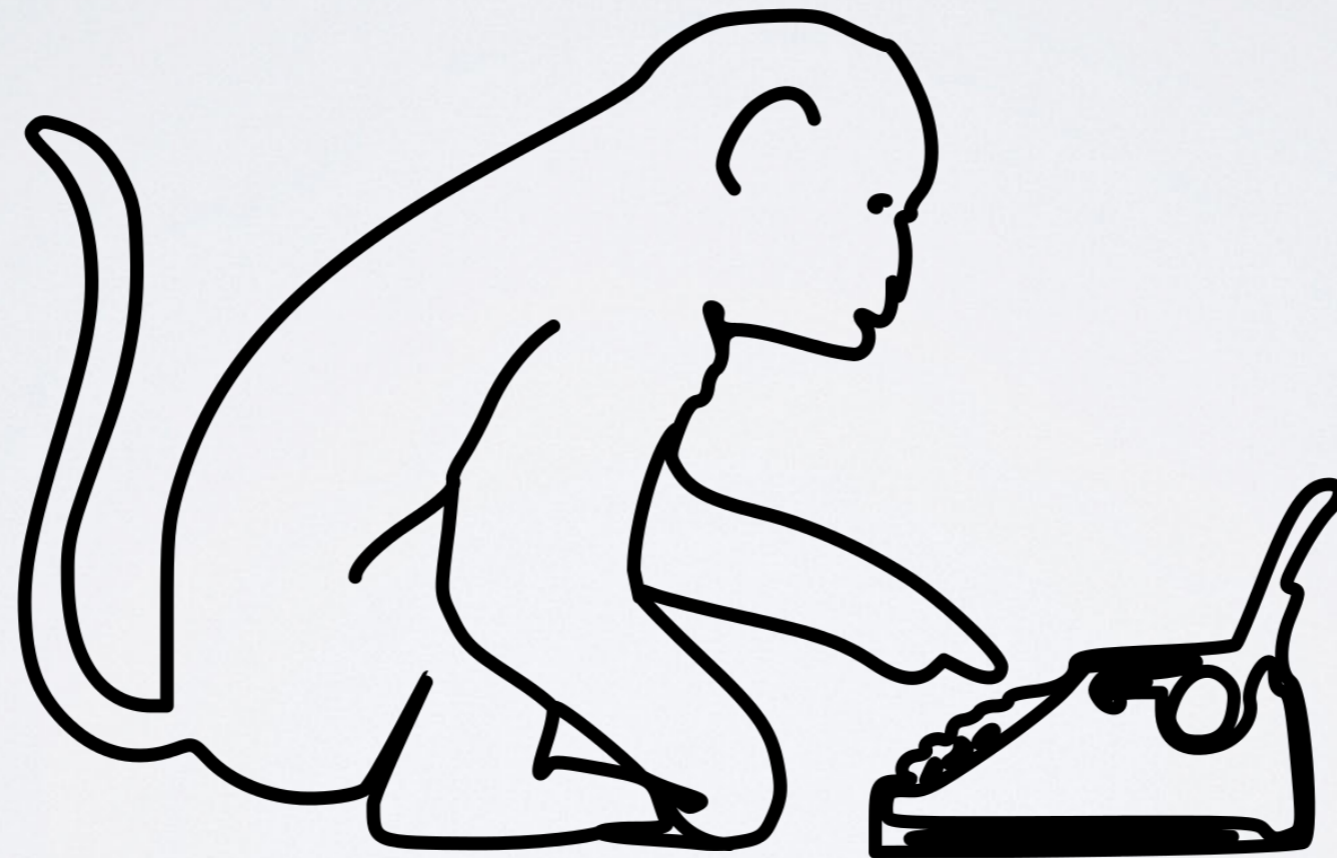
keine Automatisierung

**automatische
Automatisierung!**

keine Zeit



**Infinite-Monkey-Theorem:
Wenn eine Affe nur lange genug auf einer
Schreibmaschine tippt,
schreibt er irgendwann alle Werke von Shakespeare.**



**Wir ersetzen die Schreibmaschine
mit einem Computer..**



Infinite Monkey

```
1. public static void main(String... args) throws Exception {
2.     Robot robot = new Robot();
3.     while (true) {
4.         robot.mouseMove(random.nextInt(maxX), random.nextInt(maxY));
5.         robot.mousePress(InputEvent.BUTTON1_DOWN_MASK);
6.         robot.mouseRelease(InputEvent.BUTTON1_DOWN_MASK);
7.         robot.delay(200);
8.         for (char inputChar : randomString().toCharArray()) {
9.             robot.keyPress((int) inputChar);
10.            robot.keyRelease((int) inputChar);
11.            robot.delay(10);
12.        }
13.        robot.keyPress(KeyEvent.VK_ENTER);
14.        robot.keyRelease(KeyEvent.VK_ENTER);
15.    }
16. }
```

Infinite Monkey

DEMO

Workbook2

Search in Sheet

Home Layout Tables Charts SmartArt Formulas Data Review

Edit Font Alignment Number Format Cells Themes

Paste Calibri (Body) 12 B I U Conditional Formatting Styles Actions Themes

A1 Arbeitspakete

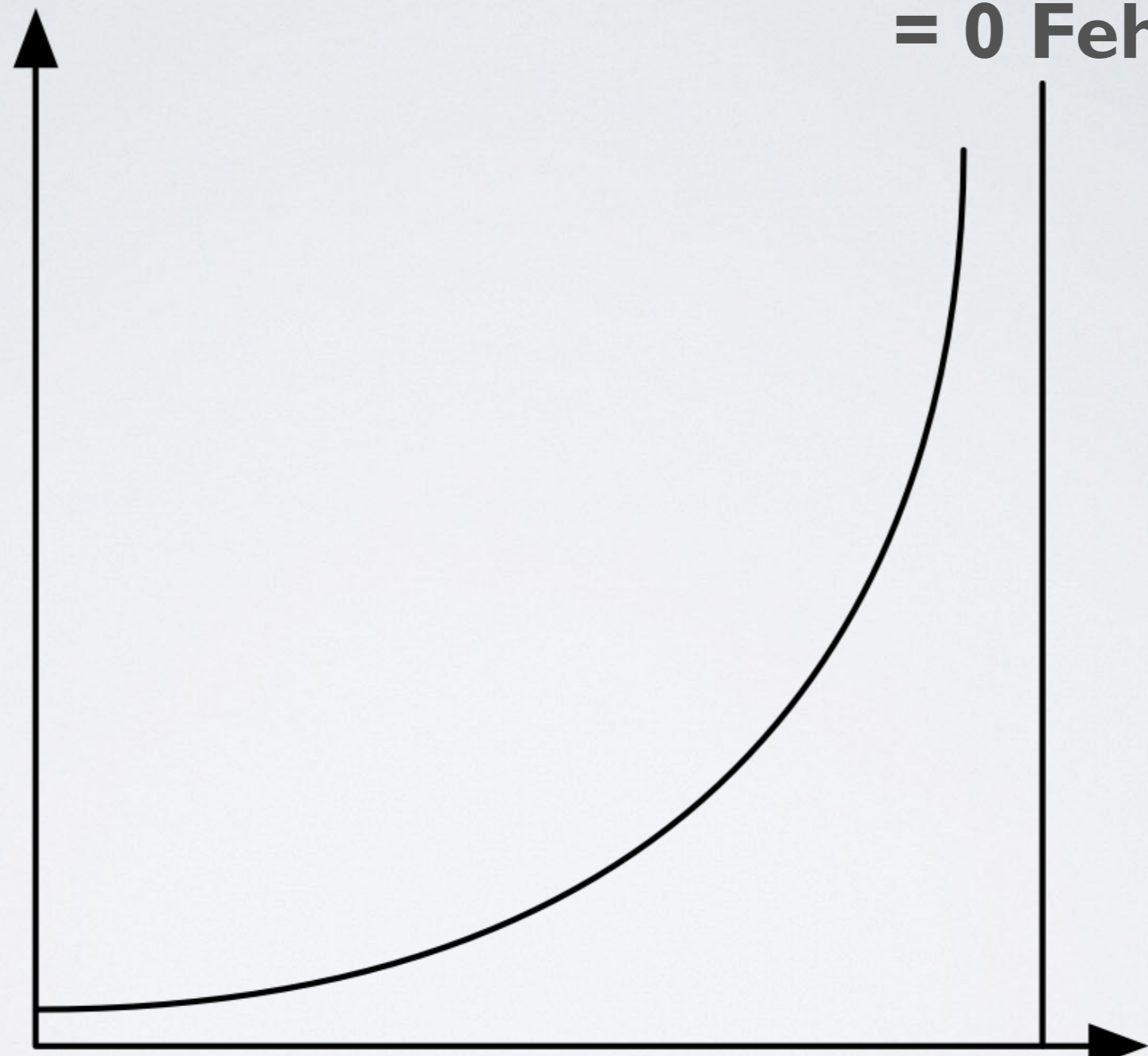
	A	B	C	D	E	F	G	H	I	J
1	Arbeitspakete									
2	Einschalten									
3	Ausschalten									
4	Umschalten				Ich					
5	Abschalten				Du					
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										

Sheet1 +

Normal View Ready



**Aufwand
pro Fehler**



**100%
= 0 Fehler**

Qualität

Dummer Affe



Intelligenter Affe



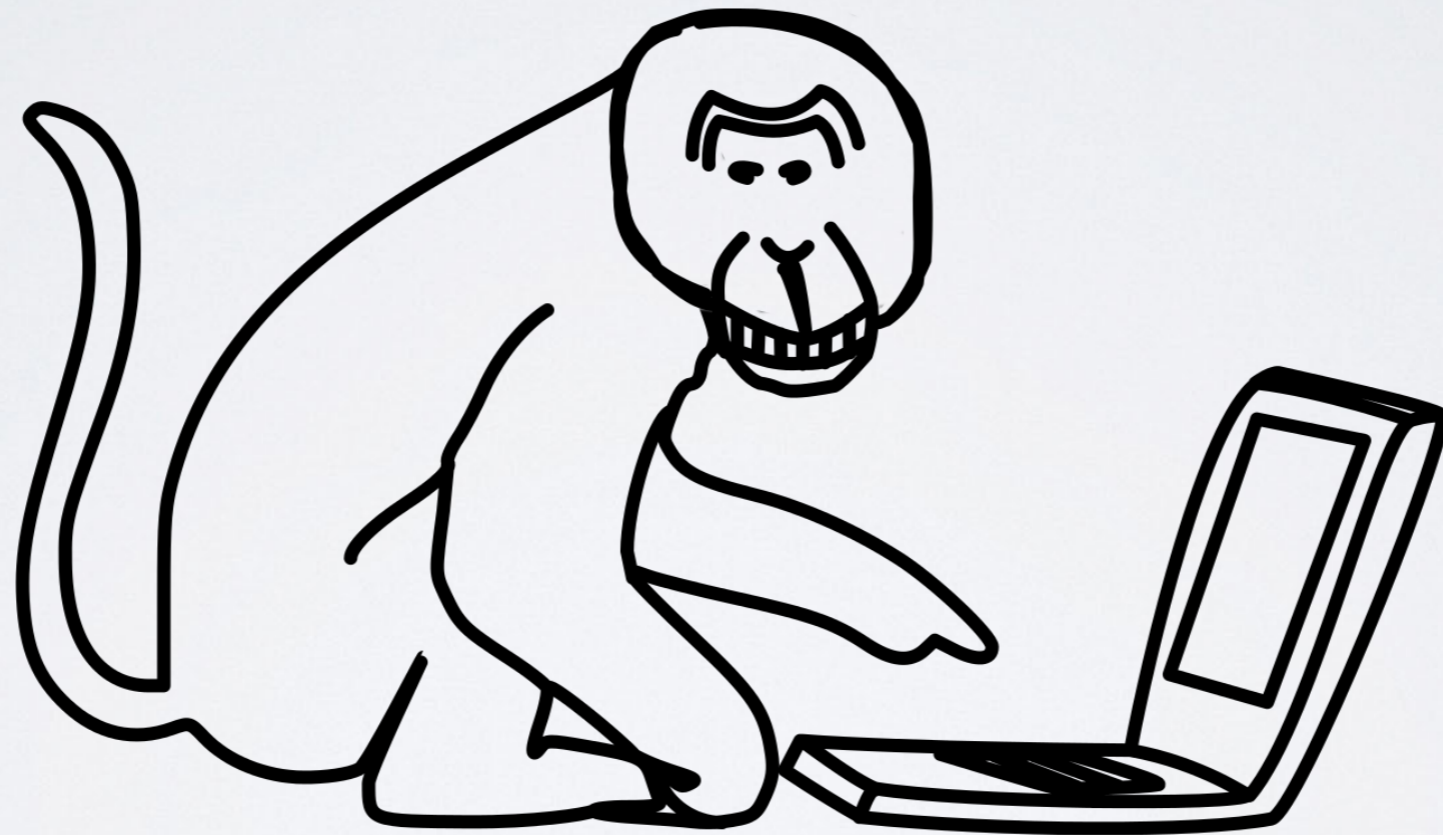
Intelligenter Affe

```
1. public static void main(String... args) throws Exception {
2.     WebDriver driver = new FirefoxDriver();
3.     driver.get("http://www.retest.de");
4.     while (true) {
5.         List<WebElement> links = driver.findElements(By.tagName("a"));
6.         links.get(random.nextInt(links.size())).click();
7.         Thread.sleep(500);
8.         List<WebElement> fields =
9.             driver.findElements(By.xpath("//input[@type='text']"));
10.        WebElement field = fields.get(random.nextInt(fields.size()));
11.        field.sendKeys(randomString());
12.        Thread.sleep(500);
13.    }
14. }
```

