

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGHANN SYEN

Bereich Berufsnummer IHK-Nummer Prüflingsnummer  
3 0 6 5 1 1 101 20541

Sp. 1-2

Sp. 3-6

Sp. 7-9

Sp. 10-14



Bearbeitungsbogen-Nr. 1

Korrekturrand

## Aufgabenanalyse:

Eine Firma möchte einen optischen Autokorrelator bauen, welcher mit einem Laser, halbdurchlässigen und ~~te~~ vollreflektierenden Spiegeln, einem Kristall, und einem Filter und einem Detektor punktiert. Der Detektor muss hierbei die Autokorrelationsfunktion des Eingangssignals, die Aufschluss über die Breite des Laserpulses oder die Ursprungsfrequenz gibt. Die Messdaten werden allerdings ~~mit~~ vom Detektor permanent und nebenläufig geliefert, was mit einem Thread modelliert werden soll, welcher mit 20 Hz, also alle 0,05 Sekunden eine neue Messdatei zur Verfügung stellt und die alte überschreibt. Der Lese-Thread ist hierbei Thread A und läuft ständig weiter, indem die Daten "0.txt" bis "9.txt" immer wieder eingelesen werden und den aktuellen Datensatz überschreiben. Der Thread B soll dann diese Messdaten weiterverarbeiten und mit den später erklärten mathematischen Methoden die Funktion modellieren, bzw. die Messwerte umrechnen und glätten. Schließlich wird die fertige Berechnung dann an Thread C weitergegeben, um die Daten in einem auch später erklärten Format in eine Datei auszugeben, welche als Präfix "out" besitzt und ansonsten gleich heißt. Das Programm wird dann beendet, wenn alle Messdaten verarbeitet wurden. Es kann durchaus vorkommen, dass eine Messdatei häufiger

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMANN SVEIN			
Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer
3 0	6 5 1 1	11 0 11	2005411
Sp. 1-2	Sp. 3-6	Sp. 7-9	Sp. 10-14



Bearbeitungsbogen-Nr.

21

Korrekturrand

eingelesen wird. Diese muss dann allerdings nicht erneut verarbeitet werden, es wird dann einfach auf eine noch nicht verarbeitete Datei gewartet.

## Beschreibung der mathematischen Methoden

Als erstes werden alle x-Werte in Phasenwellen umgerechnet, was mit der Formel  $\hat{x}_k = \frac{\tilde{x}_k}{2^{18}-1} \cdot 266,3 - 132,3$ ,  $\forall k \in [0, N-1]$  passieren soll. Zur ~~zeit~~ Zeitsparnis und Verbesserung der Genauigkeit würde hierbei eine Umstellung sinnvoll sein, sodass der Bruch  $\frac{266,3}{2^{18}-1}$  nicht immer wieder berechnet werden muss. Also wird es eine Variable tmp geben, welche diesen Bruch beschreibt und die Formel wird zu  $\hat{x}_k = \text{tmp} \cdot \tilde{x}_k - 132,3$ . Des Weiteren werden noch alle y-Werte normiert, sprich jeder Wert wird durch das Maximum ~~der~~ aller y-Werte geteilt.

Danach sollen die Daten geglättet werden, indem der sog. gleitende Mittelwert berechnet werden soll. Die Formel hierfür ist  $x_k = \frac{1}{n} \sum_{i=0}^{n-1} \hat{x}_{k+i} \text{ mit } z = \frac{n-1}{2}, \forall k \in [0, N-1]$  und  $n = \lfloor 0,002 \cdot N \rfloor - 1$ , für  $\lfloor 0,002 \cdot N \rfloor$  gerade, wobei n  $\lceil 0,002 \cdot N \rceil$ , für  $\lceil 0,002 \cdot N \rceil$  ungerade die Größe des Mittelungsfensters beschreibt, ungerade ist und 0,2% von N entspricht. Es sollen zudem noch geeignete Werte für  $k < z$  und  $k > N - 1 - z$  bestimmt werden, sprich den linken und rechten Rand der Messungen. Das Symbol  $\lfloor \rfloor$  heißt unter Gaußklammern und beschreibt, wie auf ganze Zahlen gerundet werden soll.

