
Class ListElement

```
java.lang.Object
|
+--Al.ListElement
```

< [Methods](#) >

```
public class ListElement
extends java.lang.Object
```

Methods

getElem

```
public java.lang.Object getElem()
```

Gibt das gespeicherte Objekt zurueck

Returns:

Gespeichertes Objekt

getNext

```
public ListElement getNext()
```

Gibt die Referenz auf den Rest der Liste zurueck

Returns:

Referenz auf den Rest der Liste

insert

```
public int insert(java.lang.Object x,  
                  int n,  
                  int i)
```

Wenn $n == 0$ ist, dann wird das Objekt x hinter dieses ListElement gefuegt. Ansonsten wird die Aufgabe an das Folgende ListElement weitergereicht und n um 1 dekrementiert.

Parameters:

x - Zu Speicherndes Objekt
 n - Indexzaehler fuer den Speicherort
 i - Dereferenzierungs-Zaehler - Hilfsvariable fuer ListImpl

Returns:

Anzahl der Dereferenzierungen

Interface Liste

< [Methods](#) >

```
public interface Liste
```

Methods

cons

```
public void cons(java.lang.Object x)
```

Ein Element vorne anfüegen

Parameters:

x - Zuzufuegendes Element

getStepCounter

```
public int getStepCounter()
```

Gibt die Anzahl an Dereferenzierungen aus

Returns:

Anzahl an Dereferenzierungen

head

```
public java.lang.Object head()
```

Das vorderste Element von der Liste entfernen und zurueckgeben

Returns:

Erstes Element der Liste oder bei leerer Liste eine NullPointerException

insert

```
public boolean insert(java.lang.Object x,  
                      int n)
```

Nach n Elementen x in die Liste einfuegen

Parameters:

x - Einzufuegendes Element

n - Position in der Liste, an der das Element eingefuegt werden soll - beginnend mit 0

Returns:

Wurde das Element erfolgreich eingefuegt (True) oder nicht (False)

isempty

```
public boolean isempty()
```

Ist die Liste leer oder nicht

Returns:

True (leer)oder False (nicht leer)

length

```
public int length()
```

Gibt die Laenge der Liste aus

Returns:

Laenge der Liste

resetStepCounter

```
public void resetStepCounter()
```

Setzt den Dereferenzierungscouter zurueck

top

```
public java.lang.Object top()
```

Das vorderste Element der Liste ausgeben (ohne entfernen)

Returns:

Erstes Element der Liste

Class Messung

```
java.lang.Object
|
+--Al.Messung
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class Messung
extends java.lang.Object
```

Klasse zum Speichern und Auswerten von Messreihen von Zahlen

Author:

Steffen Giersch, Birger Kamp, Maria Luedemann

Fields

akku_avg

```
public double akku_avg
    Summe aller Messungen fuer den Mittelwert
```

mssg

```
public java.util.List mssg
    Liste aller Messungen
```

Constructors

Messung

```
public Messung()
    Constructor fuer die Messung
```

Methods

add

```
public void add(double n)
```

Fuegt einen Messwert hinzu

Parameters:

n - Hinzuzufuegender Messwert

average

```
public double average()
```

Errechnet den Mittelwert aller Messungen

Returns:

Mittelwert aller Messungen

varianz

```
public double varianz()
```

Errechnet die Varianz der Messungen

Returns:

Varianz der Messungen

Class ProcessingMain

```
java.lang.Object
|
+-- java.awt.Component
|   |
|   +-- java.awt.Container
|       |
|       +-- java.awt.Panel
|           |
|           +-- java.applet.Applet
|               |
|               +-- processing.core.PApplet
|                   |
|                   +-- A1.ProcessingMain
```

All Implemented Interfaces:

java.awt.MenuContainer, java.awt.event.FocusListener, java.awt.event.KeyListener, java.awt.event.MouseListener, java.awt.event.MouseMotionListener, java.awt.event.MouseWheelListener, java.awt.image.ImageObserver, java.io.Serializable, java.lang.Runnable, javax.accessibility.Accessible, processing.core.PConstants

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class ProcessingMain
extends processing.core.PApplet
```

Wird zum Ausfuehren des Processing-Graphen benutzt. Ist unabhaengig vom Rest der Aufgabe.

Author:

Steffen Giersch, Birger Kamp, Maria Luedemann

Fields

ANZAHLMESSUNGEN

```
public static final int ANZAHLMESSUNGEN
    Anzahl der durchgefuehrten Messungen pro Bild
```

CIRCLESIZE

```
public static final float CIRCLESIZE
    Groesse der angezeigten Kreise
```

FPS

```
public static final float FPS
    Bilder pro Sekunde
```

LISTENLAENGE

```
public static final int LISTENLAENGE  
    Laenge der erstellten Listen
```

RIGHTSHIFT

```
public static final int RIGHTSHIFT  
    Rechtsshift fuer das Koordinatensystem in Pixeln
```

Constructors

ProcessingMain

```
public ProcessingMain()
```

Methods

draw

```
public void draw()
```

Overrides:

draw in class processing.core.PApplet

main

```
public static void main(java.lang.String[] args)
```

setup

```
public void setup()
```

Overrides:

setup in class processing.core.PApplet

Class main

```
java.lang.Object
|
+--main
```

< [Constructors](#) > < [Methods](#) >

```
public class main
extends java.lang.Object
```

Constructors

main

```
public main()
```

Methods

main

```
public static void main(java.lang.String[] args)
```

Parameters:

args -

INDEX

A

[add](#) ... 5
[akku_avg](#) ... 4
[average](#) ... 5
[ANZAHLMESSUNGEN](#) ... 6

C

[cons](#) ... 2
[CIRCLESIZE](#) ... 6

D

[draw](#) ... 7

F

[FPS](#) ... 6

G

[getElem](#) ... 1
[getNext](#) ... 1
[getStepCounter](#) ... 2

H

[head](#) ... 3

I

[insert](#) ... 2
[insert](#) ... 3
[isempty](#) ... 3

L

[length](#) ... 3
[Liste](#) ... 2
[ListElement](#) ... 1
[LISTENLAENGE](#) ... 7

M

[main](#) ... 7
[main](#) ... 8
[main](#) ... 8
[main](#) ... 8
[mssg](#) ... 4
[Messung](#) ... 4
[Messung](#) ... 4

P

[ProcessingMain](#) ... 6
[ProcessingMain](#) ... 7

R

[resetStepCounter](#) ... 3
[RIGHTSHIFT](#) ... 7

S

[setup](#) ... 7

T

[top](#) ... 4

V

[varianz](#) ... 5