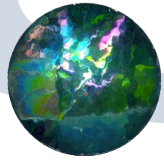
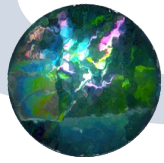


# グラフを表すデータ構造 Javaでの実装



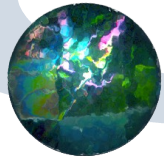
# なぜJavaを使うか

- グラフの実装
  - 点、弧及びその関連を記述する
  - 点の数、弧の数を柔軟に変える必要あり
  - グラフ探索など、リンクをたどる必要あり
- オブジェクト指向言語が向いている
  - オブジェクト数の柔軟な変更
  - 再帰的関数・メソッド



# なぜJavaを使うかII

- リストなどの豊富なライブラリ
  - `java.util.Vector`
- 使い易い開発環境
  - クラス管理
  - プログラミング支援
- グラフィックスを容易に作成可能
  - `javax.swing`コンポーネント
- Osに依らずにプログラミング可能
- 豊富な文献



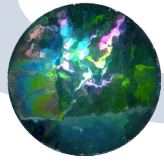
# Java開発環境

- NetBeans

- <http://www.netbeans.org/>
- <http://www.netbeans.jp/>
- GUI開発が容易
- UMLとの連携ができる

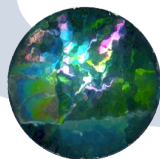
- Eclipse

- <http://www.eclipse.org/>

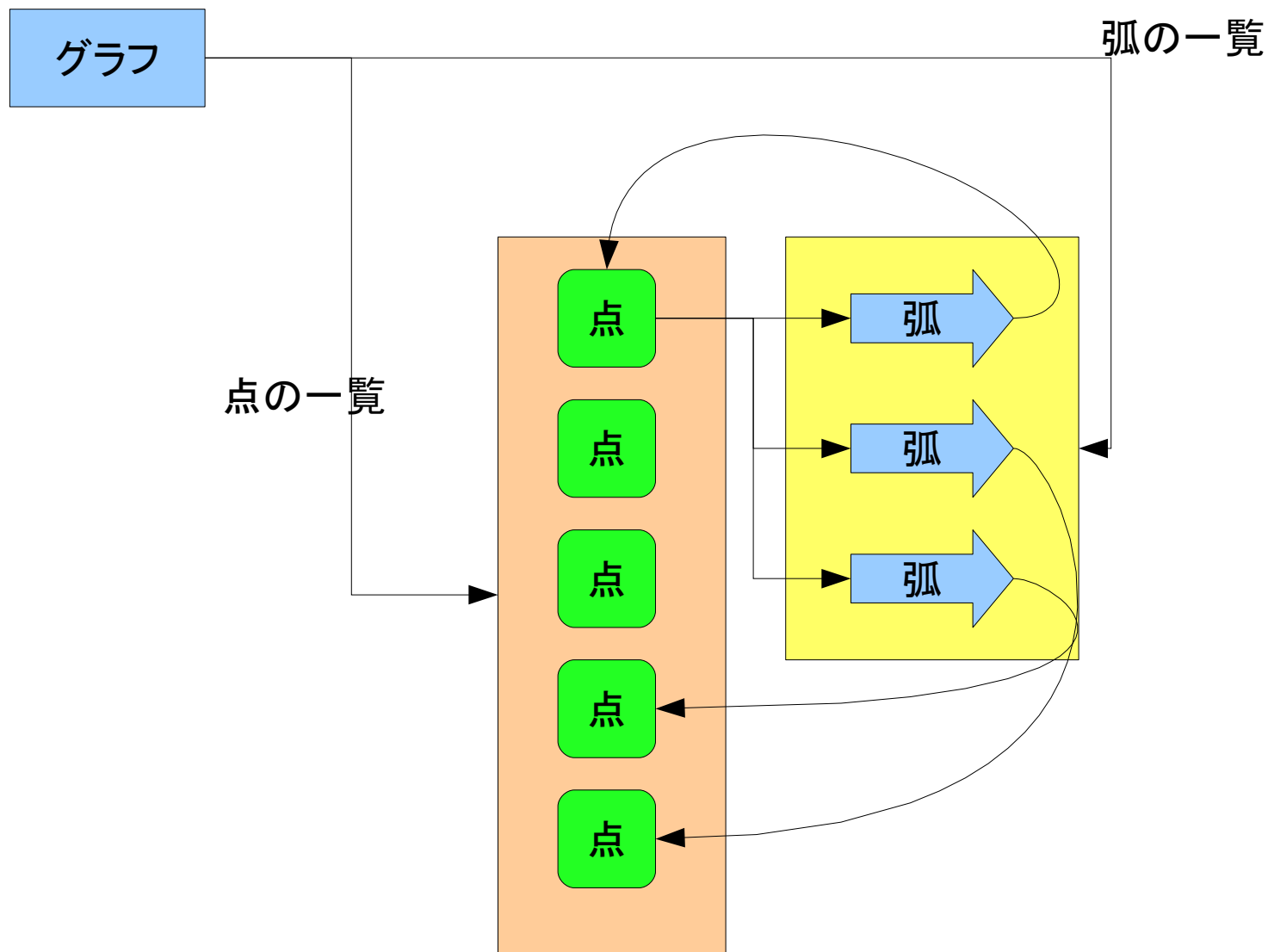


# グラフの構造を記述するクラス

- graphLib.graph
  - グラフ全体
- graphLib.Vertex
  - 点
  - 点を始点とする弧
  - 点を2次元面に表示するための座標
- graphLib.Arc
  - 弧の始点と終点



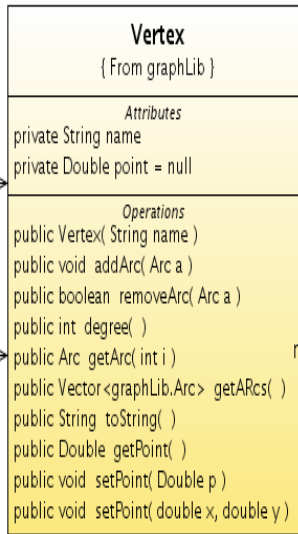
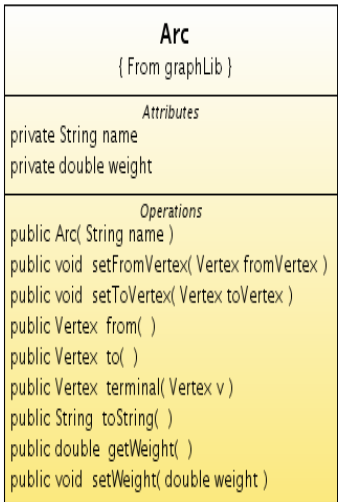
# グラフの構造



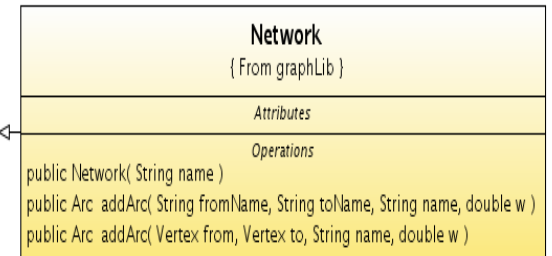
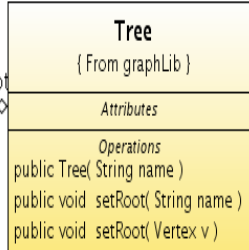
**Vector<graphLib.Arc>**  
{ From graphLib }

