

[Skip navigation links](#)

- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

Class `CompDijkstraPath<E extends Edge>`

- `java.lang.Object`
 - `CompDijkstraPath<E>`

- Type Parameters:
 - `E` -

All Implemented Interfaces:

`java.lang.Comparable<CompDijkstraPath<E>>`

```
public class CompDijkstraPath<E extends Edge>
    extends java.lang.Object
    implements java.lang.Comparable<CompDijkstraPath<E>>
```

Ett köelement som används för Dijkstras algoritm

Author:

Erik Öhm & Paula Eriksson Imable

- ◦ **Constructor Summary**

Constructors

Constructor and Description

[CompDijkstraPath](#)(`int` node, `double` cost, `java.util.ArrayList`<[E](#)> path)

Skapar ett köelement.

- **Method Summary**

All Methods [Instance Methods](#) [Concrete Methods](#)

Modifier and Type	Method and Description
<code>int</code>	compareTo (CompDijkstraPath < E > dijkstraElement) Jämför med ett annat köelement.
<code>double</code>	getCost ()
<code>int</code>	getNode ()
<code>java.util.ArrayList</code> < E >	getPath ()

■ Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

- ◦ **Constructor Detail**

■ **CompDijkstraPath**

```
public CompDijkstraPath(int node,  
                        double cost,  
                        java.util.ArrayList<E> path)
```

Skapar ett köelement. Får en nod, kostnaden dit från startnoden samt vägen dit från startnoden

Parameters:

node - noden
cost - kostnaden dit från startnoden
path - vägen dit från startnoden

Throws:

java.lang.IndexOutOfBoundsException - om cost är mindre än noll

○ **Method Detail**

■ **getPath**

```
public java.util.ArrayList<E> getPath()
```

Returns:

vägen till noden från startnoden

■ **getCost**

```
public double getCost()
```

Returns:

kostnaden till noden via vägen den tagit

■ **getNode**

```
public int getNode()
```

Returns:

noden

■ **compareTo**

```
public int compareTo(CompDijkstraPath<E> dijkstraElement)
```

Jämför med ett annat köelement.

Specified by:

compareTo in interface java.lang.Comparable<[CompDijkstraPath<E>](#) extends [Edge](#)>>

Parameters:

dijkstraElement - det andra elementet

Throws:

java.lang.NullPointerException - om det andra elementet är null

[Skip navigation links](#)

- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)