
Class AbstractQuadMatrix

```
java.lang.Object
|
+--A2.AbstractQuadMatrix
```

All Implemented Interfaces:

[QuadMatrix](#)

Direct Known Subclasses:

[QuadMatrixArrayImpl](#), [QuadMatrixArrayListImpl](#), [QuadMatrixListImpl](#)

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

public abstract class **AbstractQuadMatrix**
extends java.lang.Object
implements [QuadMatrix](#)

Constructors

AbstractQuadMatrix

```
public AbstractQuadMatrix()
```

Methods

add

```
public QuadMatrix add(QuadMatrix mtx)
```

equals

```
public boolean equals(java.lang.Object mtx)
```

Overrides:

equals in class java.lang.Object

getSize

```
public int getSize()
```

mul

```
public QuadMatrix mul(QuadMatrix mtx)
```

mul

```
public QuadMatrix mul(double n)
```

pow

```
public QuadMatrix pow(int n)
```

time

```
public int time()
```

timeReset

```
public void timeReset()
```

Interface QuadMatrix

< [Methods](#) >

```
public interface QuadMatrix
```

Author:

Steffen Giersch, Birger Kamp, Maria Janna Martina Luedemann

Methods

add

```
public QuadMatrix add(QuadMatrix mtx)
```

Matrixaddition ueber zwei QuadMatrix Implementationen

Parameters:

mtx - Zu this zu addierende QuadMatrix

Returns:

Ergebnis der Matrixaddition (QuadMatrix)

equals

```
public boolean equals(java.lang.Object mtx)
```

Gleichheitsfunktion fuer QuadMatrix

Parameters:

mtx - Zu vergleichendes Objekt

Returns:

Sind die Matrizen gleich (true) oder nicht (false)

Overrides:

equals in class java.lang.Object

get

```
public double get(int m,  
                  int n)
```

Gibt das Element am eingegebenen Index zurueck

Parameters:

m - Zeilenindex beginnend mit 1
n - Spaltenindex beginnend mit 1

Returns:

Element an diesem Index

getSize

```
public int getSize()
```

Gibt die Groesse (int) der Matrix zurueck

Returns:

Groesse der Matrix (int)

init

```
public QuadMatrix init(int n)
```

Gibt eine neue Instanz der QuadMatrix Implementation mit der Groesse n (int) zurueck

Parameters:

n - Groesse der zu erzeugenden Matrix

Returns:

Neue Instanz einer QuadMatrix Implementation

mul

```
public QuadMatrix mul(QuadMatrix mtx)
```

Matrixmultiplikation in der Form this*mtx

Parameters:

mtx - Mit this zu multiplizierende Matrix

Returns:

Ergebnis der Matrixmultiplikation

mul

```
public QuadMatrix mul(double n)
```

Skalarmultiplikation fuer QuadMatrix

Parameters:

n - Skalar mit dem multipliziert werden soll

Returns:

Ergebnis der Skalarmultiplikation

pow

```
public QuadMatrix pow(int n)
```

Potenzierung von this zum Exponenten n (int)

Parameters:

n - Potenz (int)

Returns:

Ergebnis der Matrixmultiplikation

set

```
public void set(int m,  
               int n,  
               double x)
```

Setzt das Element am eingegebenen Index auf den Wert x (double)

Parameters:

m - Zeilenindex beginnend mit 1
n - Spaltenindex beginnend mit 1
x - Element an diesem Index

setGen

```
public void setGen(int m,  
                  int n,  
                  double x)
```

space

```
public int space()
```

Gibt den benoetigten Platz dieser Implementation wieder

Returns:

Benoetigter Platz (int)

time

```
public int time()
```

Gibt die benoetigte Anzahl der Dereferenzierungen vom letzten Reset bis jetzt wieder

Returns:

Anzahl der Dereferenzierungen

timeReset

```
public void timeReset()
```

Resetet die Anzahl der Dereferenzierungen

Class QuadMatrixArrayImpl

```
java.lang.Object
|
+-- AbstractQuadMatrix
|
+-- A2.QuadMatrixArrayImpl
```

All Implemented Interfaces:

[QuadMatrix](#)

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

```
public class QuadMatrixArrayImpl
extends AbstractQuadMatrix
```