Intelligenter Affe

DEMO

Intelligenter Affe



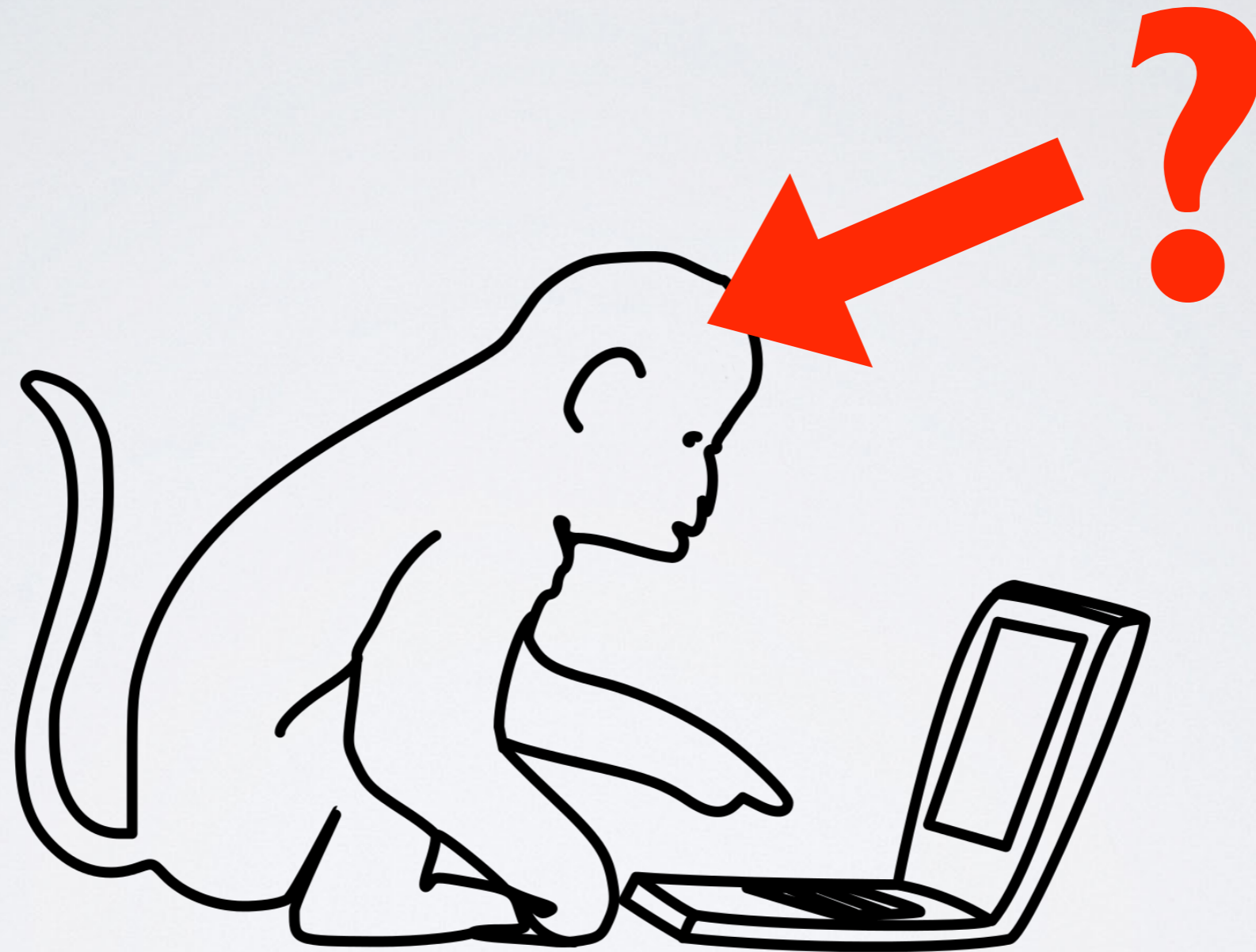
Open Source Produkte

NetFlix Chaos Monkey

gremlins.js

UI/Application Exerciser Monkey

Intelligenter Affe



HumiCalc with Uncertainty

File Options Help

Configuration

Temperature Scale: ITS-90 Carrier Gas: Dry Air Mode: Normal

Equilibrium Over: Water Apply Enhancement Factors: Known: Dew Point

Known Values (Standard u)

Dew Point: 10.0 ±0.0577

Temperature: 25.0 ±0.025

Pressure: 14.7 ±0.025

Calculate

Units

Temperature: °C Pressure: psia Vapor Pressure: Pa Density and Abs Humidity: g/m³ Enthalpy: J/g

Calculated Values (Expanded U with 95.45% Confidence)

%RH	38.7340793	±0.3212	Specific Humidity	0.0076028	±6E-005
Frost Point			Absolute Humidity	8.962178468	±0.0694
Dew Point	10.0	±0.1155	Dry Air Density	1169.837556	±4.0342
PPMv	12314.00364	±105.31	Moist Air Density	1178.799735	±4.033
PPMw	7661.045263	±65.521	Saturation Temperature		
Grains/lb	53.62731684	±0.4586	Saturation Pressure		
Enthalpy	44.63021277	±0.1744	Wet Bulb Temperature	16.10904106	±0.0559
SVP@Tt	3169.90395	±9.4493	Mixing Ratio by Volume	0.012314004	±0.0001
SVP@Td	1228.13339	±9.5028	Mixing Ratio by Weight	0.007661045	±7E-005
SVP@Ts			Percent by Volume	1.216421347	±0.0103
F@Tt,Pt	1.004109348	±1E-005	Percent by Weight	0.760279987	±0.0065
F@Td,Pt	1.003863842	±1E-005	Vapor Mole Fraction	0.012164213	±0.0001
F@Ts,Ps			Dry Air Mole Fraction	0.987835787	±0.0001

Jahreszeugnisse RBE

Name, Vorname: Mastmann, Emil

Kapiteln: Verhalten: 2, Mitarbeit: 2

Leistungen in den einzelnen Fächern u. Fächerverbänden:

EPK:	4	Mat.	4
Deutsch:	4	Physik:	4,3
Englisch:	4	Chemie:	3,9
Mathematik:	4	Biologie:	3,7
Geschichte:	0	Sport:	3
EWG:	3	Musik:	4
Erkunde:		Bildende Kunst:	4
Gemeinschaftslehre:		Technik:	3
TOP Informatik:	2	TOP WIR:	0
TOP Tech. Arbeiten:		TOP Sozial. Engagement:	2

Arbeitsgemeinschaften: 0

Fehltag: 5 unentschuldig., 0

Bemerkungen: Emil war für die Klassenkasse zuständig.

Anwahl: Erster Schüler: 4 (Schüler Nr. 6 von 25), Letzter Schüler: Mastmann, Emil (3,4)

Lagerverwaltung WN 2000

Titel: Lager-Liste

Suchen: Schnellsuche

Artikelnr.: ALU00700

Spezifikation: Dichte: 10,000 in/mm

Material: Profi

Eigenschaften: Umrechnung: 1,0000

Legenart: Durchschl.

Bestell Dem: Bille Eisen

Bemerkung: Spezial-Alu für Demonstrator

S.F.K. GmbH 2003 - Geschäftsjahr 10-2003 - Buchzeitraum 01.01.2003 - 31.10.2003

Buchen: Sachkonten Allgemeinbeleg

Konto: 6200

Debitoren: 5210

OP Anz.Nr.: 03.223

Buchungsdatum: 01.10.03

Belegdatum: 01.10.03

Fälligkeitsbelegung: 1 (30 Tage Netto)

Umsatz: 2.320,00

Steuerbetrag: 320,00

Konto	KM	SKN	SKZ	Beleg	Datum	Soll	Haben	Text	Faktura	Anz.Nr.
1610		6700		147	13.10.2003	30,60		Gewerbesteuer Rückz. 2001		147
1610		4895		147	13.10.2003		500,00	Speisen		147
1610		4915	1	146	09.10.2003		107,70	Telefon		146
1610		0		145	08.10.2003		1,875,00	EP-Jobba		145
1610		4270		144	06.10.2003		369,38	144 09.03		144
1610		4830		144	06.10.2003		966,44	144 9.03		144
1610		4895		143	02.10.2003		20,50	OP		143
1610		4895		143	02.10.2003		500,00	Erhaltung		143

Status: Alle Buchungen eingetrag. in Arbeit

Summe Primarsalden: 7.260,30

S.F.K. GmbH 2003 - Geschäftsjahr 10-2003 - Buchzeitraum 01.01.2003 - 31.10.2003

Buchen: Sachkonten Allgemeinbeleg

Konto: 6200

Debitoren: 5210

OP Anz.Nr.: 03.223

Buchungsdatum: 01.10.03

Belegdatum: 01.10.03

Fälligkeitsbelegung: 1 (30 Tage Netto)

Umsatz: 2.320,00

Steuerbetrag: 320,00

Konto	KM	SKN	SKZ	Beleg	Datum	Soll	Haben	Text	Faktura	Anz.Nr.
1610		6700		147	13.10.2003	30,60		Gewerbesteuer Rückz. 2001		147
1610		4895		147	13.10.2003		500,00	Speisen		147
1610		4915	1	146	09.10.2003		107,70	Telefon		146
1610		0		145	08.10.2003		1,875,00	EP-Jobba		145
1610		4270		144	06.10.2003		369,38	144 09.03		144
1610		4830		144	06.10.2003		966,44	144 9.03		144
1610		4895		143	02.10.2003		20,50	OP		143
1610		4895		143	02.10.2003		500,00	Erhaltung		143

Status: Alle Buchungen eingetrag. in Arbeit

Summe Primarsalden: 7.260,30

Lagerverwaltung WN 2000

Titel: Lager-Liste

Suchen: Schnellsuche

Artikelnr.: ALU00700

Spezifikation: Dichte: 10,000 in/mm

Material: Profi

Eigenschaften: Umrechnung: 1,0000

Legenart: Durchschl.

Bestell Dem: Bille Eisen

Bemerkung: Spezial-Alu für Demonstrator

Jahreszeugnisse RBE

Name, Vorname: Mastmann, Emil

Kapiteln: Verhalten: 2, Mitarbeit: 2

Leistungen in den einzelnen Fächern u. Fächerverbänden:

EPK:	4	Mat.	4
Deutsch:	4	Physik:	4,3
Englisch:	4	Chemie:	3,9
Mathematik:	4	Biologie:	3,7
Geschichte:	0	Sport:	3
EWG:	3	Musik:	4
Erkunde:		Bildende Kunst:	4
Gemeinschaftslehre:		Technik:	3
TOP Informatik:	2	TOP WIR:	0
TOP Tech. Arbeiten:		TOP Sozial. Engagement:	2

Arbeitsgemeinschaften: 0

Fehltag: 5 unentschuldig., 0

Bemerkungen: Emil war für die Klassenkasse zuständig.

Anwahl: Erster Schüler: 4 (Schüler Nr. 6 von 25), Letzter Schüler: Mastmann, Emil (3,4)

S.F.K. GmbH 2003 - Geschäftsjahr 10-2003 - Buchzeitraum 01.01.2003 - 31.10.2003

Buchen: Sachkonten Allgemeinbeleg

Konto: 6200

Debitoren: 5210

OP Anz.Nr.: 03.223

Buchungsdatum: 01.10.03

Belegdatum: 01.10.03

Fälligkeitsbelegung: 1 (30 Tage Netto)

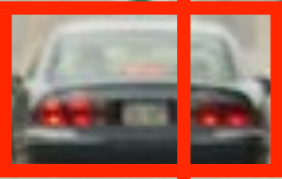
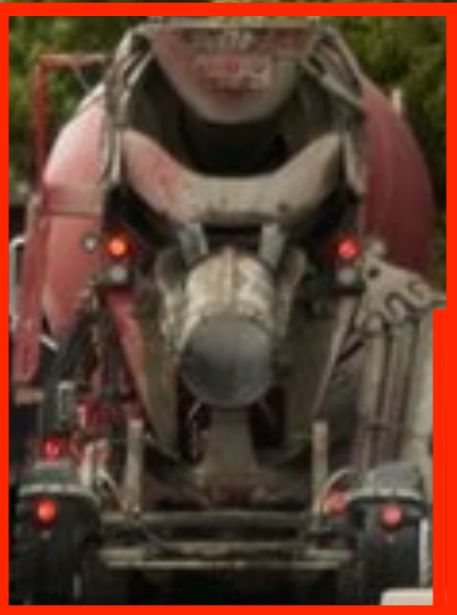
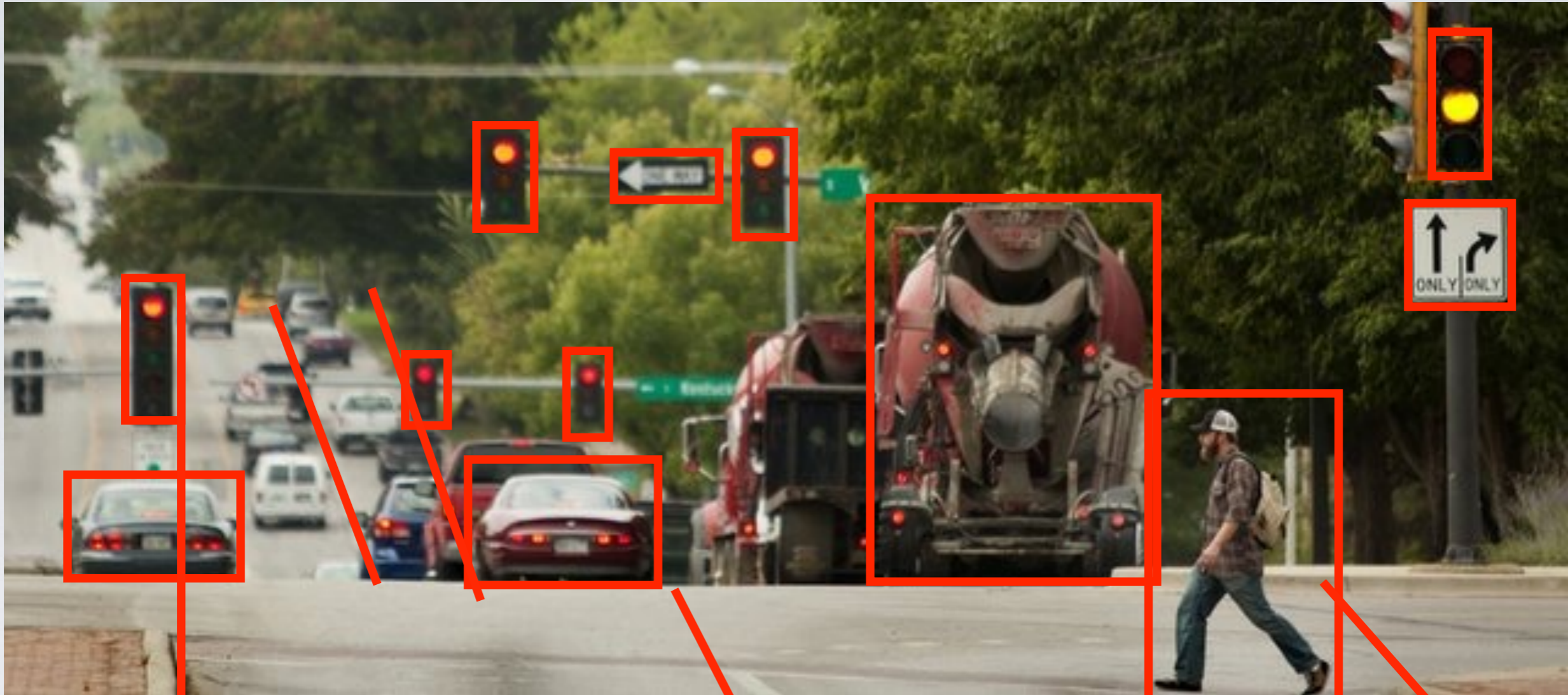
Umsatz: 2.320,00