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMANN SWEN

Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer
3 0	6 5 1 1	1 0 1	2 0 5 4 1

Sp. 1-2

Sp. 3-6

Sp. 7-9

Sp. 10-14



Bearbeitungsbogen-Nr.

3

Korrekturrand

Als nächstes wird die obere Einkrüllende bestimmt, welche die Auto Korrelationsfunktion oben komplett einschließt. Da dies über numerisch sehr aufwändig ist, reicht es aus, diese Einkrüllende einfacher zu approximieren und zwar indem zuerst von links beginnend jedem Positions-  
wert der zuletzt höchste Intensitätswert zugewendet wird,  
solange bis man am Maximum ist, dann wird das  
gleiche Verfahren von rechts wiederholt.

Zuletzt wird noch die Pulsbreite  $b$  berechnet, welche sich über den Abstand der beiden Punkte  $L$  und  $R$  berechnet. Diese sind Punkte auf der oberen Einkrüllenden, wobei die Gerade die durch  $L$  und  $R$  geht, genau auf der Hälfte der Grundlinie und des Maximums liegt. Die Grundlinie stellt dabei die niedrigere Höhe des äußersten linken Prozents der Intensitätswerte dar.  
~~das heißt die drei gemittelten y-Werte der ersten Berechnung gewinnen~~  
~~finden hier nochmal an Bedeutung. Es müssen also zuerst~~  
die gemittelten Werte der ersten Berechnung aufsteigend sortiert werden, wobei dann geschaut wird, welche Werte alle ~~wiederholen~~ im Intervall  $[0,010; 0,01]$  liegen. ~~die~~  
Diese werden dann ebenfalls gemittelt und der daraus resultierende Wert beschreibt die Grundlinie.

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMANN SVEIN

Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer
3 0	6 5 1 1	1 0 1	2 0 5 4 1

Sp. 1-2

Sp. 3-6

Sp. 7-9

Sp. 10-14



Bearbeitungsbogen-Nr.

4

Konzepterstellung zur Notwendigkeit von Einlesen, Verarbeiten, und Ausgeben

Korrekturrand

Bei dem gegebenen Problem könnte man ganz ähnlich nach dem IPO (Input - Process - Output) oder EVA (Eingabe - Verarbeitung - Ausgabe) Prinzip gearbeitet werden, wobei jedoch nur genau ein Thread für das Liefern der Daten, also den Input zuständig ist. Hierfür könnte ein Interface bereitgestellt werden, welches einen ~~fp~~ Thread oder Prozess implementiert. Dieses Interface stellt dann eine default ~~to~~ Methode bereit, die in einem Intervall von 20 Hz immer wieder aus der zu implementierenden read Methode die Daten holt und weitergibt. Der Thread B wird dann z.B., wie auch der Thread C, so lange laufen, wie noch Daten nicht verarbeitet wurden. Wenn Thread A also eine Datei bereitstellt, startet Thread B und berechnet das Ergebnis, welches dann vom Thread C geschrieben werden kann. Thread A, B und C können somit gleichzeitig ausgeführt werden, da die Verarbeitung unabhängig voneinander passiert. Für Thread B würde also ebenfalls ein Interface bereitgestellt werden, welches einen Thread implementiert und die Methode ~~berechne~~<sup>process</sup> mit den Parametern aus Thread A ~~A~~ implementieren bereitstellen muss. Der Thread C sieht dann ebenfalls ein Interface vor, welches einen Thread implementiert und die Methode ~~ausgabe~~<sup>write</sup> bereitstellt. Falls Thread C alle Messdaten geschrieben hat, wird Thread A gestoppt und

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

B E R G M A N N S R E N

Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer
3   0	6   5   1   1	1   0   1	2   0   5   4   1
Sp. 1-2	Sp. 3-6	Sp. 7-9	Sp. 10-14



