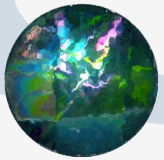
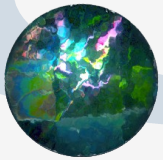
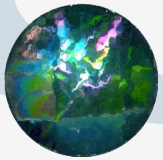


例：結合写像型最適速度模型



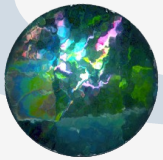
シミュレーションの例題

- モデルとGUIを分離する
- モデル
  - 高速に実行する
  - 計算結果をファイルに書く
- GUI
  - 動作をチェックする
  - デモンストレーションに使う

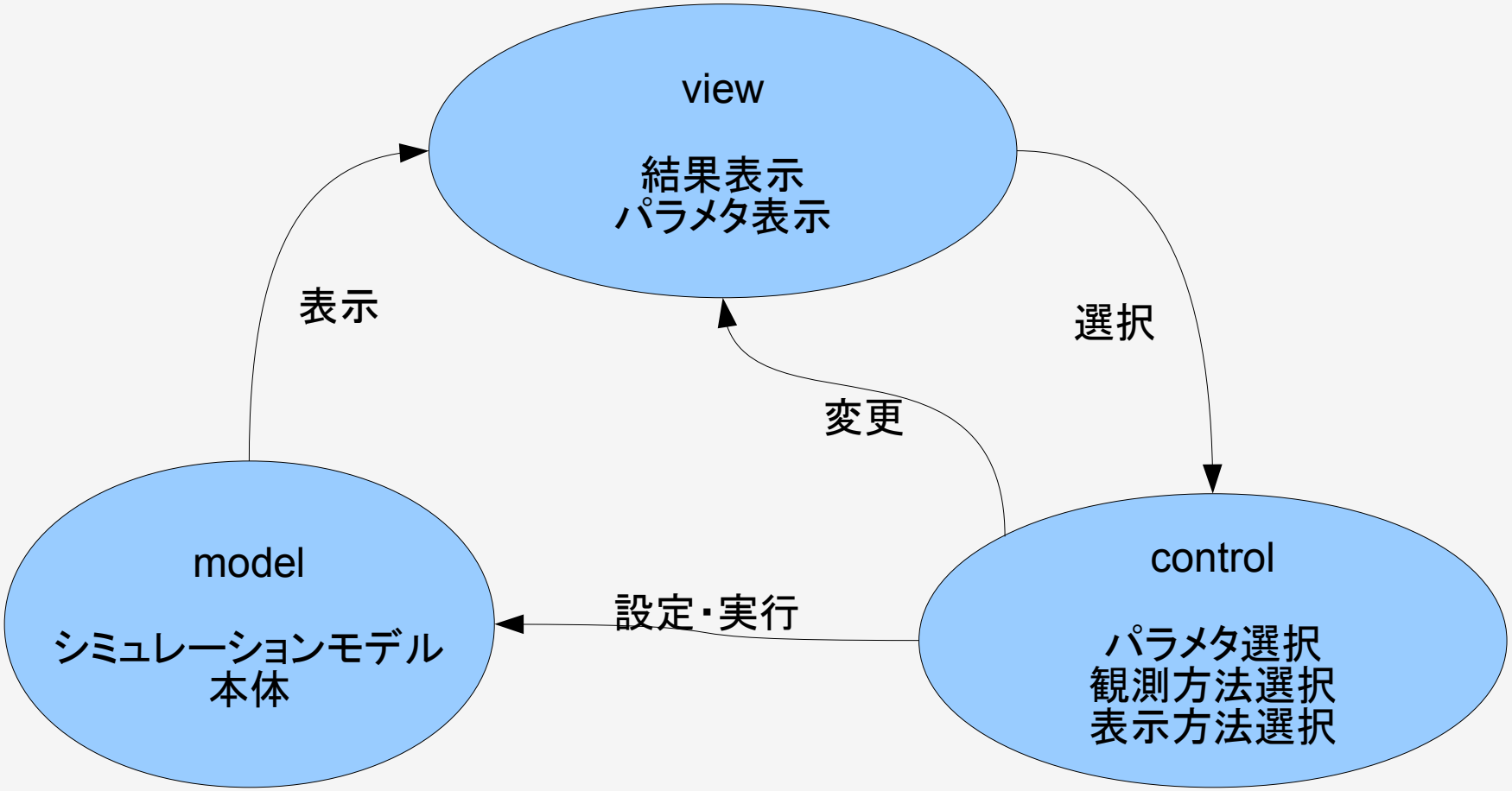


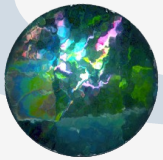
MVC (Model-View-Control)