Steuerbetrag: 320,00

Konto	KM	SKN	SKZ	Beleg	Datum	Soll	Haben	Text	Faktura	Anz.Nr.
1610		6700		147	13.10.2003	30,60		Gewerbesteuer Rückz. 2001		147
1610		4895		147	13.10.2003		500,00	Speisen		147
1610		4915	1	146	09.10.2003		107,70	Telefon		146
1610		0		145	08.10.2003		1,875,00	EP-Jobba		145
1610		4270		144	06.10.2003		369,38	144 09.03		144
1610		4830		144	06.10.2003		966,44	144 9.03		144
1610		4895		143	02.10.2003		20,50	OP		143
1610		4895		143	02.10.2003		500,00	Erhaltung		143

Status: Alle Buchungen eingetrag. in Arbeit

Summe Primarsalden: 7.260,30



Problemraum

**Problem
raum**





Jahreszeugnisse RBE

Name, Vorname: **Mustermann** Emil

Kapitelnummer: 2 Mitarbeit: 2

Leistungen in den einzelnen Fächern u. Fächerverbindungen:

EBK	4	Mat.	4
Deutsch	4	Physik	4,3
Englisch	4	Chemie	3,9
Mathematik	4	Biologie	3,7
Geschichte	0	Sport	3
EWG	3	Musik	4
Erdkunde		Bildende Kunst	4
Gemeinschaftskunde		Technik	3
TOP Informatik	2	TOP WIR	0
TOP Tech. Arbeiten		TOP Social. Engagement	2

Arbeitsgemeinschaften: AG Auswahl

Fehlitage: 5 unentschuldig; 0 Fehlbefreiungen; 2 unentschuldig; 0

Bemerkungen: Versetzt: Emil war für die Klassenkasse zuständig.

Verstärkt: Klassensprecher Tagbuchhelfer Klassenkasse Medienteam

Nicht verstärkt: Press. Jugend für Olympia Schachspieler Wertschöpfungsbeauftragter Schließdienst England Frankreich Müttertag Befähigung Ergänzung Bemerkungen

Auswahl: Erster Schüler: 4 | Letzter Schüler: 25

Eingabemodus: Noteneingabe nach Schülern Noteneingabe nach Fächern

Notenübersichtswert: 3,4

Lagerverwaltung WIN 2000

Die Liste: Eingabemaske | Statusinfo | Text/Bezeichnung

Artikel: ALU00700 | Spezifikation | Dicke: 10.000 | in mm | Bestand: 2 | 1.952 | in

Eigenschaften: Zu Abgabe | Lieferanten | Text/Bemerkungen

Umschreibung: 1.0000 | pro | ar | Spez. Dicke: 2.700 | kg/m³ | ar | Grundfläche: 0,0400 | dm² | ar

EK-Preis 1: 0 | 5,52 | €/pro | 0 | VK-Preis 1: 0 | 7,18 |

EK-Preis 2: 0 | 5,96 | €/pro | 0 | VK-Preis 2: 0 | 7,75 |

EK-Wert: 0 | 11,04 | € | VK-Wert: 0 | 14,93 |

Lagerort: Durchwahl | Min. Menge: 2.000 |

Artikel ist Best.: Ja | B. Datum: 05.06.2007 | B. Menge: 10.000 | Kg | Aktive Bestellungen: 1 | St.

Bestell Bem.: Bitte E-Mails |

Bemerkung: Special ALU für Demonstrator

S.F.K. GmbH 2003 - Geschäftsjahr 10.2003 - Buchstamm 01.01.2003 - 31.10.2003

Buchen: Sachkonten Allgemeinbeleg

Deutsche Asphal HOFFMANN | EUR: 0,00 | Erlöse Holze - Wartung | EUR: 142.747,17 H

Konto: 5210 | Eigenkapital | 9 | 16,00% MwSt in 100

Belag: OP Anz.Nr.: 03.223 | Buchungdatum: 01.10.03 | Belegdatum: 01.10.03 | Zahlungsbelegung: 1 (30 Tage Netto)

Text: Vortrag 4. Quartal | Wert: 20.000,00 |

Faktura: 00.00.00 |

Umsatz: 2.320,00 | Soll Haben: 300,00 | Steuerbetrag: 300,00

Konto	KM	SKN	SKN2	SKN3	SKN4	Beleg	Datum	Soll	Haben	Text	Faktura	Anz.Nr.
1610		6700				147	13.10.2003	30,60		Gewerbesteuer Rückz. 2003	147	147
1610		4995				147	13.10.2003		500,00	Speisen	147	147
1610		4915	1	146		146	09.10.2003		107,70	Telefon	146	146
1610		0				145	08.10.2003	1.835,00		10. Probe	145	145
1610		4270				144	06.10.2003		369,00	144 09.03	144	144
1610		4830				144	06.10.2003		966,44	144 9.03	144	144
1610		4995				143	02.10.2003		25,50	OP	143	143
1610		4995				143	02.10.2003		500,00	Erhaltung	143	143
								Summe Primärkonto	7.260,30			

S.F.K. GmbH 2003 - Geschäftsjahr 10.2003 - Buchstamm 01.01.2003 - 31.10.2003

Buchen: Sachkonten Allgemeinbeleg

Deutsche Asphal HOFFMANN | EUR: 0,00 | Erlöse Holze - Wartung | EUR: 142.747,17 H

Konto: 5210 | Eigenkapital | 9 | 16,00% MwSt in 100

Belag: OP Anz.Nr.: 03.223 | Buchungdatum: 01.10.03 | Belegdatum: 01.10.03 | Zahlungsbelegung: 1 (30 Tage Netto)

Text: Vortrag 4. Quartal | Wert: 20.000,00 |

Faktura: 00.00.00 |

Umsatz: 2.320,00 | Soll Haben: 300,00 | Steuerbetrag: 300,00

Konto	KM	SKN	SKN2	SKN3	SKN4	Beleg	Datum	Soll	Haben	Text	Faktura	Anz.Nr.
1610		6700				147	13.10.2003	30,60		Gewerbesteuer Rückz. 2003	147	147
1610		4995				147	13.10.2003		500,00	Speisen	147	147
1610		4915	1	146		146	09.10.2003		107,70	Telefon	146	146
1610		0				145	08.10.2003	1.835,00		10. Probe	145	145
1610		4270				144	06.10.2003		369,00	144 09.03	144	144
1610		4830				144	06.10.2003		966,44	144 9.03	144	144
1610		4995				143	02.10.2003		25,50	OP	143	143
1610		4995				143	02.10.2003		500,00	Erhaltung	143	143
								Summe Primärkonto	7.260,30			

Lagerverwaltung WIN 2000

Die Liste: Eingabemaske | Statusinfo | Text/Bezeichnung

Artikel: ALU00700 | Spezifikation | Dicke: 10.000 | in mm | Bestand: 2 | 1.952 | in

Eigenschaften: Zu Abgabe | Lieferanten | Text/Bemerkungen

Umschreibung: 1.0000 | pro | ar | Spez. Dicke: 2.700 | kg/m³ | ar | Grundfläche: 0,0400 | dm² | ar

EK-Preis 1: 0 | 5,52 | €/pro | 0 | VK-Preis 1: 0 | 7,18 |

EK-Preis 2: 0 | 5,96 | €/pro | 0 | VK-Preis 2: 0 | 7,75 |

EK-Wert: 0 | 11,04 | € | VK-Wert: 0 | 14,93 |

Lagerort: Durchwahl | Min. Menge: 2.000 |

Artikel ist Best.: Ja | B. Datum: 05.06.2007 | B. Menge: 10.000 | Kg | Aktive Bestellungen: 1 | St.

Bestell Bem.: Bitte E-Mails |

Bemerkung: Special ALU für Demonstrator

Jahreszeugnisse RBE

Name, Vorname: **Mustermann** Emil

Kapitelnummer: 2 Mitarbeit: 2

Leistungen in den einzelnen Fächern u. Fächerverbindungen:

EBK	4	Mat.	4
Deutsch	4	Physik	4,3
Englisch	4	Chemie	3,9
Mathematik	4	Biologie	3,7
Geschichte	0	Sport	3
EWG	3	Musik	4
Erdkunde		Bildende Kunst	4
Gemeinschaftskunde		Technik	3
TOP Informatik	2	TOP WIR	0
TOP Tech. Arbeiten		TOP Social. Engagement	2

Arbeitsgemeinschaften: AG Auswahl

Fehlitage: 5 unentschuldig; 0 Fehlbefreiungen; 2 unentschuldig; 0

Bemerkungen: Versetzt: Emil war für die Klassenkasse zuständig.

Verstärkt: Klassensprecher Tagbuchhelfer Klassenkasse Medienteam

Nicht verstärkt: Press. Jugend für Olympia Schachspieler Wertschöpfungsbeauftragter Schließdienst England Frankreich Müttertag Befähigung Ergänzung Bemerkungen

Auswahl: Erster Schüler: 4 | Letzter Schüler: 25

Eingabemodus: Noteneingabe nach Schülern Noteneingabe nach Fächern

Notenübersichtswert: 3,4

S.F.K. GmbH 2003 - Geschäftsjahr 10.2003 - Buchstamm 01.01.2003 - 31.10.2003

Buchen: Sachkonten Allgemeinbeleg

Deutsche Asphal HOFFMANN | EUR: 0,00 | Erlöse Holze - Wartung | EUR: 142.747,17 H

Konto: 5210 | Eigenkapital | 9 | 16,00% MwSt in 100

Belag: OP Anz.Nr.: 03.223 | Buchungdatum: 01.10.03 | Belegdatum: 01.10.03 | Zahlungsbelegung: 1 (30 Tage Netto)

Text: Vortrag 4. Quartal | Wert: 20.000,00 |

Faktura: 00.00.00 |

Umsatz: 2.320,00 | Soll Haben: 300,00 | Steuerbetrag: 300,00

Konto	KM	SKN	SKN2	SKN3	SKN4	Beleg	Datum	Soll	Haben	Text	Faktura	Anz.Nr.
1610		6700				147	13.10.2003	30,60		Gewerbesteuer Rückz. 2003	147	147
1610		4995				147	13.10.2003		500,00	Speisen	147	147
1610		4915	1	146		146	09.10.2003		107,70	Telefon	146	146
1610		0				145	08.10.2003	1.835,00		10. Probe	145	145
1610		4270				144	06.10.2003		369,00	144 09.03	144	144
1610		4830				144	06.10.2003		966,44	144 9.03	144	144
1610		4995				143	02.10.2003		25,50	OP	143	143
1610		4995				143	02.10.2003		500,00	Erhaltung	143	143
								Summe Primärkonto	7.260,30			

___ / ___ / ____



Geburtsdatum

Geburtsdatum ✕

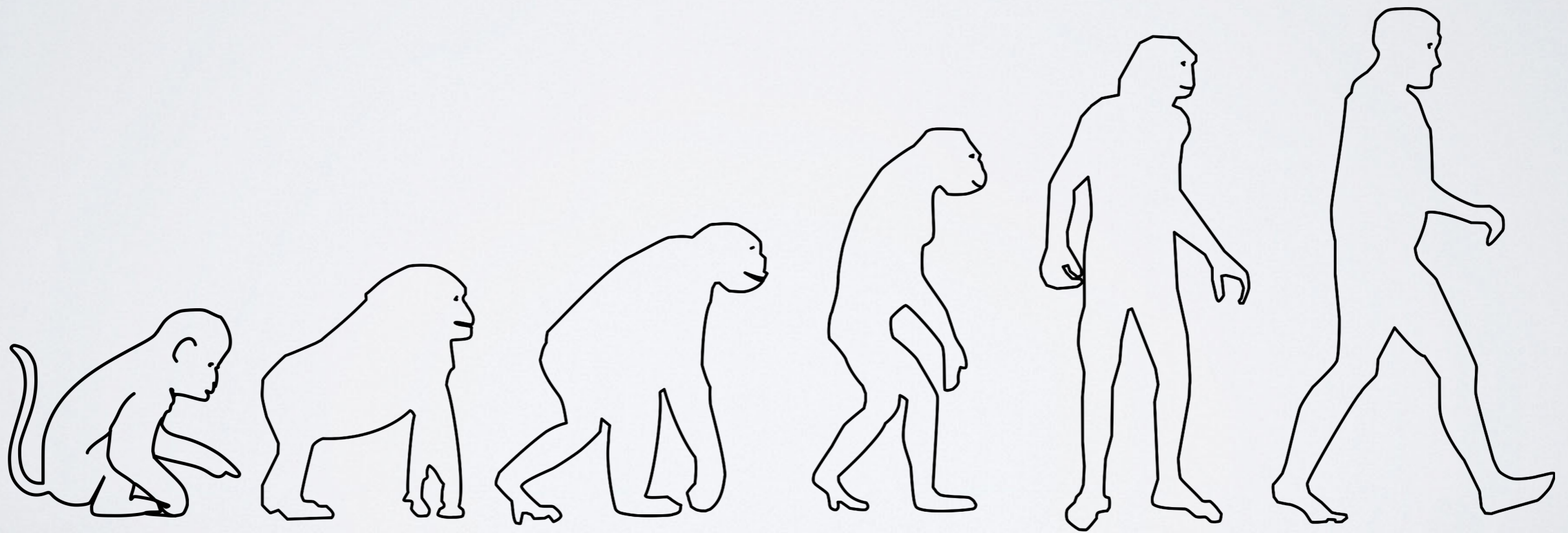
◀ **December 1980** ▶

S	M	T	W	T	F	S
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

Intelligenter Affe



Intelligenter Affe



Türme von Hanoi



© André Karwath aka Aka

Türme von Hanoi

```
1.class TuermeVonHanoi():
2.     def __init__(self):
3.         self.A = [6, 5, 4, 3, 2, 1]
4.         self.B = []
5.         self.C = []
6.
7.     def AtoB(self): self.B.append(self.A.pop())
8.
9.     def AtoC(self): self.C.append(self.A.pop())
10.
11.    def BtoA(self): self.A.append(self.B.pop())
12.
13.    def BtoC(self): self.C.append(self.B.pop())
14.
15.    def CtoA(self): self.A.append(self.C.pop())
16.
17.    def CtoB(self): self.B.append(self.C.pop())
18.
19.    def valid(self):
20.        return all(self.A[i + 1] < self.A[i] for i in range(len(self.A)-1)) and \
21.            all(self.B[m + 1] < self.B[m] for m in range(len(self.B)-1)) and \
22.            all(self.C[n + 1] < self.C[n] for n in range(len(self.C)-1))
```