Bearbeitungsbogen-Nr. 5

Korrekturrand

danach auch Thread B. Nachdem alle Threads gestoppt wurden, wird auch das Programm beendet. (Weiter auf S.12)

Entwurf der vier Algorithmen zur Berechnung der Ergebnisse

Nassi - Schneidermann: Umrechnung und Normierung der Daten:

$$\text{Berechne } \text{tmp} = \frac{266 \cdot 3}{2^{18}} = ,$$

Einstelle Lösungsvektor für  $\hat{x}$  mit Länge  $N-1$

Solange  $k < N$ ,  $u = 0$

$$\text{Berechne } \hat{x}_u = \text{tmp} \cdot \hat{x}_u + 132 \cdot 3$$

Speichere  $\hat{x}_u$  in Lösungsvektor für  $\hat{x}$

Incrementiere  $u$  um 1

Einstelle Lösungsvektor für  $\hat{y}$  mit Länge  $N-1$

Berechne Maximum von  $y$  und speichere das in  $yMax$

Solange  $u < N$ ,  $u = 0$

$$\text{Berechne } \hat{y}_u / yMax \text{ und } = \hat{y}_u$$

Speichere  $\hat{y}_u$  in Lösungsvektor für  $\hat{y}$

Incrementiere  $u$  um 1

< Returne Lösungsvektor für  $\hat{y}$  und ~~Lösungsvektor~~  $\hat{x}$  >

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMANN	SVEN					
Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer			
3   0   6   5   1   1   101   20541	Sp. 1-2	Sp. 3-6	Sp. 7-9	Sp. 10-14		



Bearbeitungsbogen-Nr.

6

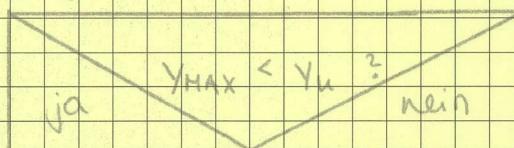
Korrekturrand

Berechne Maximum von  $y$ :

Setze  $y_{\max} = y_0$

(Wieder auf Seite 12)

Solang  $k < N, u = 0$



Incrementiere  $u$

Erstelle Vektor für  $\tilde{x}_k$

Solang  $k < N, u = 0$

Returne  $y_{\max}$

Glättung der Daten:

Beispiel: Sei  $\hat{x}, y \in \{0,1; 0,2; 0,3; 0,4; 0,5; 0,6\}$

$$\text{Also } z = \frac{n-1}{2} = -1$$

Mit  $n = -1$ ,  $k < z$  mit  $h = \{ \}$

$u > 6 - 1 - z \Rightarrow u > 6$  mit  $h = \emptyset$

$\Rightarrow$  wahrscheinlich kein Problem bei ~~10000~~  $N < 50$ .

(Wieder auf Seite 11)

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMANN		SVEN					
Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer				
3 0	6 5 1 1	1 0 1	2 0 5 4 1				

Sp. 1-2

Sp. 3-6

Sp. 7-9

Sp. 10-14



Bearbeitungsbogen-Nr.

7

Korrekturrand

Nassi-Schneidermann:

ja

$10.002 \cdot N \geq 2$  gleich 0 ?

nein

$$n = [0.002 \cdot N] - 1$$

$$n = 10.002 \cdot N$$

Berechne  $\tilde{z} = (n-1)/2$

Berechne  $m = 1/n$ , erstelle Lösungsvektor für  $x_u$

Solange  $u < N$ ,  $u = 0$

Solange  $i < n$ ,  $i = 0$ ,  $x_u = 0$

$$x_u = x_u + \hat{x}_{k-\tilde{z}+i}$$

$$x_u = x_u \cdot m$$

Speichere  $x_u$  in Lösungsvektor

Returne Lösungsvektor für  $x_u$

Obere Einhüllende:

