

グラフを表すデータ構造 JAVAでの実装

グラフの構造を記述するクラス

- パッケージgraphLib
- graphLib.GraphBase
 - グラフ全体
 - 弧と頂点の関連付け
- graphLib.Graph
 - 弧や頂点の追加削除
 - グラフのコピー、ファイル入出力
- graphLib.Vertex
 - 頂点
 - 頂点を2次元面に表示するための座標
- graphLib.Arc
 - 弧

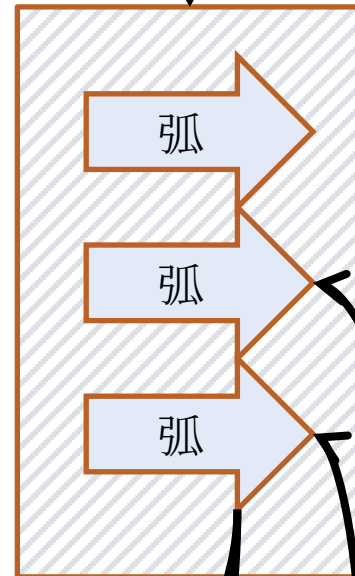
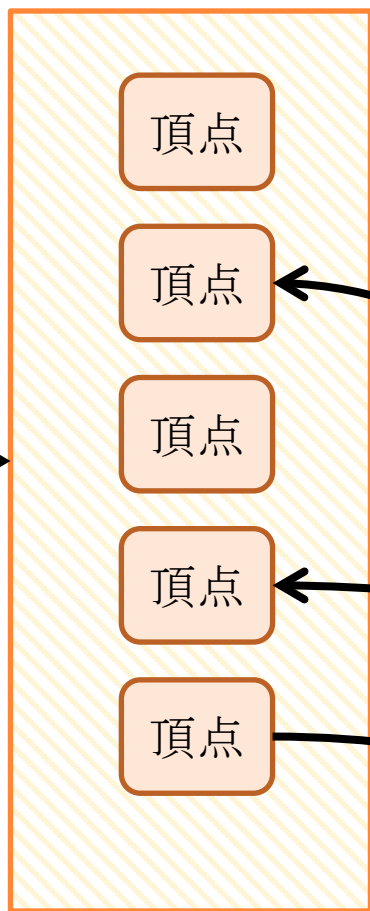