Türme von Hanoi

Rekursiv

```
1.  def bewege(schritte, i, a_name, b_name, c_name):
2.      if (i > 0):
3.          bewege(schritte, i-1, a_name, c_name, b_name)
4.          schritte.append('tuerme.' + a_name + 'to' + c_name + '()')
5.          bewege(schritte, i-1, b_name, a_name, c_name)
6.      return schritte
7.
8.  def algorithmus():
9.      return bewege([], 6, 'A', 'B', 'C')
```

Türme von Hanoi

Manuell

```
47. 'tuerme.BtoA()' , \ #[43][52][61] -> [432][5][61]
48. 'tuerme.CtoA()' , \ #[432][5][61] -> [4321][5][6]
49. 'tuerme.BtoC()' , \ #[4321][5][6] -> [4321][][65]
50. 'tuerme.AtoB()' , \ #[4321][][65] -> [432][1][65]
51. 'tuerme.AtoC()' , \ #[432][1][65] -> [43][1][652]
52. 'tuerme.BtoC()' , \ #[43][1][652] -> [43][][6521]
53. 'tuerme.AtoB()' , \ #[43][][6521] -> [4][3][6521]
54. 'tuerme.CtoA()' , \ #[4][3][6521] -> [41][3][652]
55. 'tuerme.CtoB()' , \ #[41][3][652] -> [41][32][65]
56. 'tuerme.AtoB()' , \ #[41][32][65] -> [4][321][65]
57. 'tuerme.AtoC()' , \ #[4][321][65] -> [][321][654]
58. 'tuerme.BtoC()' , \ #[][321][654] -> [][32][6541]
59. 'tuerme.BtoA()' , \ #[][32][6541] -> [2][3][6541]
60. 'tuerme.CtoA()' , \ #[2][3][6541] -> [21][3][654]
61. 'tuerme.BtoC()' , \ #[21][3][654] -> [21][][6543]
62. 'tuerme.AtoB()' , \ #[21][][6543] -> [2][1][6543]
63. 'tuerme.AtoC()' , \ #[2][1][6543] -> [][1][65432]
64. 'tuerme.BtoC()' ] \ #[][1][65432] -> [][][654321]
```

Türme von Hanoi Möglichkeiten

$$\geq 2^{63}$$

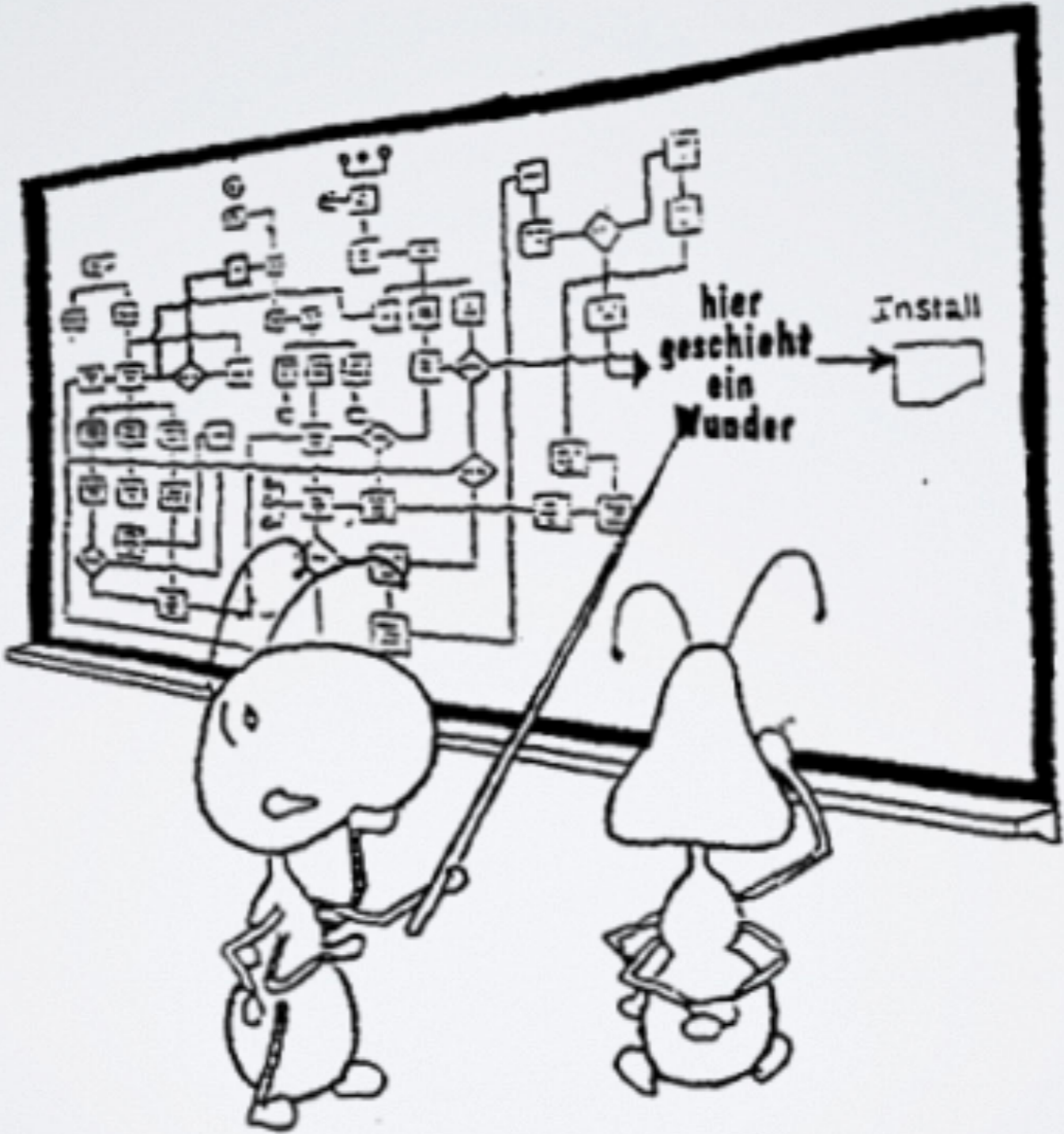
Türme von Hanoi Möglichkeiten

9.223.372.036.854.775.808

Türme von Hanoi Möglichkeiten

292.471.208 Jahre

Türme von Hanoi Automatisch?



Türme von Hanoi

Zielfunktion

```
1. def min_zielfunktion(tuerme): return (21 - sum(tuerme.C))
```

Türme von Hanoi

Stochastischer Bergsteigeralgorithmus

```
1. def algorithmus():
2.     aktueller_zustand = []
3.     aktueller_zielwert = max_wert
4.     count = 0
5.     while count < versuche and aktueller_zielwert > 0:
6.         neuer_zustand = kopiere(aktueller_zustand)
7.         neuer_zustand.append(random.choice(optionen))
8.         neuer_zielwert = berechne_zielwert(neuer_zustand)
9.         if neuer_zielwert < aktueller_zielwert:
10.            aktueller_zustand = neuer_zustand
11.            aktueller_zielwert = neuer_zielwert
12.        count += 1
13.    return aktueller_zustand
14.
15.
16.
17.
18.
```

Türme von Hanoi

Stochastischer Bergsteigeralgorithmus

DEMO

Türme von Hanoi

Stochastischer Bergsteigeralgorithmus

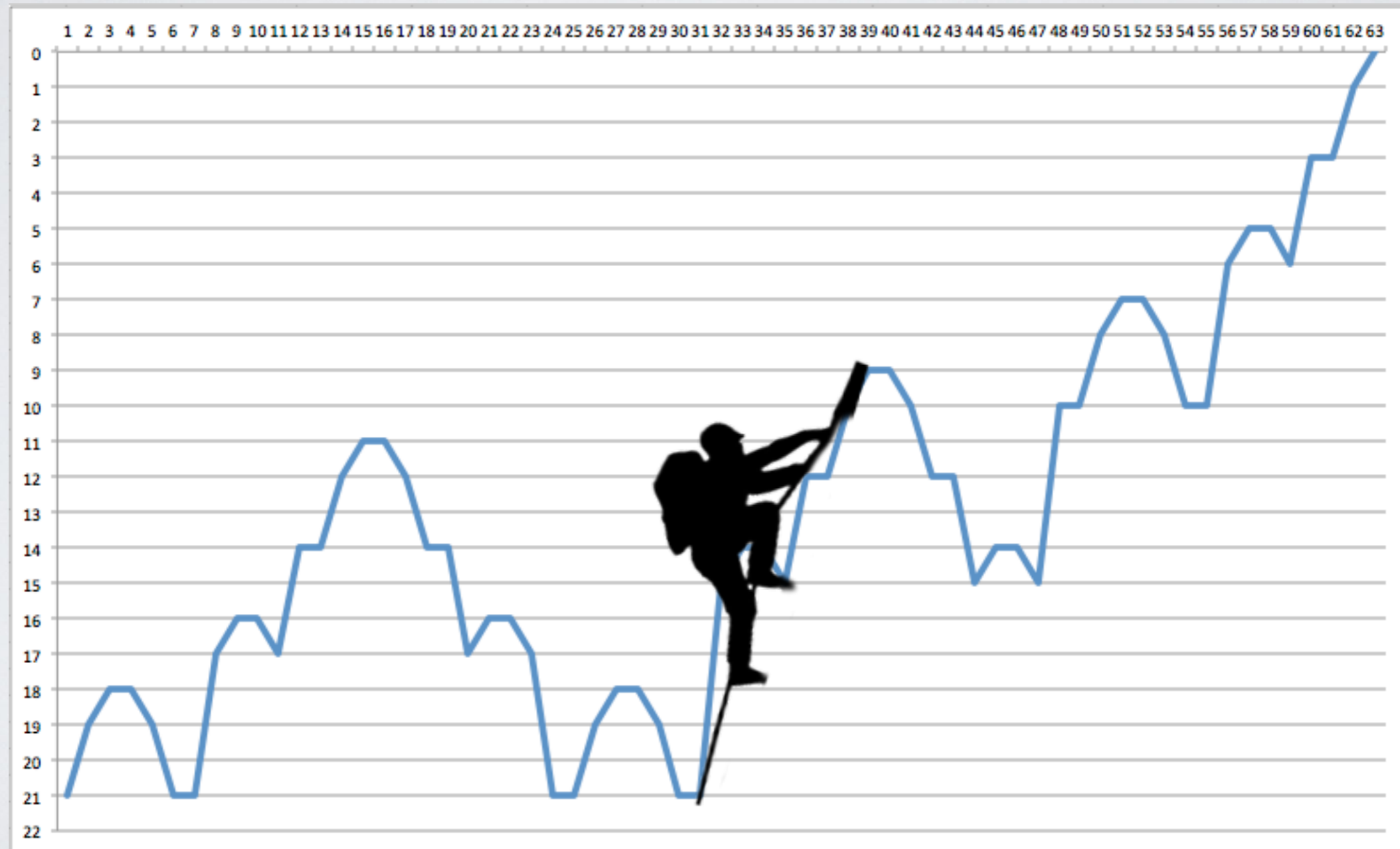
`['tuerme.AtoC()'] -> [65432][][1], Anzahl Schritte: 1, Zielwert: 20`

Türme von Hanoi

Zielfunktion

```
1. def min_zielfunktion(tuerme): return (21 - sum(tuerme.C))
```


Türme von Hanoi Suchraumlanschaft



Türme von Hanoi

Zufallsbewegung

```
1. def algorithmus():
2.     aktueller_zustand = []
3.     aktueller_zielwert = max_wert
4.     count = 0
5.     while count < versuche and aktueller_zielwert > 0:
6.         neuer_zustand = kopiere(aktueller_zustand)
7.         neuer_zustand.append(random.choice(optionen))
8.         neuer_zielwert = berechne_zielwert(neuer_zustand)
9.         if neuer_zielwert < aktueller_zielwert or \
10.            (neuer_zielwert < max_wert and random.choice([True, False])):
11.             aktueller_zustand = neuer_zustand
12.             aktueller_zielwert = neuer_zielwert
13.         count += 1
14.     return aktueller_zustand
15.
16.
17.
18.
```