Berechne Maximum von  $y$  und speichere das in  $y_{MAX}$

Erstelle Lösungsvektor für  $y_{OE}$

Erstelle  $y_{TMP} = 0$

Solange  $k < N$  und  $y_{TMP} \neq y_{MAX}$

ja

$y_{TMP} \leq y_k$  ?

nein

$$y_{TMP} = y_k$$

Speichere  $y_{TMP}$  in Lösungsvektor für  $y_{OE}$

Solange  $k > 0$  und  $y_{TMP} \neq y_{MAX}$

ja

$y_{TMP} < y_k$  ?

nein

$$y_{TMP} = y_k$$

Speichere  $y_{TMP}$  in Lösungsvektor für  $y_{OE}$

ja == gleich  $y_{MAX}$  ?

$y_{TMP} == y_{MAX}$  ?

nein

Speichere  $y_{MAX}$  in  $y_{OE}$

Returne Lösungsvektor für  $y_{OE}$

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMANN SVEN

Bereich Berufsnummer IHK-Nummer Prüflingsnummer

3 0 - 6 5 1 1 - 1 0 1 - 2 0 5 4 1

Sp. 1-2

Sp. 3-6

Sp. 7-9

Sp. 10-14



Bearbeitungsbogen-Nr.

8

Korrekturrand

Anmerkungen zu Nassl-Schneidersmann für OE:

- Der Lösungsvektor  $y_{\text{OE}}$  für  $\text{y}_{\text{OE}}$  ist immer ein indizierter Lösungsvektor, d.h. "spezielle Wert" bedeutet immer "spezielle Wert an Stelle ...".
- Schleifen sind Kopfgesteuert, weshalb die Abfrage von  $y_{\text{tmp}}$  und  $y_{\text{MAX}}$  auf Gleichheit punktuiert.

Pulsbreite:

Erstelle  $y_{\text{tmp}}$  mit  $y_{\text{tmp}} = 0.002 \cdot 0.0$ Erstelle  $y_{\text{MAX}}$  aus  $y_{\text{MAX}} = \text{Berechne } y_{\text{MAX}}$  $y_{\text{MAX}} = \text{Berechne Maximum von } y$  $y_{\text{tmp}} = y_{\text{MAX}} \cdot 0.001$ ; Erstelle tmpArray von ~~start~~ intsSolange  $y_{\text{tmp}} < y_2 \leq y_{\text{tmp}}$ ;  $i = 0$ ;  $i < N$  $\text{tmpArray}[i] = y_{\text{start}}[i]$ 

Sum = 0

Solange  $i < \text{An Länge}(\text{tmpArray})$ ;  $i = 0$ 

sum = sum + tmpArray[i]

 $y_1 = \text{sum} / \text{Länge}(\text{tmpArray})$  $\text{OE}_{\text{MAX}} = 0$ ; index = 0Solange  $i < N$ ;  $i = 0$ ja  $\text{OE}_{\text{MAX}} < \text{OE}[i] ?$ 

nein

 $\text{OE}_{\text{MAX}} = \text{OE}[i]$ ; index = i ~~$a_{12} = \frac{1}{2} \cdot (y_{\text{start}}[\text{index}] - y_{\text{tmp}})$~~ Solange  $\text{tmp} < a_{12}$ ;  $i = 0$ ;  $i < N$ ; indexL = 0

indexL = i

(Weiter auf Seite 11)

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMANN SVEN

Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer
3 0	6 5 1 1	1 0 1	2 0 5 4 1

Sp. 1-2

Sp. 3-6

Sp. 7-9

Sp. 10-14



Bearbeitungsbogen-Nr.

9

Korrekturrand

Sequenzdiagramm Programmablauf:

sd: Programm

Main

Thread A

Thread B

Thread C

<<create>>

<<create>>  
Fordert Daten  
Schickt Daten

<<create>>  
Fordert Daten  
Schickt Daten

\*

Nassi - Schneidermann Einlesen der Daten:

Erstelle Liste aus Paaren mit Integern

Solang die Datei noch Zeilen hat, ~~leeres~~

LESE Zeile z

nein

Fängt z mit "ff" an?

ja

Erstelle ein Paar aus den  
listen Zeichen bis Tabulator  
und aus den Zeichen nach  
dem Tabulator, beides als  
Integer-Wert gepaart.