- OOPプログラムの指針
- MVCを分離することで、開発効率を上げる
- Model
  - シミュレーションモデル、ビジネスロジックなど
- View
  - ユーザインターフェイス
- Control
  - UIからの操作への反応



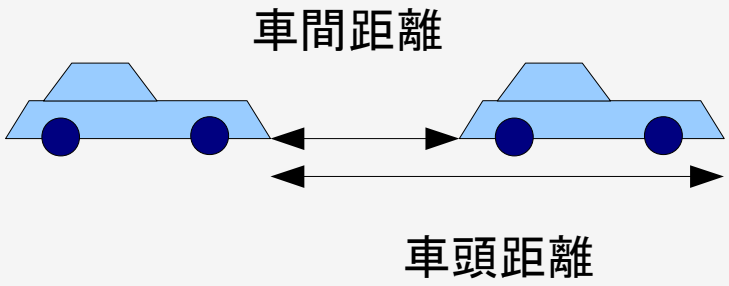
シミュレーションの場合

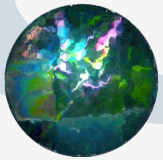




結合写像型最適速度模型

- 交通流のモデル
- 車頭距離に応じた最適速度がある
- 最適速度に調整するように加減速する





モデルの定義

時刻 t での、ある車輌のその先行車輌への車頭距離を $\Delta x(t)$ とする。このとき加速度 $a(t)$ は

$$a(t)=\alpha\left[V_{\text{optimal}}\left(\Delta x(t)\right)-v(t)\right] \tag{1}$$

で与えられる。 $v(t)$ は時刻 t でのその車輌の速度であり、最適速度は次式である。

$$V_{\text{optimal}}\left(\Delta x\right)=\frac{v_{\max}}{2}\left[\tanh\left(2\frac{\Delta x-d}{w}\right)+c\right] \tag{2}$$