Constructors

QuadMatrixArrayImpl

```
public QuadMatrixArrayImpl(int n)
```

Methods

get

```
public double get(int m,  
                  int n)
```

init

```
public QuadMatrix init(int n)
```

set

```
public void set(int m,  
                int n,  
                double x)
```

setGen

```
public void setGen(int m,  
                   int n,  
                   double x)
```

space

```
public int space()
```

Class QuadMatrixArrayListImpl

```
java.lang.Object  
|  
+--AbstractQuadMatrix  
|  
+--A2.QuadMatrixArrayListImpl
```

All Implemented Interfaces:

[QuadMatrix](#)

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

```
public class QuadMatrixArrayListImpl
```

extends [AbstractQuadMatrix](#)

Constructors

QuadMatrixArrayListImpl

```
public QuadMatrixArrayListImpl(int n)
```

Methods

get

```
public double get(int m,  
                 int n)
```

init

```
public QuadMatrix init(int n)
```

set

```
public void set(int m,  
               int n,  
               double x)
```

setGen

```
public void setGen(int m,  
                  int n,  
                  double x)
```

space

```
public int space()
```

Class QuadMatrixGenerator

```
java.lang.Object
|
+--A2.QuadMatrixGenerator
```

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

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

Constructors

QuadMatrixGenerator

```
public QuadMatrixGenerator()
```

Methods

random

```
public static QuadMatrix random(QuadMatrix mtx,
                                int x,
                                double n,
                                double m)
```

Gibt eine Zufallsgenerierte QuadMatrix der Implementation von mtx wieder

Parameters:

mtx - Typ der Matrix, der wiedergegeben werden soll
x - Groesse der Matrix
n - Wahrscheinlichkeit, dass ein Element nicht 0.0 ist
m - Reichweite von 0..m die ein Element haben kann

Returns:

Generierte Matrix

Class QuadMatrixListImpl

```
java.lang.Object
|
+--AbstractQuadMatrix
    |
    +--A2.QuadMatrixListImpl
```

All Implemented Interfaces:

[QuadMatrix](#)

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

```
public class QuadMatrixListImpl
extends AbstractQuadMatrix
```

Constructors

QuadMatrixListImpl

```
public QuadMatrixListImpl(int n)
```

Methods

get

```
public double get(int m,
                  int n)
```

init

```
public QuadMatrix init(int n)
```

set

```
public void set(int m,
                int n,
                double x)
```

setGen

```
public void setGen(int m,  
                  int n,  
                  double x)
```

space

```
public int space()
```

Class main

```
java.lang.Object  
|  
+--A2.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](#) ... 1
[add](#) ... 3
[AbstractQuadMatrix](#) ... 1
[AbstractQuadMatrix](#) ... 1

E

[equals](#) ... 1
[equals](#) ... 3

G

[get](#) ... 3
[get](#) ... 7
[get](#) ... 8
[get](#) ... 10
[getSize](#) ... 2
[getSize](#) ... 4

I

[init](#) ... 4
[init](#) ... 7
[init](#) ... 8
[init](#) ... 10

M

[main](#) ... 11
[main](#) ... 11
[main](#) ... 11
[mul](#) ... 2
[mul](#) ... 2
[mul](#) ... 4
[mul](#) ... 4

P

[pow](#) ... 2
[pow](#) ... 5

Q

[QuadMatrix](#) ... 2
[QuadMatrixArrayImpl](#) ... 6
[QuadMatrixArrayImpl](#) ... 6
[QuadMatrixArrayListImpl](#) ... 7
[QuadMatrixArrayListImpl](#) ... 8
[QuadMatrixGenerator](#) ... 9
[QuadMatrixGenerator](#) ... 9
[QuadMatrixListImpl](#) ... 10
[QuadMatrixListImpl](#) ... 10

R

[random](#) ... 9

S

[set](#) ... 5
[set](#) ... 7
[set](#) ... 8
[set](#) ... 10
[setGen](#) ... 5
[setGen](#) ... 7
[setGen](#) ... 8
[setGen](#) ... 11
[space](#) ... 5
[space](#) ... 7
[space](#) ... 8
[space](#) ... 11

T

[time](#) ... 2
[time](#) ... 6
[timeReset](#) ... 2
[timeReset](#) ... 6