Türme von Hanoi

Zufallsbewegung

DEMO

Türme von Hanoi Zufallsbewegung

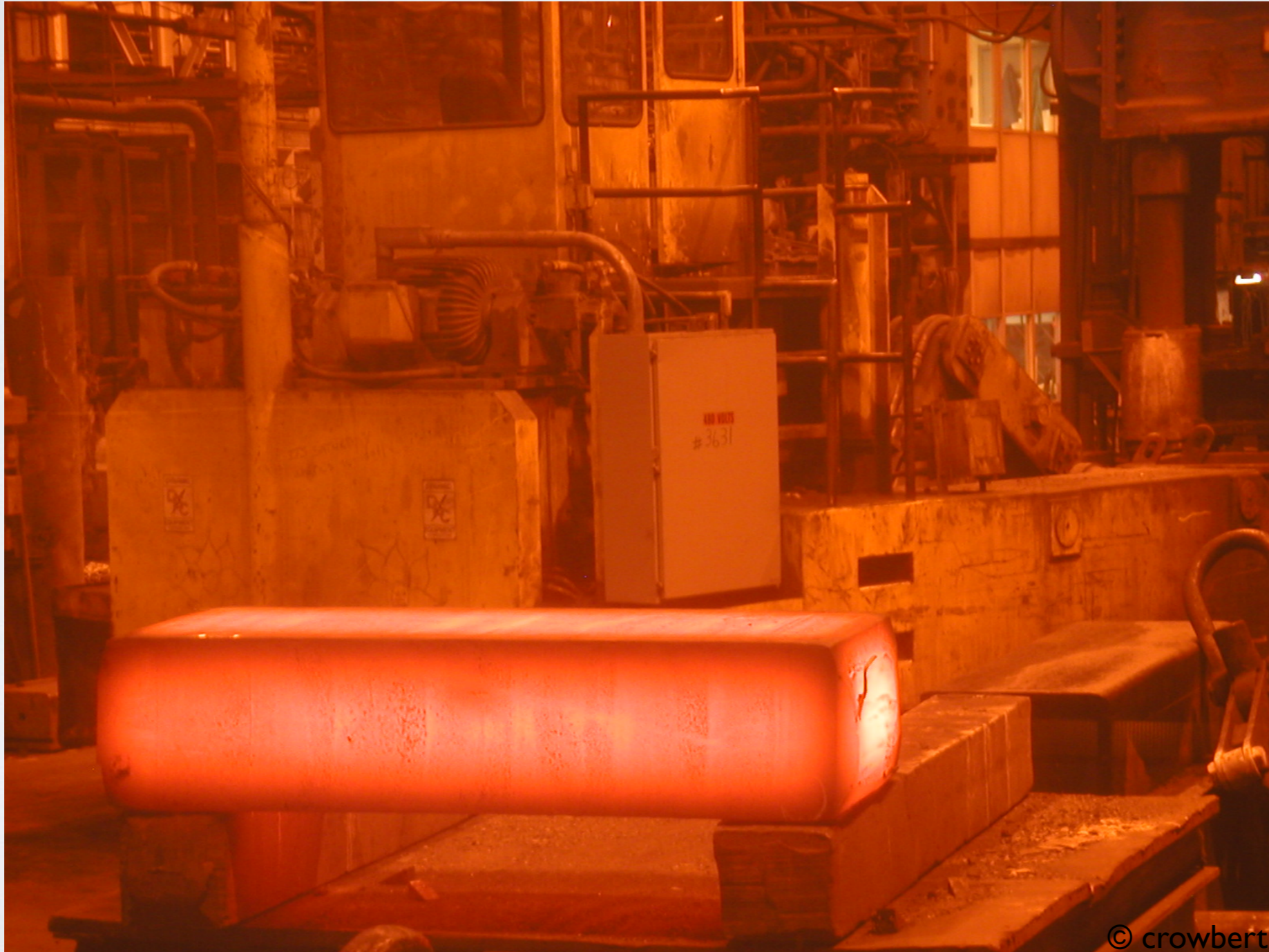
```
[tuerme.AtoB(), tuerme.BtoC(), tuerme.CtoB(), tuerme.BtoA(),  
tuerme.AtoB(), ...] -> [51][][6432], Anzahl Schritte: 2018, Zielwert: 6
```

Türme von Hanoi

Zufallsbewegung

```
1. def algorithmus():
2.     aktueller_zustand = []
3.     aktueller_zielwert = max_wert
4.     count = 0
5.     while count < versuche and aktueller_zielwert > 0:
6.         neuer_zustand = list(aktueller_zustand)
7.         neuer_zustand.append(random.choice(optionen))
8.         neuer_zielwert = berechne_zielwert(neuer_zustand)
9.         if neuer_zielwert < aktueller_zielwert or \
10.            (neuer_zielwert < max_wert and randint(0, versuche) > count):
11.             aktueller_zustand = neuer_zustand
12.             aktueller_zielwert = neuer_zielwert
13.         count += 1
14.     return aktueller_zustand
15.
16.
17.
18.
```

Türme von Hanoi Simulierte Abkühlung



© crowbert

Türme von Hanoi

Simulierte Abkühlung

DEMO

Türme von Hanoi

Simulierte Abkühlung

```
[tuerme.AtoB(), tuerme.AtoC(), tuerme.BtoA(), tuerme.AtoB(), tuerme.BtoA(), ...] ->  
[6][][54321], Anzahl Schritte: 1869, Zielwert: 6
```


Türme von Hanoi

Simulierte Abkühlung

```
1. def algorithmus():
2.     aktueller_zustand = []
3.     aktueller_zielwert = max_wert
4.     count = 0
5.     while count < versuche and aktueller_zielwert > 0:
6.         neuer_zustand = list(aktueller_zustand)
7.         neuer_zustand.append(random.choice(optionen))
8.         neuer_zielwert = berechne_zielwert(neuer_zustand)
9.         if neuer_zielwert < aktueller_zielwert or \
10.            (neuer_zielwert < max_wert and randint(0, versuche) > count):
11.             aktueller_zustand = neuer_zustand
12.             aktueller_zielwert = neuer_zielwert
13.         count += 1
14.     return aktueller_zustand
15.
16.
17.
18.
```

Türme von Hanoi

Balkensuche

```
1. def algorithmus():
2.     aktuelle_zustaende = [(max_wert, []) for i in range(0, k)]
3.     count = 0
4.     while count < versuche and aktuelle_zustaende[0][0] > 0:
5.         neue_zustaende = []
6.         for idx in xrange(len(aktuelle_zustaende)):
7.             (aktueller_zielwert, aktueller_zustand) = ziehe(aktuelle_zustaende)
8.             neuer_zustand = kopiere(aktueller_zustand)
9.             neuer_zustand.append(random.choice(optionen))
10.            neuer_zielwert = berechne_zielwert(neuer_zustand)
11.            if neuer_zielwert < aktueller_zielwert or \
12.                (neuer_zielwert < max_wert and randint(1, versuche) > count):
13.                insert(neue_zustaende, neuer_zielwert, neuer_zustand)
14.            else:
15.                insert(neue_zustaende, aktueller_zielwert, aktueller_zustand)
16.            count += 1
17.        aktuelle_zustaende = neue_zustaende
18.    return aktuelle_zustaende[0][1]
```

Türme von Hanoi

Balkensuche

DEMO

Türme von Hanoi

Balkensuche

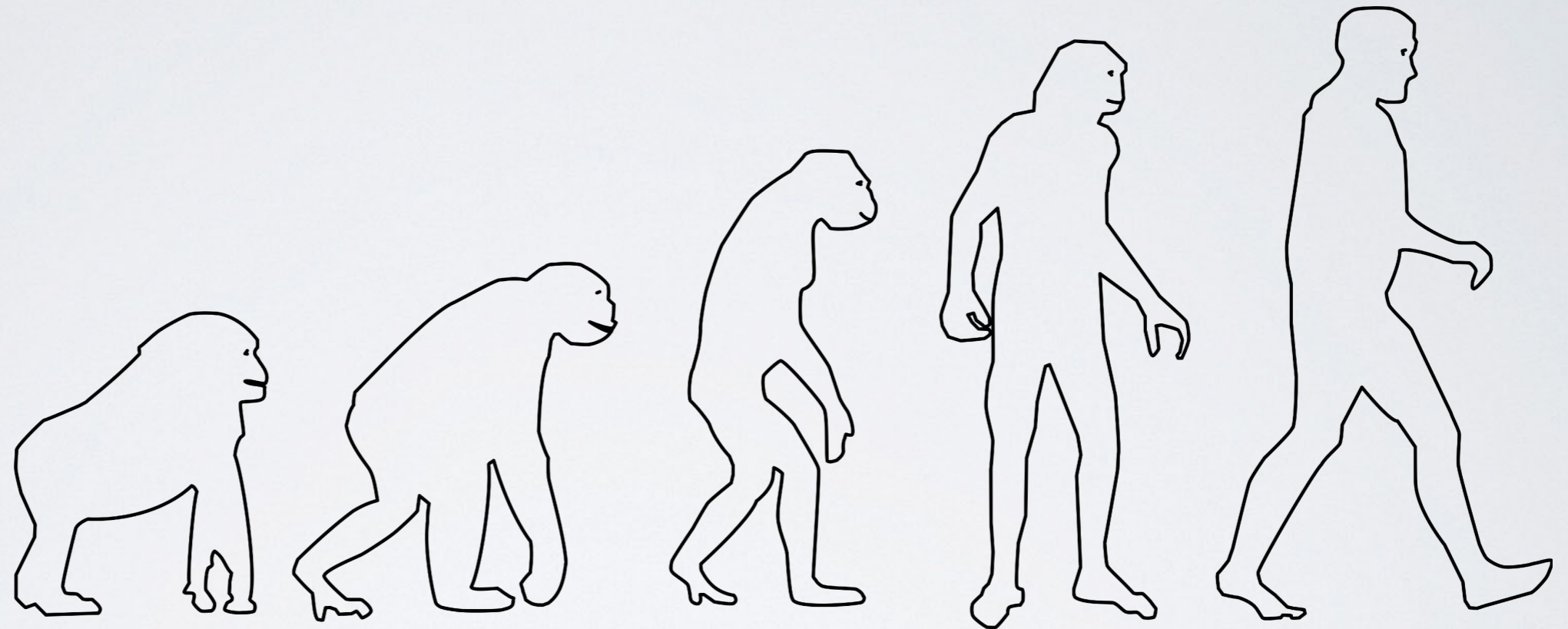
```
[tuerme.AtoC(), tuerme.AtoB(), tuerme.CtoB(), tuerme.BtoC(), tuerme.CtoA(), ...] ->  
[5][][64321], Anzahl Schritte: 1891, Zielwert: 5
```

Türme von Hanoi

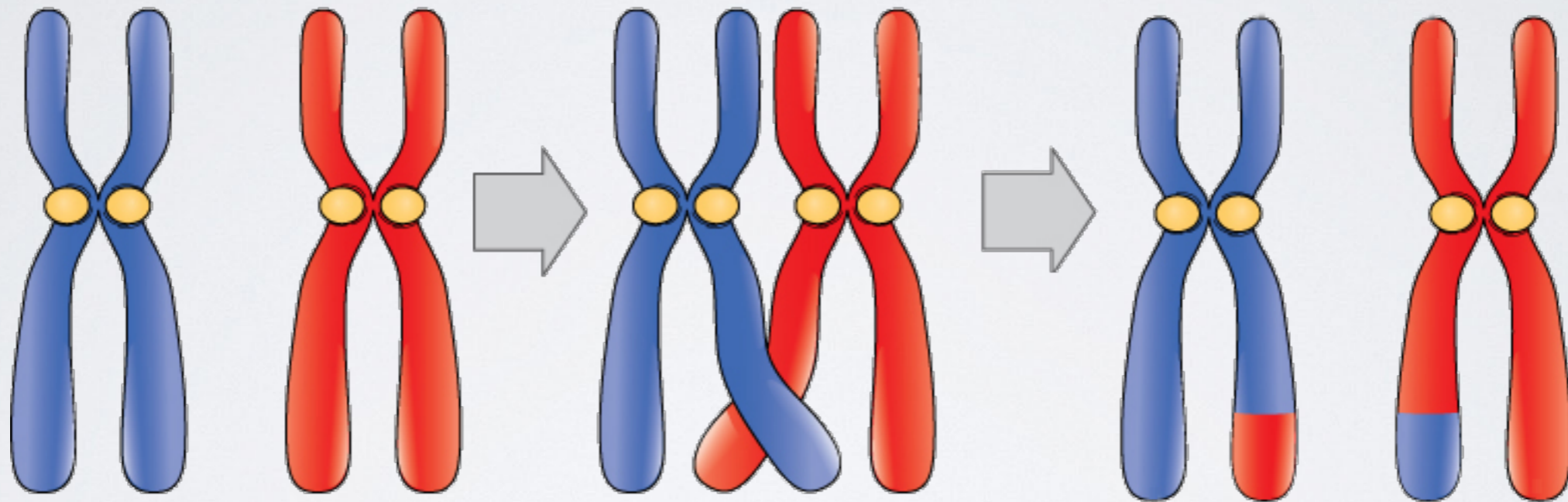
Balkensuche

```
1. def algorithmus():
2.     aktuelle_zustaende = generiere_initiale(k)
3.     count = 0
4.     while count < versuche and aktuelle_zustaende[0][0] > 0:
5.         neue_zustaende = []
6.         for idx in xrange(len(aktuelle_zustaende)):
7.             (aktueller_zielwert, aktueller_zustand) = ziehe(aktuelle_zustaende)
8.             neuer_zustand = kopiere(aktueller_zustand)
9.             neuer_zustand.append(random.choice(optionen))
10.            neuer_zielwert = berechne_zielwert(neuer_zustand)
11.            if neuer_zielwert < aktueller_zielwert or \
12.                (neuer_zielwert < max_wert and randint(1, versuche) > count):
13.                insert(neue_zustaende, neuer_zielwert, neuer_zustand)
14.            else:
15.                insert(neue_zustaende, aktueller_zielwert, aktueller_zustand)
16.            count += 1
17.        aktuelle_zustaende = neue_zustaende
18.    return aktuelle_zustaende[0][1]
```