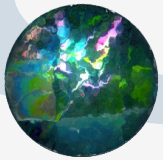
となる。その結果、次の時刻 $t+\Delta t$ での速度は

$$v(t+\Delta t)=v(t)+a(t)\Delta t \tag{3}$$

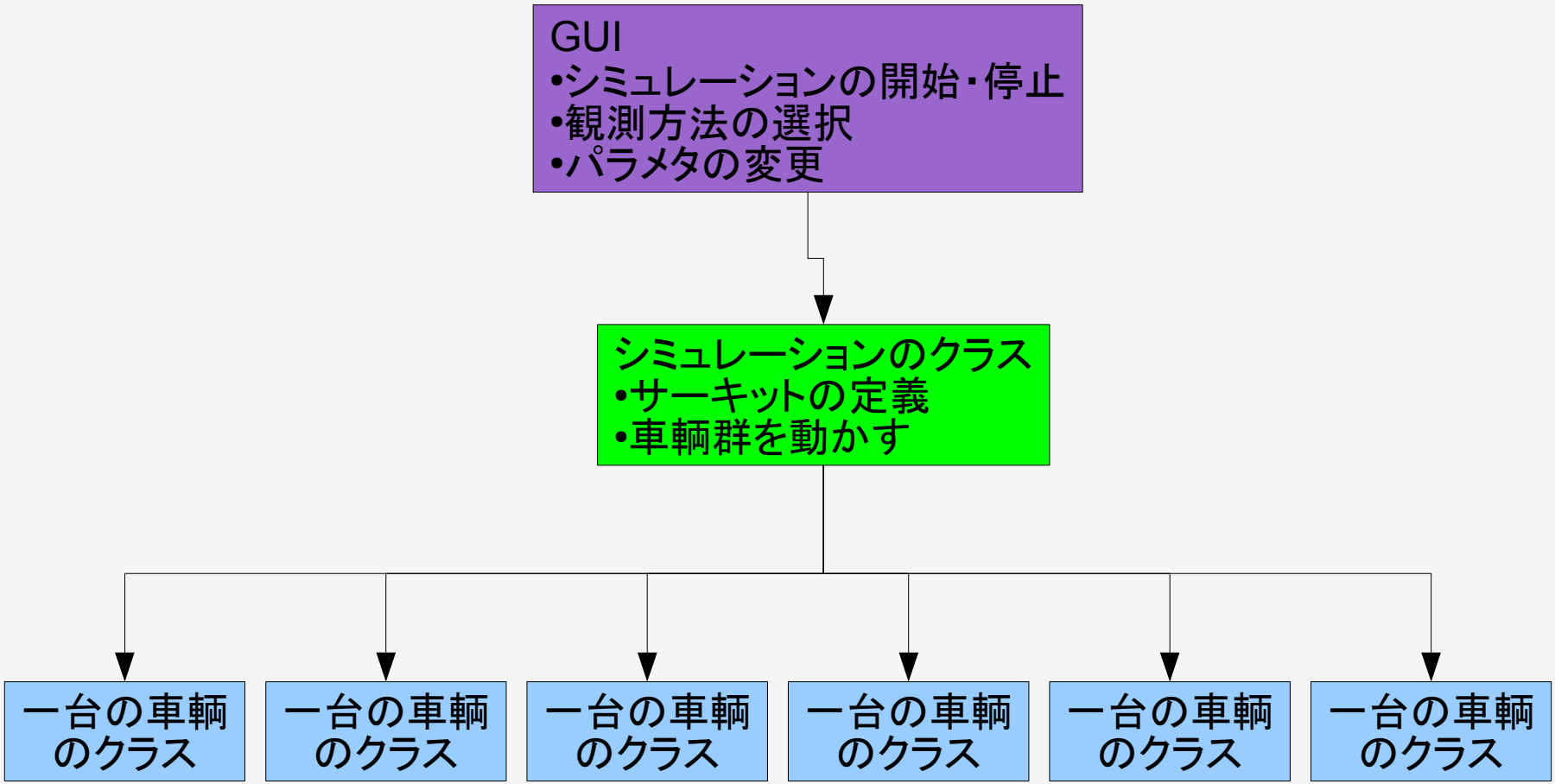
となる。また、位置は

$$x(t+\Delta t)=x(t)+v(t)\Delta t \tag{4}$$

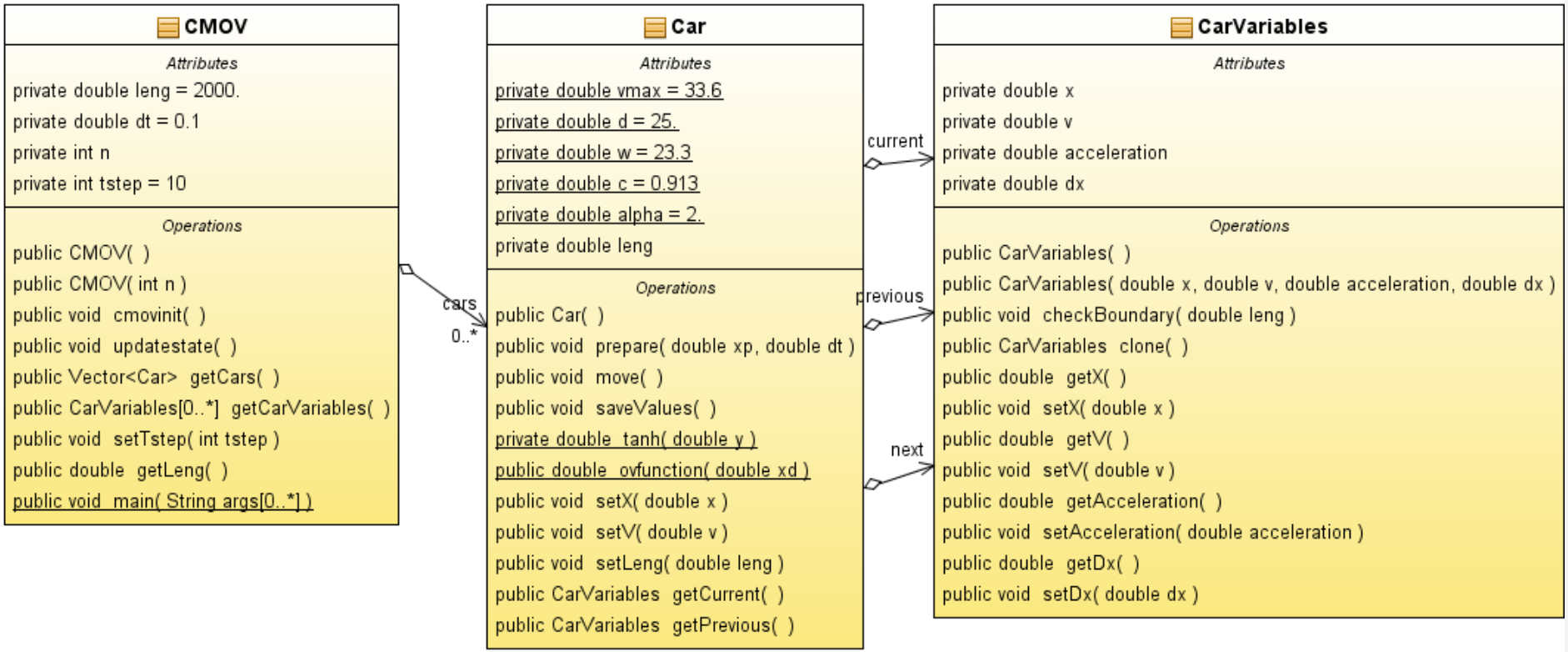
となる。

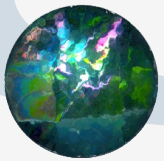


全体構成


GUI
•シミュレーションの開始・停止
•観測方法の選択
•パラメタの変更
シミュレーションのクラス
•サーキットの定義
•車輌群を動かす
一台の車輌のクラス
一台の車輌のクラス
一台の車輌のクラス
一台の車輌のクラス
一台の車輌のクラス
一台の車輌のクラス

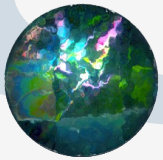
CMOVモデル


CMOV
Attributes
private double leng = 2000.
private double dt = 0.1
private int n
private int tstep = 10
Operations
public CMOV( )
public CMOV(int n)
public void cmovinit( )
public void updatestate( )
public Vector<Car> getCars( )
public CarVariables[0..*] getCarVariables( )
public void setTstep(int tstep)
public double getLeng( )
public void main(String args[0..*])
cars
0..*
Car
Attributes
private double vmax = 33.6
private double d = 25.
private double w = 23.3
private double c = 0.913
private double alpha = 2.
private double leng
Operations
public Car( )
public void prepare(double xp, double dt)
public void move( )
public void saveValues( )
private double tanh(double y)
public double ovfunction(double xd)
public void setX(double x)
public void setV(double v)
public void setLeng(double leng)
public CarVariables getCurrent( )
public CarVariables getPrevious( )
current
previous
next
CarVariables
Attributes
private double x
private double v
private double acceleration
private double dx
Operations
public CarVariables( )
public CarVariables(double x, double v, double acceleration, double dx)
public void checkBoundary(double leng)
public CarVariables clone( )
public double getX( )
public void setX(double x)
public double getV( )
public void setV(double v)
public double getAcceleration( )
public void setAcceleration(double acceleration)
public double getDx( )
public void setDx(double dx)



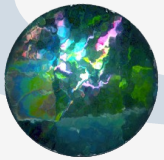
クラスCar

- 一台の車輌のクラス
- 基本となる変数は位置xと速度v
  - 一つのクラスCarVariablesにまとめる
  - 現在、過去、次の時刻の三種類を準備
  - 次の時刻の量を計算：prepare()
  - 現在の量へ更新：move()
  - 現在の値を過去の値として保存：saveValues();
    - 車輌の軌跡を描くために必要