クラスの関係

グラフ

頂点一覧

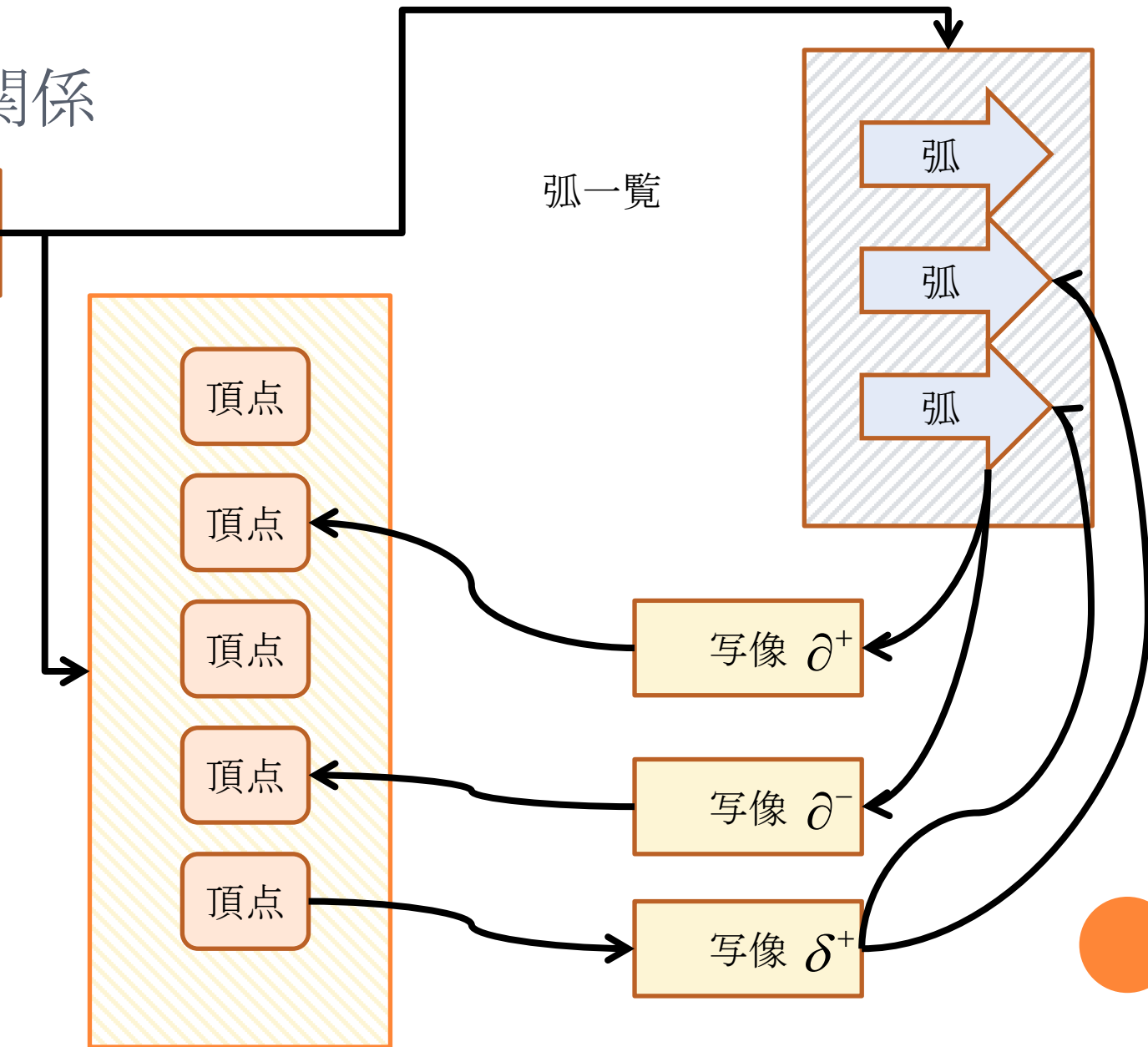
弧一覧

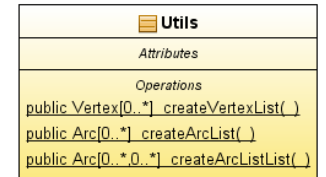
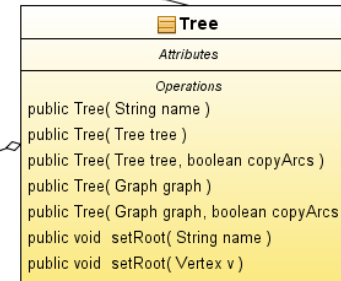
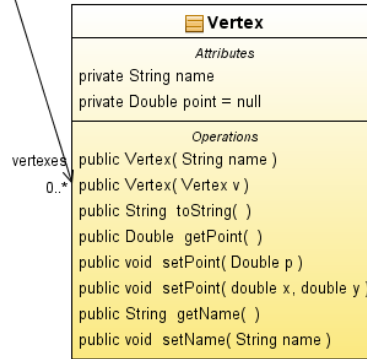
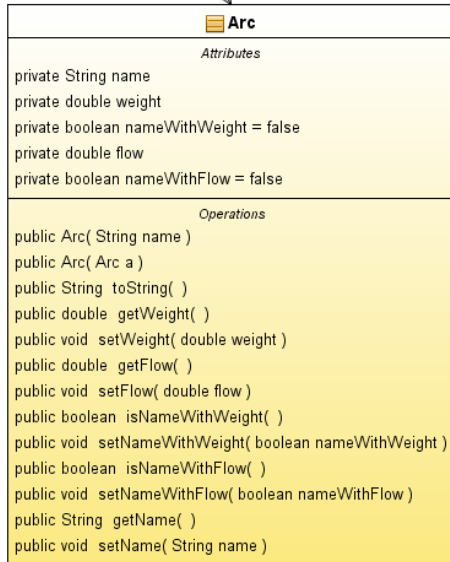
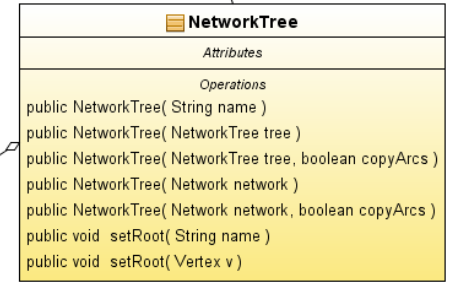
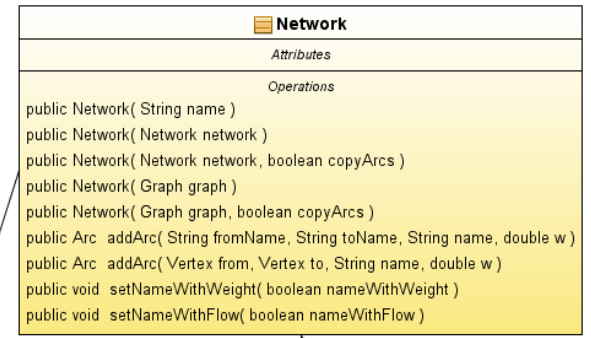
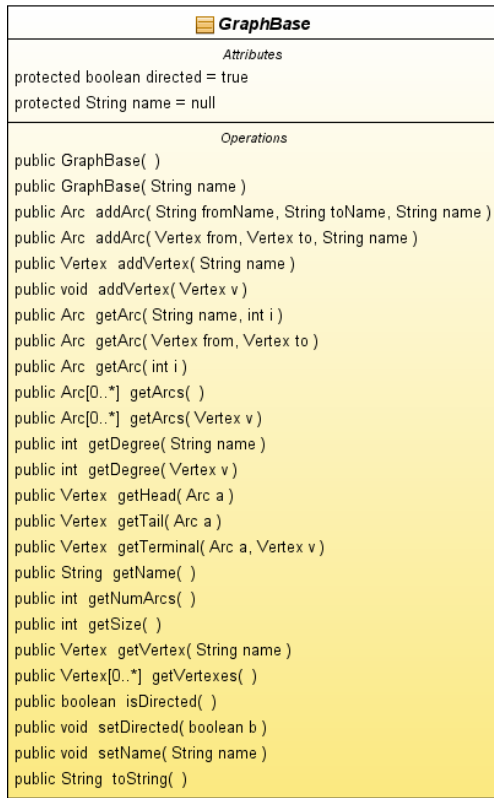