Türme von Hanoi Genetischer Algorithmus

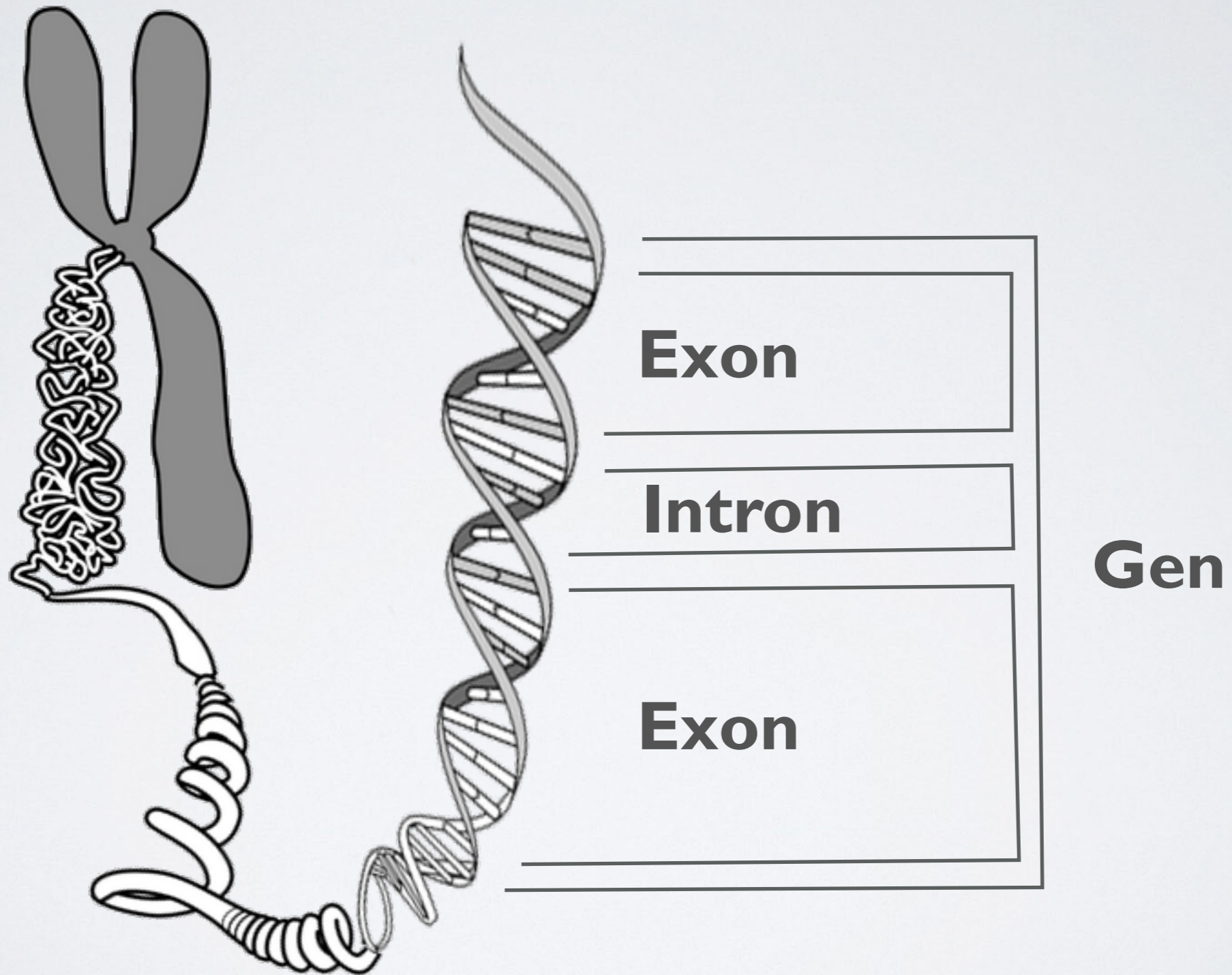


Türme von Hanoi Genetischer Algorithmus



© Robert Bear and David Rintoul

Türme von Hanoi Genetischer Algorithmus



Türme von Hanoi

Genetischer Algorithmus

```
1. def algorithmus():
2.     aktuelle_zustaende = generiere_initiale(k)
3.     count = 0
4.     while count < versuche and aktuelle_zustaende[0][0] > 0:
5.         neue_zustaende = []
6.         for idx in xrange(len(aktuelle_zustaende)):
7.             neuer_zustand = ziehe(aktuelle_zustaende)
8.             neuer_zustand = rekombiniere(neuer_zustand, ziehe(aktuelle_zustaende))
9.             neuer_zustand = mutiere(neuer_zustand)
10.            if neuer_zielwert < aktueller_zielwert or \
11.                (neuer_zielwert < max_wert and randint(1, versuche) > count):
12.                insert(neue_zustaende, neuer_zustand)
13.            else:
14.                insert(neue_zustaende, aktueller_zustand)
15.            count += 1
16.        aktuelle_zustaende = neue_zustaende
17.    return aktuelle_zustaende[0][1]
18.
```

Türme von Hanoi

Genetischer Algorithmus

DEMO

Türme von Hanoi

Genetischer Algorithmus

```
[tuerme.AtoB(), tuerme.AtoC(), tuerme.BtoA(), tuerme.AtoB(), tuerme.BtoA(), ...] ->  
[][][654321], Anzahl Schritte: 1727, Zielwert: 0
```

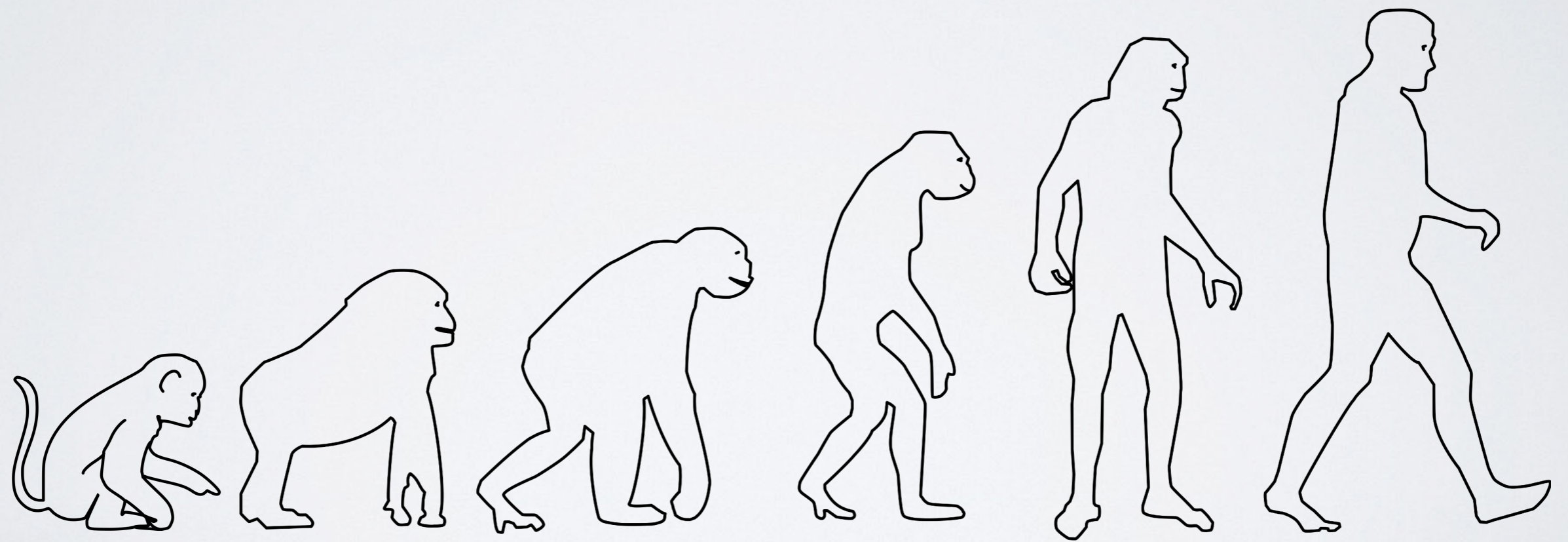
Genetischer Algorithmus

Effizienz

66 Sekunden

statt

150 Mio. Jahre



Welche Fehler kann der Affe finden?



When is a bug not a bug?



When it's a feature!

Is it a bug?

```
1. def auth(username, password):  
2.     if username == 'admin' and password == 'geheim':  
3.         return True  
4.     if hash(password + get_salt(username)) == get_pwd_hash(username):  
5.         return True  
6.     return False
```


Is it a bug?

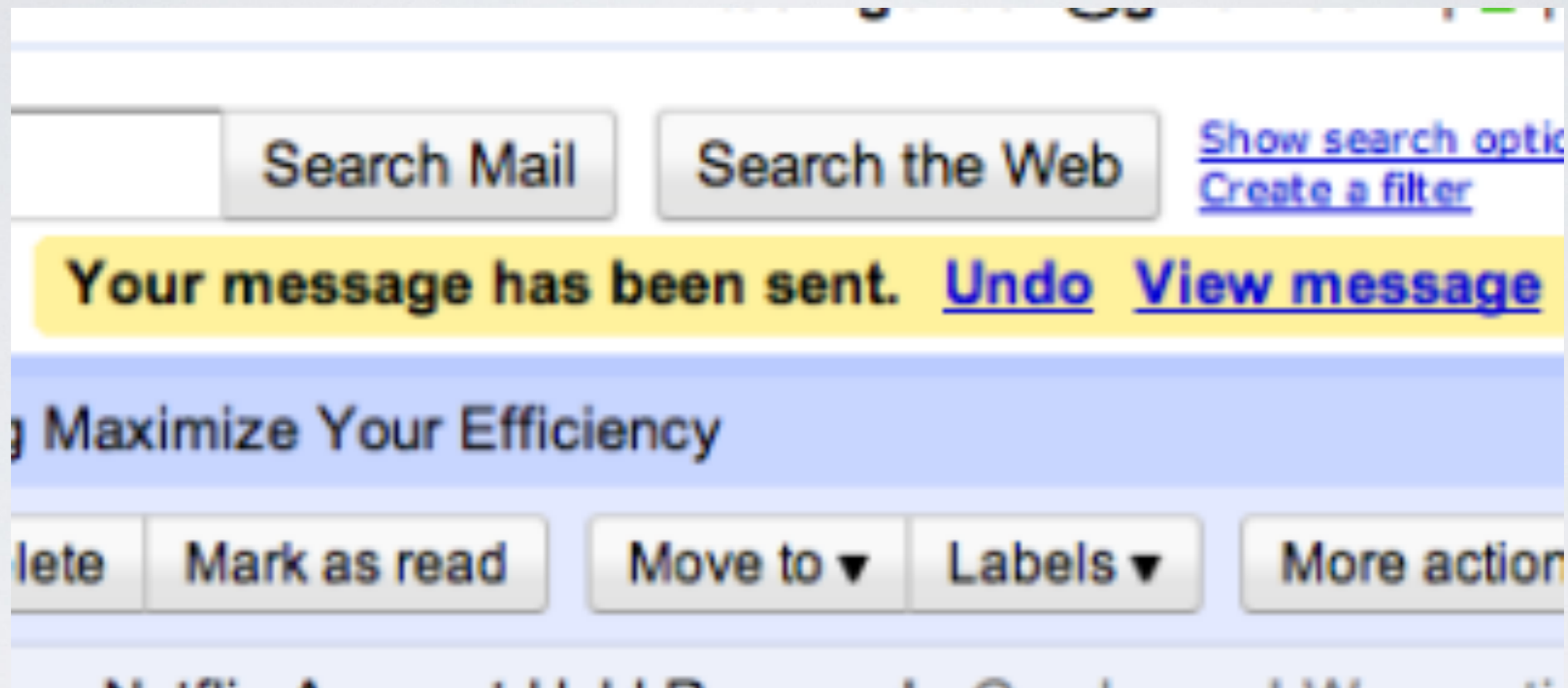
```
.  
..  
text.c  
other.files
```

```
1. if (name[0] == '.') continue;
```

Is it a bug?



Is it a bug?



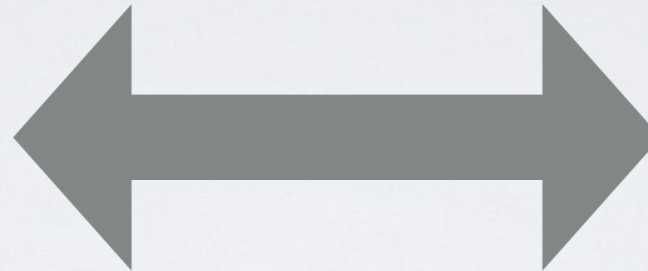
What is a bug?

“ Without specification, there are no bugs
— only surprises.

Brian Kernighan

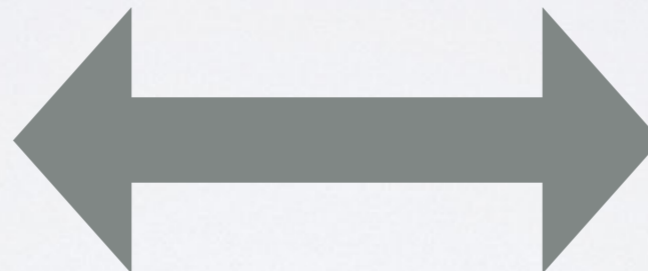
What is a bug?

Spezifikation



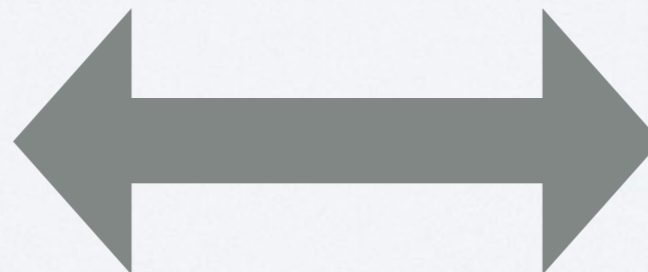
Code

Modell



Code

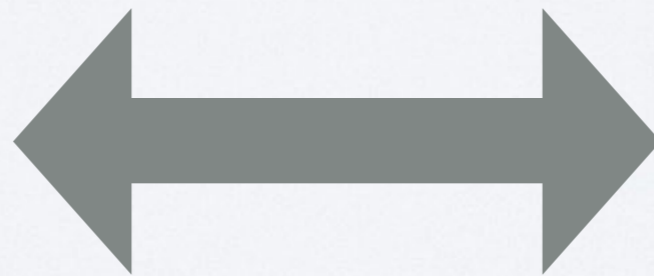
Nutzererwartung



Code

Monkey Testing

Nutzererwartung



Code

Wozu testen wir?