Speichere Paar in der Liste

Retrunre die Liste

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMANN SVEN

Bereich Berufsnummer IHK-Nummer Prüflingsnummer  
 3 0 6 5 1 1 101 20541

Sp. 1-2

Sp. 3-6

Sp. 7-9

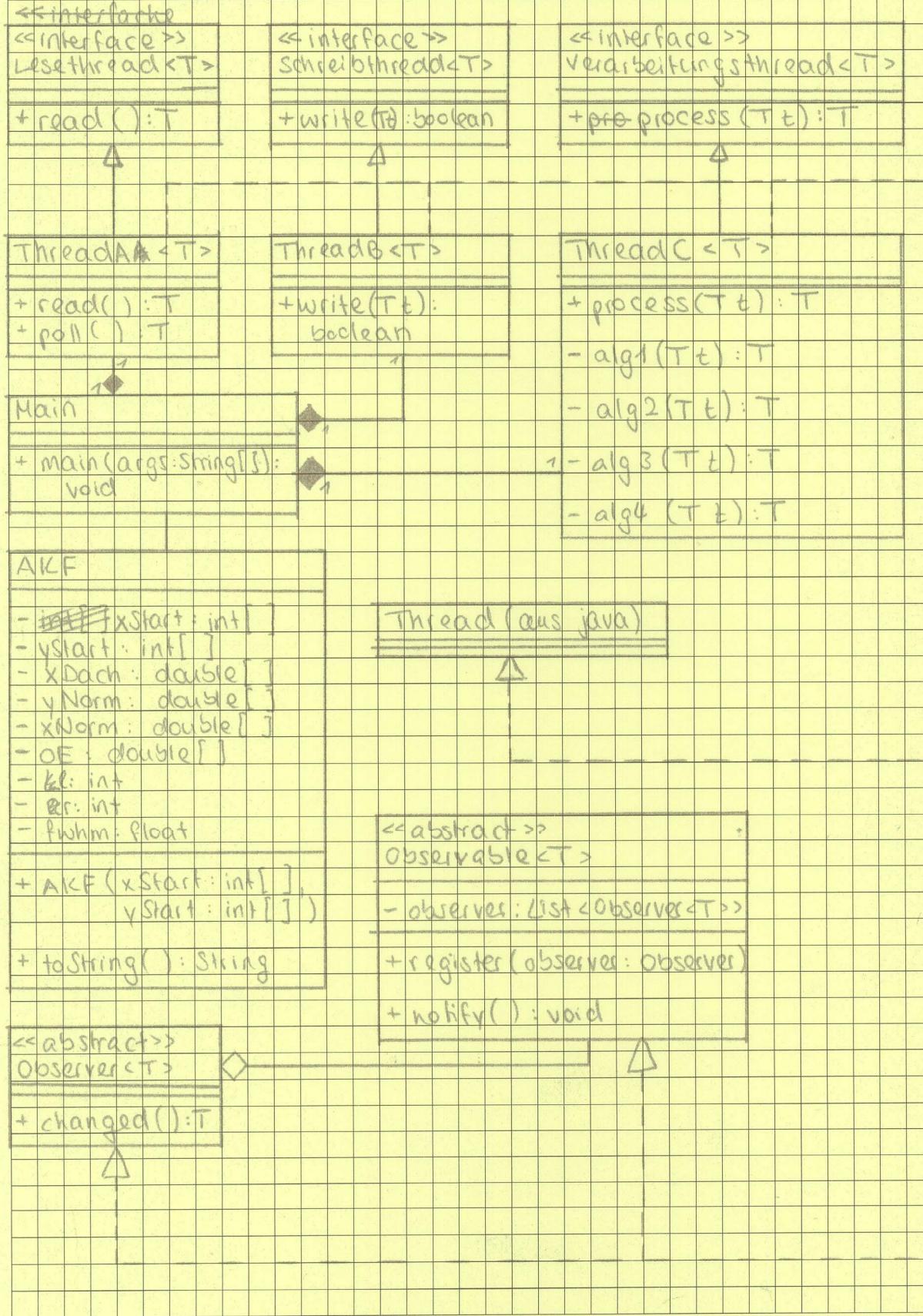
Sp. 10-14



Bearbeitungsbogen-Nr. 10

Korrekturrand

## UML - Klassendiagramm:



# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGHAHN	SVEN				
Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer		
3 0	6 5 1 1	1 0 1	2 0 5 4 1		

Sp. 1-2

Sp. 3-6

Sp. 7-9

Sp. 10-14



Bearbeitungsbogen-Nr.

11

Korrekturrand

Nassi - Shneidermann : Ausgabe der Daten:

Schreibe "FWHM = " + auf. fwhm + ", " + auf. L + ", " + auf. Br

Schreibe "ln" pos int env "

Setze n = ~~rest~~ Länge von xStart

Solang i < n, i = 0

Schreibe xNorm[i] + " \t" + yNorm[i] + " \t" + dE[i] + " \n"

Fortsetzung Nassi - Schneidermann : Pulsbreite:

(Weiter auf Seite)

Solang tmp < a12 ; i = 0 ; i < N ; indexR = 0

indexXL = i

b = yNorm[indexR] - xNorm[indexXL]

Return b, indexR, indexXL

Gedanken zu Glättung der Daten:

Problem:  $k < 2 \vee k > N - 1 - 2$

Lösung: Da die Daten ~~abtrennen~~ sind, so ist somit in einem ~~observeable~~ Intervall von  $[0; 1]$  befindlich, würde es sinnvoll sein, ~~observer: List<Observer>~~

~~- observer: List<Observer> >~~  $k$  so lange zu vergrößern, bis dies wieder  $\geq 2$  ist.

~~+ register(observer: Observer)~~

~~Allle Werte, welche so verloren gehen fallen nicht so stark~~

~~+ notify(): void~~

~~< Abtrennen. Der rechte Rand würde sich ähnlich verhalten,~~

~~Observer < T >~~

~~nur um Verzerrungen aus dem Weg zu gehen. Alle x-Werte~~

~~+ change(): void~~