写像 ∂^+

写像 ∂^-

写像 δ^+





arcs 0..*

vertices 0..*

root

root

GRAPHBASEクラス:フィールド

/** グラフのラベル*/

protected String name = null;

/** 頂点の一覧*/

protected List<Vertex> vertexes = null;

/** 弧の一覧*/

protected List<Arc> arcs = null;

/** 有向グラフであるか*/

protected boolean directed = true;

/** 弧からその始点への写像*/

protected HashMap<Arc, Vertex> a2vHead = null;

∂^+

/** 弧からその終点への写像*/

protected HashMap<Arc, Vertex> a2vTail = null;

∂^-

/** 頂点を始点とする弧のリスト*/

protected HashMap<Vertex, List<Arc>> v2a = null;

δ^+



GRAPHクラス: 主要メソッド

<code>public Graph(String name)</code>	コンストラクタ
<code>public Graph(final Graph graph)</code>	コピーコンストラクタ
<code>public Vertex addVertex(String name)</code>	頂点追加
<code>public void addVertex(Vertex v)</code>	頂点追加
<code>public Arc addArc(String fromName, String toName, String name)</code>	弧追加
<code>public Arc addArc(Vertex from, Vertex to, String name)</code>	弧追加
<code>public void setDirected(boolean b)</code>	有向・無向を設定
<code>public List<Vertex> getVertexes()</code>	頂点一覧を取得
<code>public List<Arc> getArcs()</code>	弧一覧を取得
<code>public List<Arc> getArcs(Vertex v)</code>	指定した頂点を視点とする 弧の一覧
<code>public Vertex getHead(Arc a)</code>	指定した弧の始点
<code>public Vertex getTail(Arc a)</code>	指定した弧の終点



VERTEXクラス:フィールド

/** 頂点のラベル */

private String name;

/** 頂点の位置座標 */

private Point2D.Double point = null;



VERTEXクラス: 主要メソッド

`public Vertex(String name)` コンストラクタ
`public Point2D.Double getPoint()` 頂点の座標を取得
`public void setPoint(Point2D.Double p)` 頂点の座標を設定



ARCクラス:フィールド

*/** 弧のラベル */*

private String name;

*/** 弧の重み */*

private double weight;

private boolean nameWithWeight = false;

*/** 弧を流れる流量 */*

private double flow;

private boolean nameWithFlow = false;



グラフの定義

○ graphLib.Graphを拡張

```
public class Graph0 extends Graph {  
  
    public Graph0(String name) {  
        super(name);  
        int n = 6;  
        //頂点の生成  
        graphLib.Vertex vList[] = new graphLib.Vertex[n];  
        for (int i = 0; i < n; i++) {  
            vList[i] = new graphLib.Vertex(String.valueOf(i));  
            addVertex(vList[i]);  
        }  
        //頂点の表示座標を設定  
        double d = 100.;  
        vList[0].setPoint(d, d);  
        ....  
        int k = 0;  
        addArc(vList[0], vList[1], String.valueOf(k)); k++;  
        addArc(vList[0], vList[1], String.valueOf(k)); k++;  
    }  
}
```



グラフ表示のメインプログラム

```
/**
 * 表示の開始
 * @param evt 表示開始のボタンイベント
 */
private void drawActionPerformed(java.awt.event.ActionEvent evt) {
    graphExample0 = new graphSamples.Graph0(title);
    /** グラフの定義 */
    panel.setGraph(graphExample0);
    /** グラフをパネルに設定 */
    panel.mkImage();
    /** 作図イメージ生成 */
    setTitle(graphExample0.toString());
    /** タイトル設定 */
    repaint();
    /** 全体再描画 */
}
```