**Nach
Implementierung?**

Auftragsverwaltung

Detail: Kunden Lager Glas Kontaktadressen AluAkti Optionen Waerag | Drucken Auftrag Extras | Fenster

Kundenname: Herr [Name] | Datum: 21.10.2004

Bestand: 100,00 EUR

Bestand: 11,00 EUR

Bestand: 24,20 EUR

Bestand: 5,00 EUR

Bestand: 7,00 EUR

Bestand: 60,30 EUR

Bestand: 96,7% EUR

Bestand: 980,00 EUR

Buchen: Sachkonten Allgemeinbeleg

Datum: 21.10.2004

Umsatz: 100,00 EUR

Umsatz: 11,00 EUR

Umsatz: 24,20 EUR

Umsatz: 5,00 EUR

Umsatz: 7,00 EUR

Umsatz: 60,30 EUR

Umsatz: 96,7% EUR

Umsatz: 980,00 EUR

Rechnungen eingeleitet

Region: [Auswahl]

Anrede: Herr

Vorname: [Name]

Nachname: [Name]

Strasse: [Adresse]

PLZ: [PLZ]

Ort: [Ort]

Telefon: [Telefon]

E-Mail: [E-Mail]

Homepage: [Homepage]

Personendaten

Pers. ID Nr.: 9999

Vorname: Jürgen

Nachname: Cimbölek

Strasse: [Adresse]

PLZ: [PLZ]

Ort: [Ort]

Telefon: [Telefon]

E-Mail: [E-Mail]

Produkt

Produktname: [Name]

Preis: 100,00 EUR

Produktbild:

Datum: 21.10.2004

Wohnort

Land: [Auswahl]

Stadt: [Auswahl]

Strasse: [Adresse]

PLZ: [PLZ]

Ort: [Ort]

Telefon: [Telefon]

E-Mail: [E-Mail]

Teilnahme: 201006 Mustermeister Barset; 3: Definitiv

Zuweisung	Rechnung	Budget	Kosten	Korrespondenz
Name: Mustermeister	Kursnummer: 201006			
Vorname: Daniel	Kursjahr: 2000/01			
Nachname: Harty	Kurs / Fach: eB-1C			
Strasse: Bahnhofstrasse	Stufe: Kursnummer 201006			
PLZ: 00 00 00				
Ort: [Ort]				

TableSelectionDemo

First Name	Last Name	Sport	# of Years	Vegetarian
Kathy	Smith	Snowboarding	5	<input type="checkbox"/>
John	Doe	Rowing	3	<input checked="" type="checkbox"/>
Sue	Black	Knitting	2	<input type="checkbox"/>
Jane	White	Speed reading	20	<input checked="" type="checkbox"/>
...

- Selection Mode**
- Multiple Interval Selection
 - Single Selection
 - Single Interval Selection
- Selection Options**
- Row Selection
 - Column Selection
 - Cell Selection

Jahreszeugnisse

Name, Vorname: [Name]

Nachname: [Name]

Strasse: [Adresse]

PLZ: [PLZ]

Ort: [Ort]

Telefon: [Telefon]

E-Mail: [E-Mail]

Homepage: [Homepage]

Arbeitszeugnis

Name: [Name]

Strasse: [Adresse]

PLZ: [PLZ]

Ort: [Ort]

Telefon: [Telefon]

E-Mail: [E-Mail]

Arbeitszeugnis

Name: [Name]

Strasse: [Adresse]

PLZ: [PLZ]

Ort: [Ort]

Telefon: [Telefon]

E-Mail: [E-Mail]

Arbeitszeugnis

Name: [Name]

Strasse: [Adresse]

PLZ: [PLZ]

Ort: [Ort]

Telefon: [Telefon]

E-Mail: [E-Mail]

Arbeitszeugnis

Name: [Name]

Strasse: [Adresse]

PLZ: [PLZ]

Ort: [Ort]

Telefon: [Telefon]

E-Mail: [E-Mail]

TableSelectionDemo

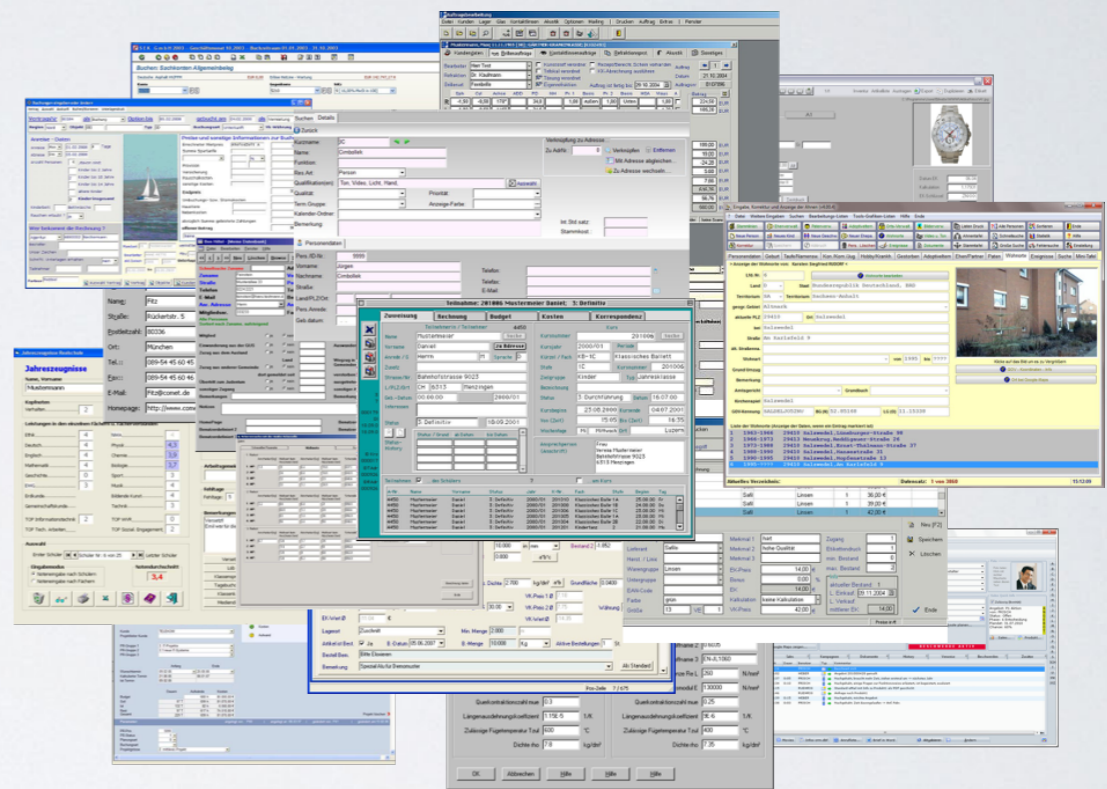
First Name	Last Name	Sport	# of Years	Vegetarian
Kathy	Smith	Snowboarding	5	<input type="checkbox"/>
John	Doe	Rowing	3	<input checked="" type="checkbox"/>
Sue	Black	Knitting	2	<input type="checkbox"/>
Jane	White	Speed reading	20	<input checked="" type="checkbox"/>

Selection Mode

- Multiple Interval Selection
- Single Selection
- Single Interval Selection

Selection Options

- Row Selection
- Column Selection
- Cell Selection



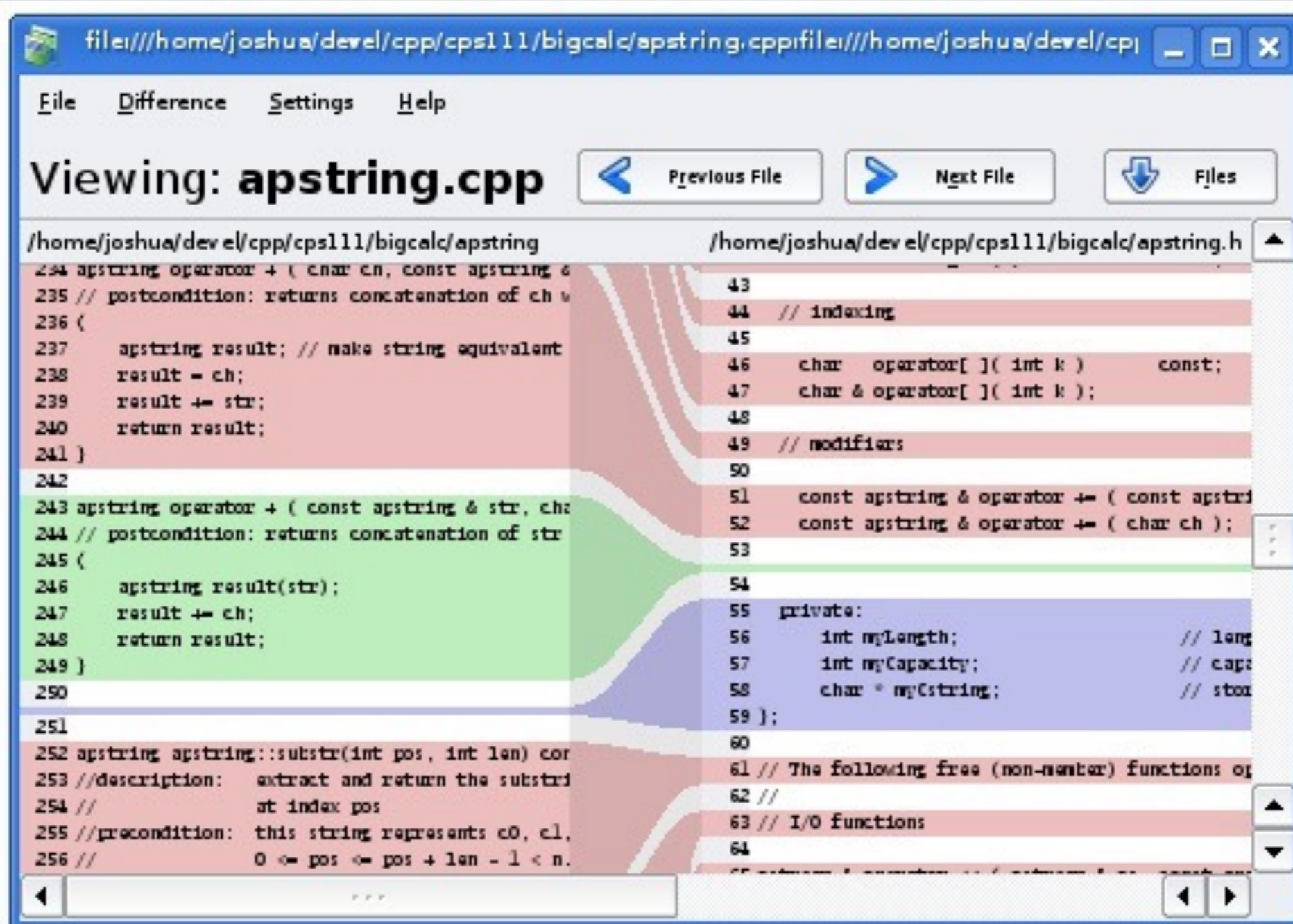
Wozu testen wir?



**Nach
Implementierung?**



**Nach
Änderung?**





```
public void testSomething() throws Exception {  
    // here be some test code ...  
    UIElement element = driver.findElement("path/to/element");  
    assertEquals(null, element.getAction());  
    assertEquals("action", element.getActionCommand());  
    assertEquals(true, element.isEnabled());  
    assertEquals(true, element.isFocusable());  
    assertEquals("Lucida Grande", element.getFont().getName());  
    assertEquals(13, element.getFont().getSize());  
    assertEquals(0, element.getFont().getStyle());  
    assertEquals(23, element.getHeight());  
    assertEquals(null, element.getIcon());  
    assertEquals(0, element.getMnemonic());  
    assertEquals(null, element.getPressedIcon());  
    assertEquals(true, element.isSelected());  
    assertEquals("label", element.getText());  
    assertEquals(119, element.getWidth());  
    assertEquals(27, element.getX());  
    assertEquals(191, element.getY());  
}
```



Ihr Vollautomatischer Regressionstester.

Weniger Risiko, weniger Stress, weniger Kosten!

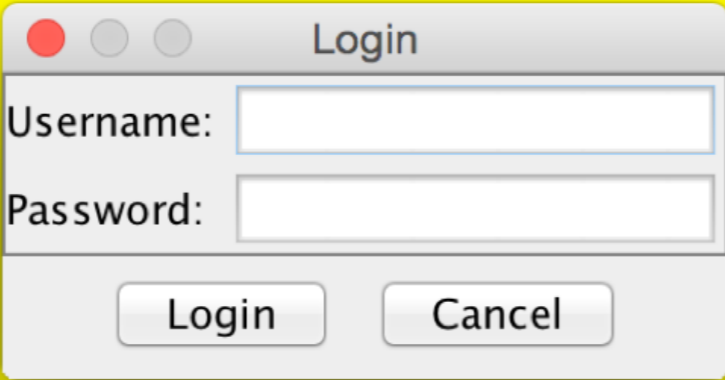
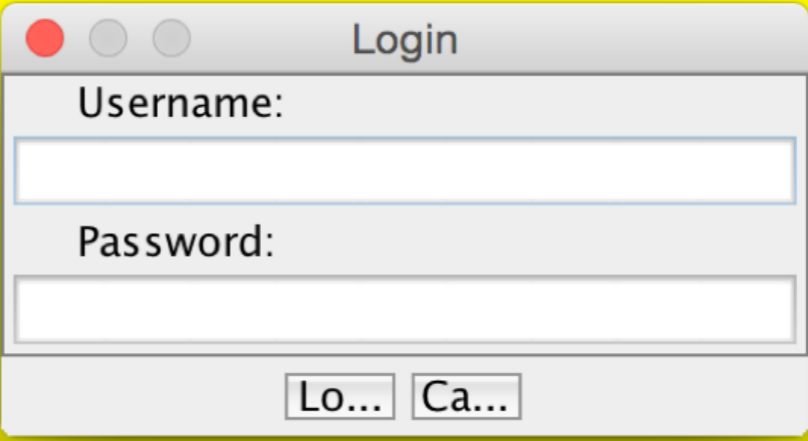
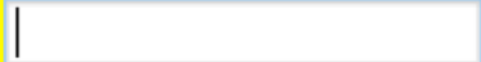
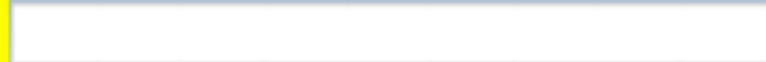
Haben Sie genug vom Testen?

Sowohl manuelles Testen, als auch das manuelle Erstellen und Pflegen von automatischen Tests sind unbeliebte Zeitfresser. Direkt aus der Forschung ist ReTest als bisher einziges Produkt seiner Art Ihr



Haben Sie genug vom

Testen?