~~die dadurch „übersprungen“ werden, sollten 0 sein und zu~~

~~der Gesamtzahl n addiert werden, sprich +1 Messwert. Beim~~

~~rechten Rand werden durch  $k > N - 1 - 2$  Messwerte verschluckt.~~

~~Sinnvoll wäre es hier, um das Mittelungsfenster noch~~

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMAHN	SVEN						
Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer				
3 0	6 5 1 1	1 0 1	2 0 5 4 1				

Sp. 1-2

Sp. 3-6

Sp. 7-9

Sp. 10-14



Bearbeitungsbogen-Nr. 1 2

Korrekturrand

relativ richtig abbilden zu können, die „Messwerte“ gleich 1 zu setzen und ebenfalls die Anzahl zu der Gesamtanzahl dazu zu addieren.

Beispiel mit  $N = 50$  Messwerten:

$$n = 1, \Sigma = 0, \bar{x}_n = x_n$$

Beispiel mit  $N = 100$  Messwerten:

$$n = 1, \Sigma = 0, \bar{x}_n = x_n$$

Weiteres zur Konzeptvorstellung:

Die Klasse AICF beinhaltet einige Arrays, welche immer die Größe  $N \times N$  haben werden und von denen eventuell nicht alle immer gebraucht werden. Laut Aufgabenstellung stellt der Thread A (Thread A) alle 0,05 Sekunden einen neuen Datensatz bereit, was wahrscheinlich oft sinnvoller ist als "Push" Methode zu implementieren. Das Weiteren habe ich das so verstanden, dass Thread A von allen Threads nur jeweils eine Instanz gleichzeitig laufen kann, was die Überschreibung und Wiederholung des Datensatzes erklären würde, jedoch in einem Produktivsystem nicht allzu sinnvoll wäre, da ansonsten Datensätze verloren gehen. Der Thread B läuft also nur einmal und verarbeitet auch nur einen Datensatz gleichzeitig, bevor der nächste geholt oder gepusht wird. Eventuell wäre daher auch das Observer - observable Pattern