**Vector<Vertex>**  
{ From graphLib }

vertices



**Vector<Arc>**  
{ From graphLib }



toVertex

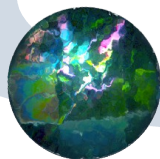
fromVertex

arcs

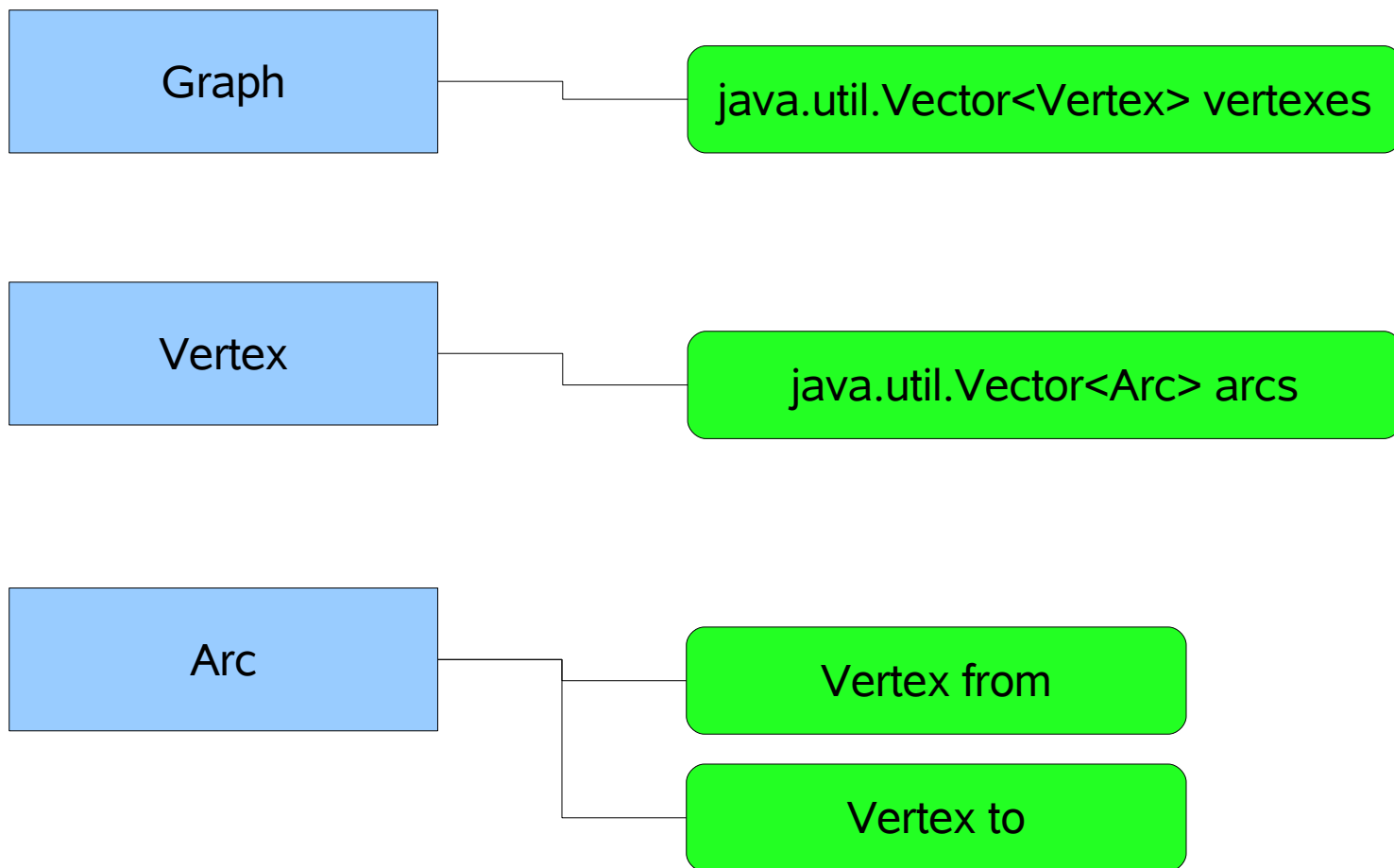
arcs

root

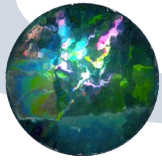
←



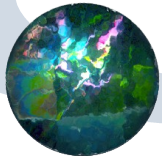
# 主要クラス間の関係



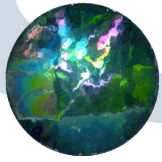




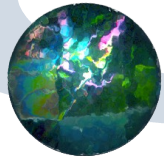
```
package graphLib;
public class Graph {
    /** グラフのラベル*/
    protected String name=null;
    /** 頂点の一覧 */
    protected java.util.Vector<Vertex> vertexes=null;
    /** 弧の一覧 */
    protected java.util.Vector<Arc> arcs=null;
}
```



```
package graphLib;
public class Vertex {
    /** 頂点のラベル */
    private String name;
    /** 頂点を始点とする弧のリスト */
    private java.util.Vector<Arc> arcs=null;
}
```

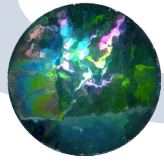


```
package graphLib;
public class Arc {
    /** 始点 */
    private Vertex fromVertex=null;
    /** 終点 */
    private Vertex toVertex=null;
    /** 弧のラベル */
    private String name;
}
```



## グラフを定義する

```
graphLib.Graph graph=new graphLib.Graph(タイトル);
graphLib.Vertex vertexes[]=new graphLib.Vertex[n];
for(int i=0;i<n;i++){//頂点を作る
    vertexes[i] = new graphLib.Vertex(頂点の名前);
    graph.addVertex(vertexes[i]);
}
//弧の定義(i とjをつなぐ)
graph.addArc(vertexes[i],verteces[j],弧の名前);
....
```



# グラフを表示するためのクラス

- `graphDraw.graphPanel`
  - グラフを表示するためのパネル
- `graphDraw.Node`
  - 点を表示する
- `graphDraw.Arrow`
  - 弧を表示する

## graphPanel

{ From graphDraw }

### Attributes

```
private Graph graph = null
private Image image = null
private double r = 10.
```

### Operations

```
public graphPanel( )
public void setGraph( Graph graph )
public void mkImage( )
public void paintComponent( Graphics g )
public void setR( double r )
private void initComponents( )
```

## Node

{ From graphDraw }

### Attributes

```
private Double center = null
private double r = 0.
private Double ellipse = null
private String name = null
private Color nodeColor = java.awt.Color.CYAN
private Color fg = java.awt.Color.BLACK
```