<input type="checkbox"/> LoginDialog		
<input type="checkbox"/> Label Username	<p style="text-align: center;">Username:</p> <p>x-Coordinate: 0 y-Coordinate: 5</p>	<p style="text-align: center;">Username:</p> <p>x-Coordinate: 24 y-Coordinate: 0</p>
<input type="checkbox"/> Field Username	 <p>x-Coordinate: 71 y-Coordinate: 1 Width: 254px</p>	 <p>x-Coordinate: 1 y-Coordinate: 23 Width: 543px</p>
<input type="checkbox"/> Label Password	<p style="text-align: center;">Password:</p> <p>x-Coordinate: 0 y-Coordinate: 44</p>	<p style="text-align: center;">Password:</p> <p>x-Coordinate: 24 y-Coordinate: 48</p>



Monkey Testing

Coverage Maximization

Saving GUI Results

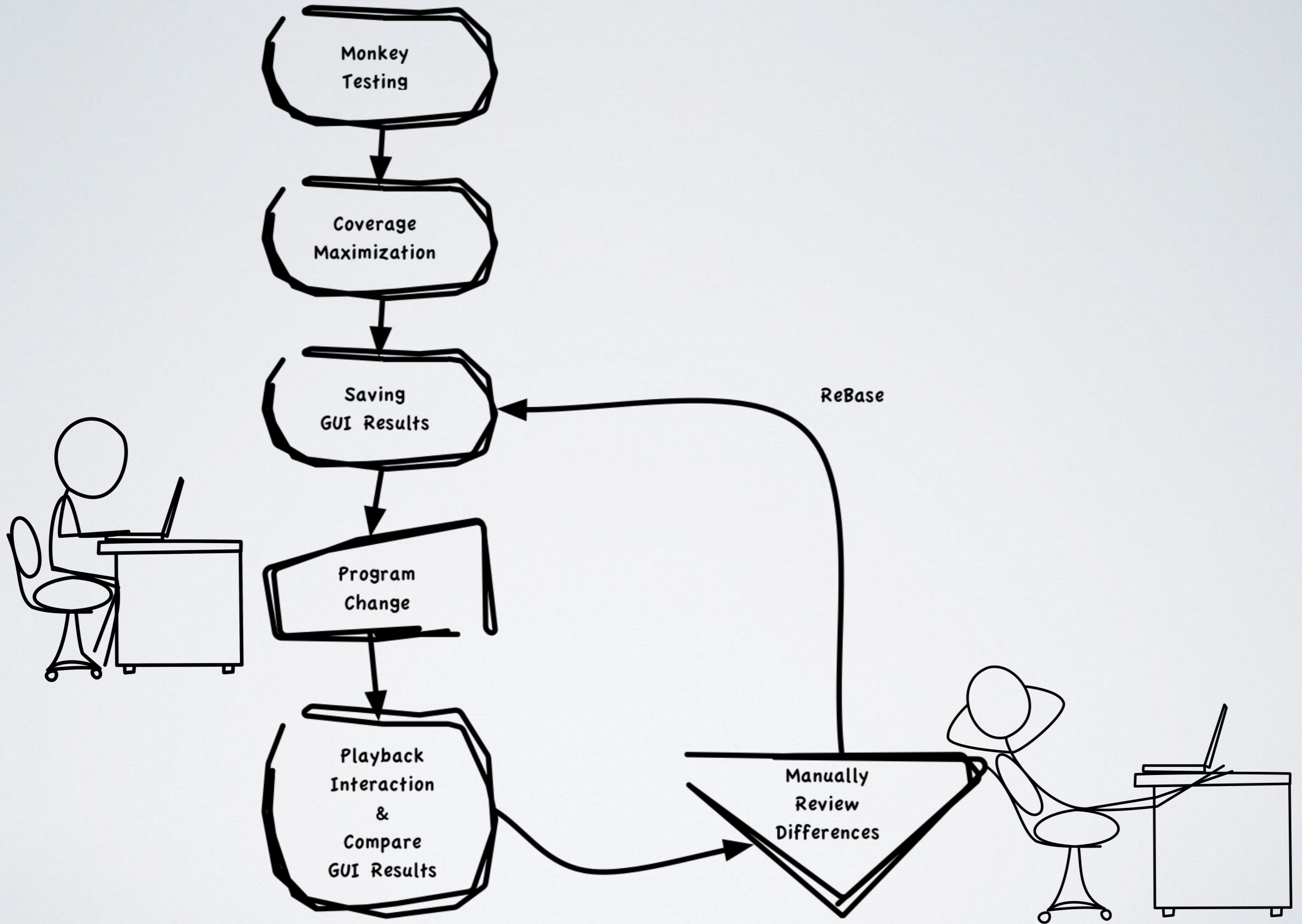
Program Change

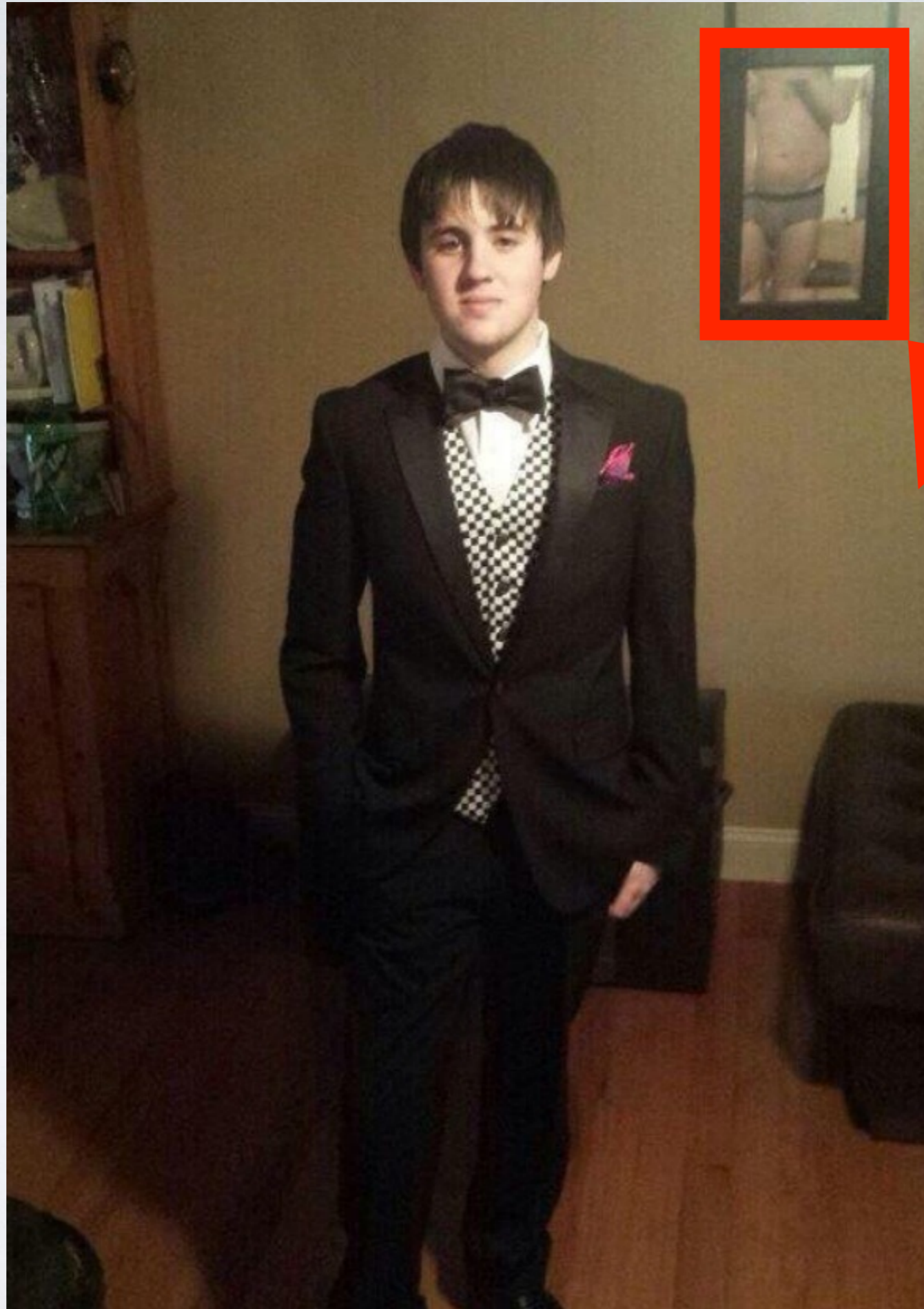
Playback Interaction & Compare GUI Results

Manually Review Differences

Repeat







Ausführung Wiederholbar?

Object.hashCode()

TableSelection

First Name	Last Name	Sport
Kathy	Smith	Snowboarding
John	Doe	Rowing
Sue	Black	Knitting
Jane	White	Speed reading
Lee	Brown	Pool

Selection Mode

- Multiple Interval Selection
- Single Selection
- Single Interval Selection

Selection Options

- Row Selection
- Column Selection
- Cell Selection

TableSelection

First Name	Last Name	Sport
Kathy	Smith	Snowboarding
Sue	Black	Knitting
John	Doe	Rowing
Jane	White	Speed reading
Lee	Brown	Pool

Selection Mode

- Multiple Interval Selection
- Single Selection
- Single Interval Selection

Selection Options

- Row Selection
- Column Selection
- Cell Selection

System.currentTimeMillis()

Datum und Zeit

Datum:

30.05.2005



Zeit:

10:24:39



Datum und Zeit:

30.05.2005 10:24:40



Thread-Schedules

Synchronization

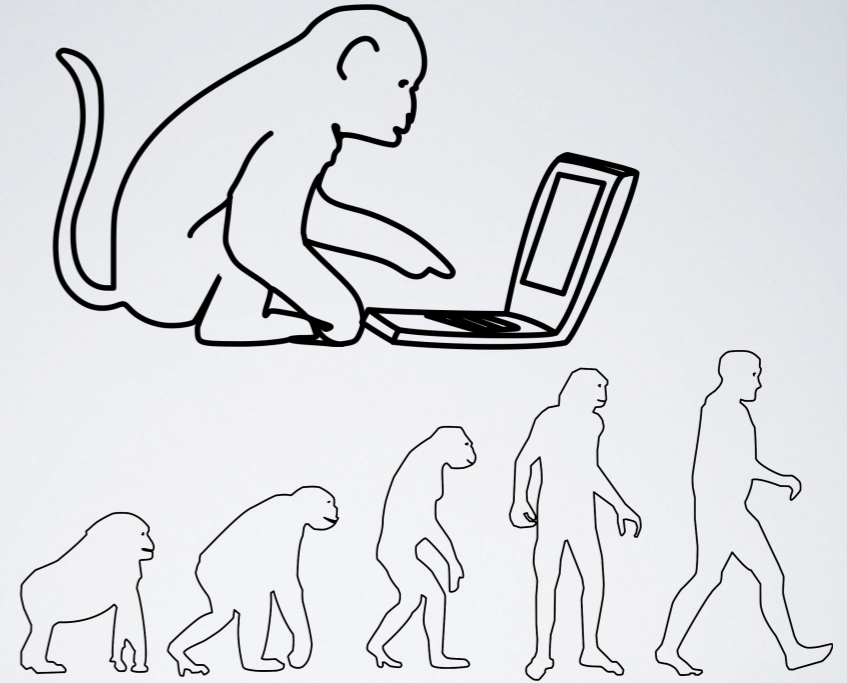
1. `WebDriverWait.until(condition-that-finds-the-element)`

System.exit()

Ihre Daten

Testbed

Intelligenter Affe

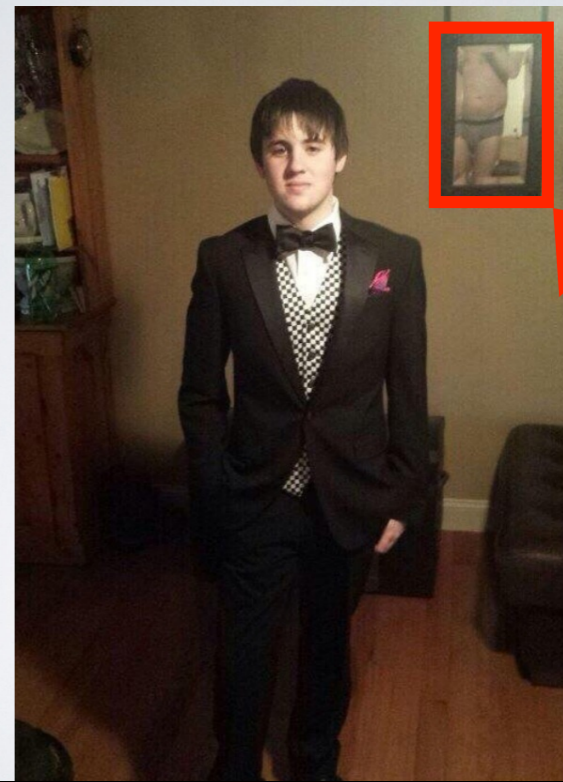


Material und Infos auf: www.retest.de!

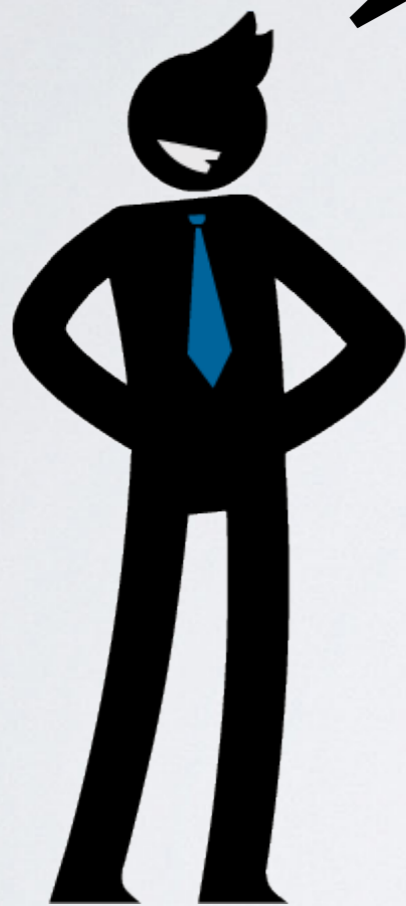
When is a bug not a bug?



When it's a feature!



Machen Sie den Test!



 ReTest

The logo for ReTest features a green icon on the left consisting of a curved line with several vertical bars of varying heights, resembling a bar chart or a stylized 'R'. To the right of this icon, the word 'ReTest' is written in a sans-serif font. The 'Re' is in green, and 'Test' is in dark blue.

www.retest.de