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMANN SVEN						
Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer			
3 0	6 5 1 1	1 c 1	2 0 5 4 1			
Sp. 1-2	Sp. 3-6	Sp. 7-9	Sp. 10-14			

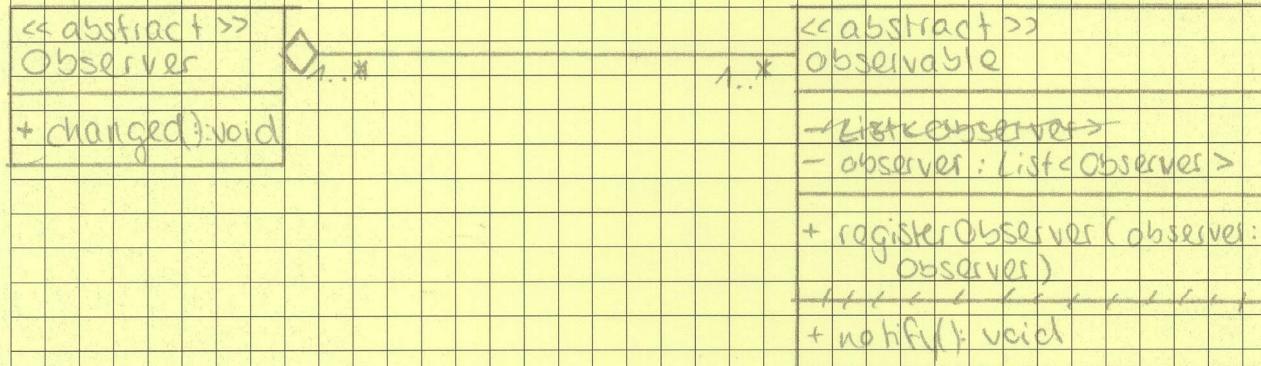


Bearbeitungsbogen-Nr. 13

Korrekturrand

sinnvoll, da der Thread B informiert werden muss, falls Thread A Daten bereitstellt. Genauso das gleich wäre dann natürlich auch für Thread B und C sinnvoll.

UML - Klassendiagramm: Observer - Observable - Pattern:



Das Pattern könnte dann so abgewandelt werden, dass ThreadA ein Observable implementiert, worin der Observer ThreadB registriert ist und über die notify Methode den neuen Datensatz bekommt. Falls der ThreadB allerdings noch arbeitet, wird das einfach ignoriert. ThreadC wird dann selbst auch als Observer im Observable ThreadB registriert und bekommt so die fertig verarbeiteten Daten. Als Abbruchbedingung könnte eventuell nach der ThreadC als Observable implementiert werden und ThreadA als Observer registriert werden, um alle bereits geschriebenen Dateinamen mitzuteilen.  
 Weiterhin habe ich in den UML Diagrammen alles generisch mit dem Typen T glassen, da am besten ein Paar aus Dateinamen und AKF übergeben werden sollte, ~~das~~ das allerdings bricht braucht ThreadB

# Mathematisch-technische/-r Softwareentwickler/-in

# Bearbeitungsbogen

Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

BERGMANN SVEN													
Bereich		Berufsnummer		IHK-Nummer		Prüflingsnummer							
3	0	6	5	1	1	1	0	1	2	0	5	4	1
Sp. 1-2	Sp. 3-6	Sp. 7-9	Sp. 10-14										



Bearbeitungsbogen-Nr.

14

Korrekturrand

Zum Beispiel nicht die komplette Klasse schicken,  
weshalb der generische Typ wesentlich sinnvoll ist.  
Theoretisch braucht ThreadC dann eben auch nur  
zwei Strings als Übergabe.