### Operations

```
public Node( Double center, double r, String name )
public void draw( Graphics2D g )
```

## Arrow

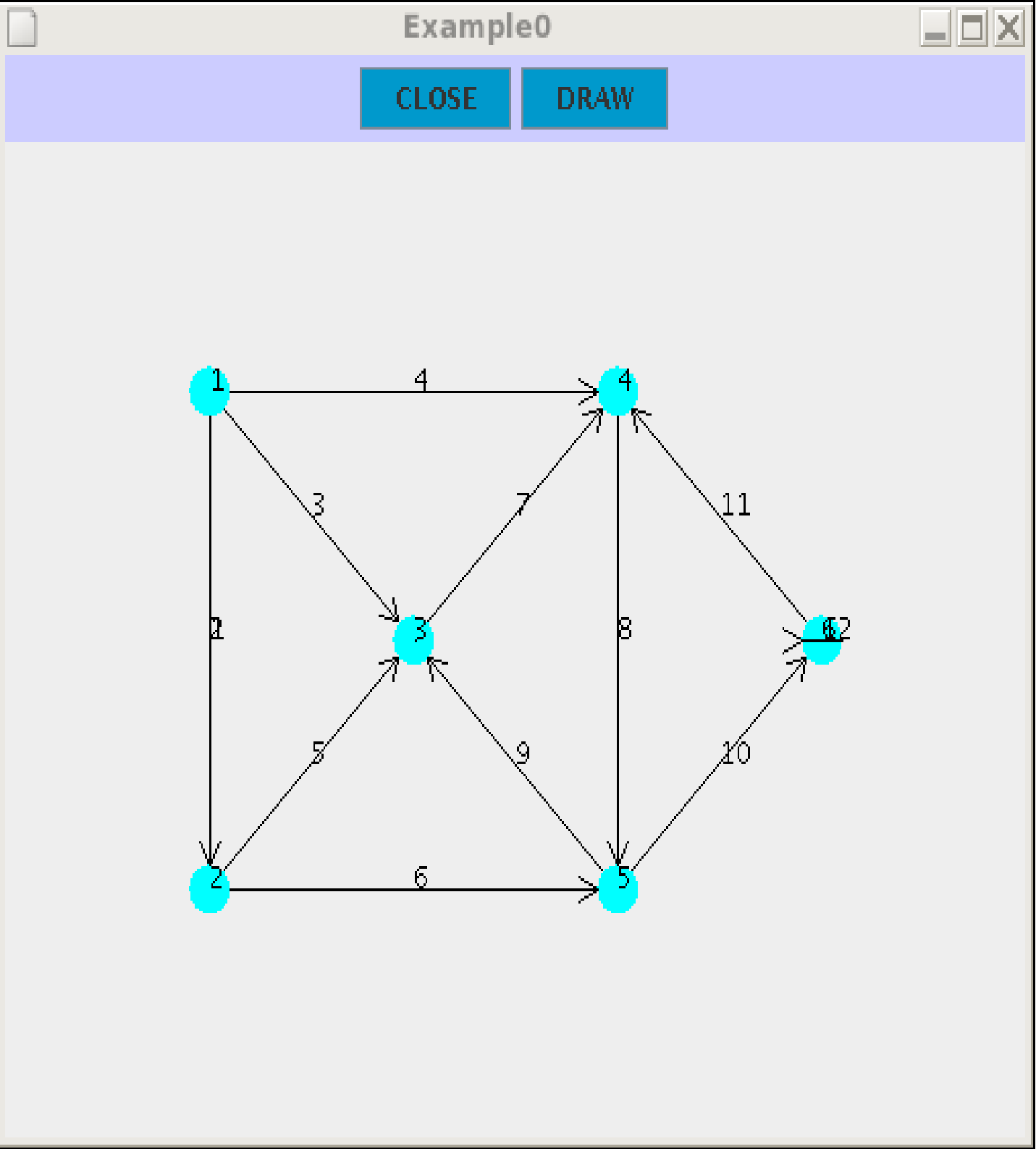
{ From graphDraw }

### Attributes

```
private Double head = null
private Double tail = null
private double r = 0.
private double angle = 0.
private double s = 10.
private String name = null
private boolean directed = true
private Color color = java.awt.Color.BLACK
```

### Operations

```
public Arrow( Double tail, Double head, double r, String name )
public Arrow( Double tail, Double head, double r, String name, boolean b )
public void setHead( Double head )
public void setTail( Double tail )
public void setR( double r )
public double getAngle( )
public void draw( Graphics2D g )
public void setS( double s )
public boolean isDirected( )
public void setDirected( boolean directed )
```



例