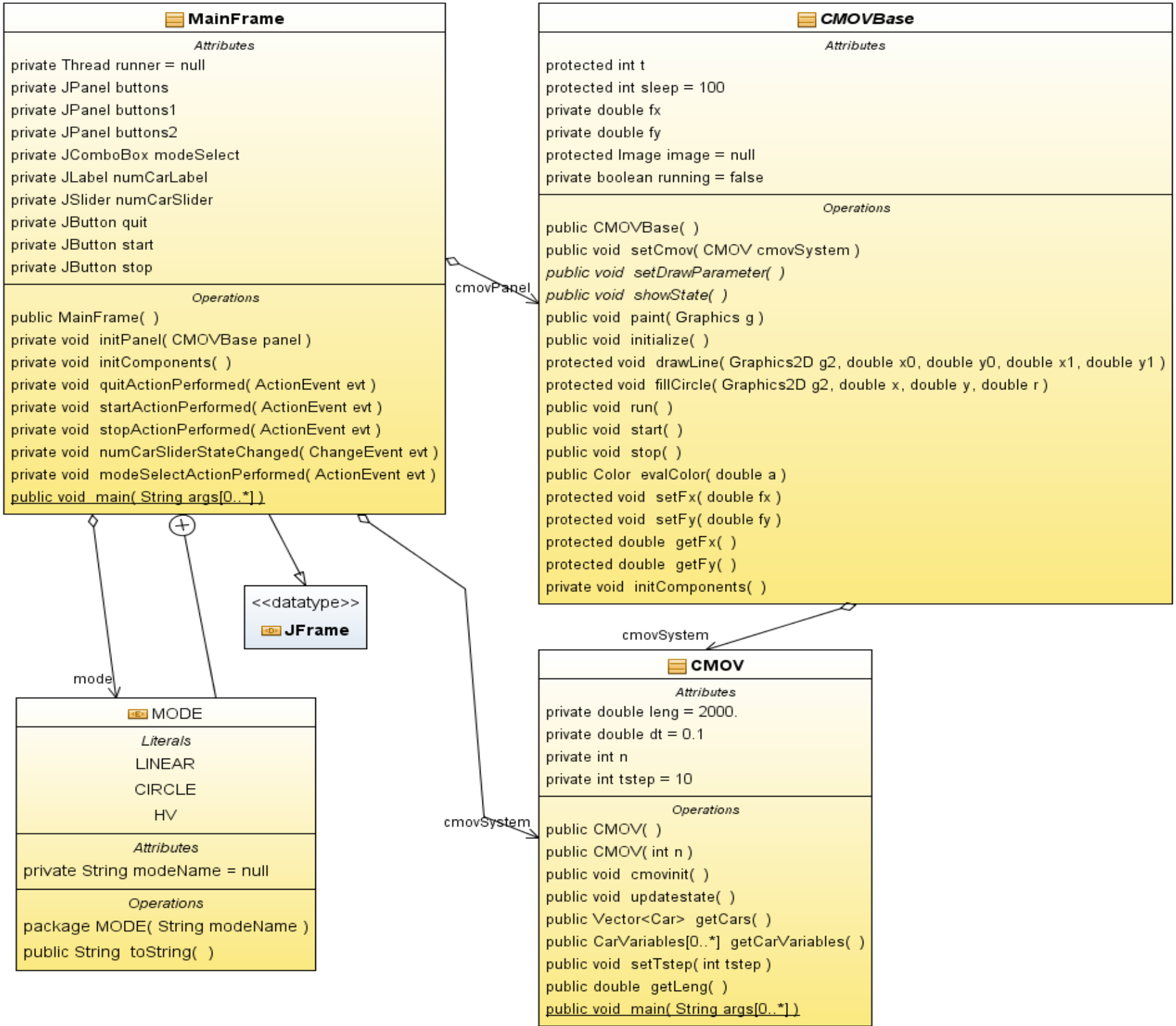


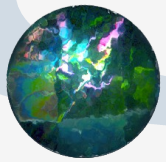
クラスCMOV

- シミュレーションを実行する
- クラスCarのインスタンスのリスト
  - private java.util.Vector<Car> cars;
- 状態更新手順
  - 現在の値を過去の値として保存
  - 次の時刻の量を計算
  - 現在の量へ更新



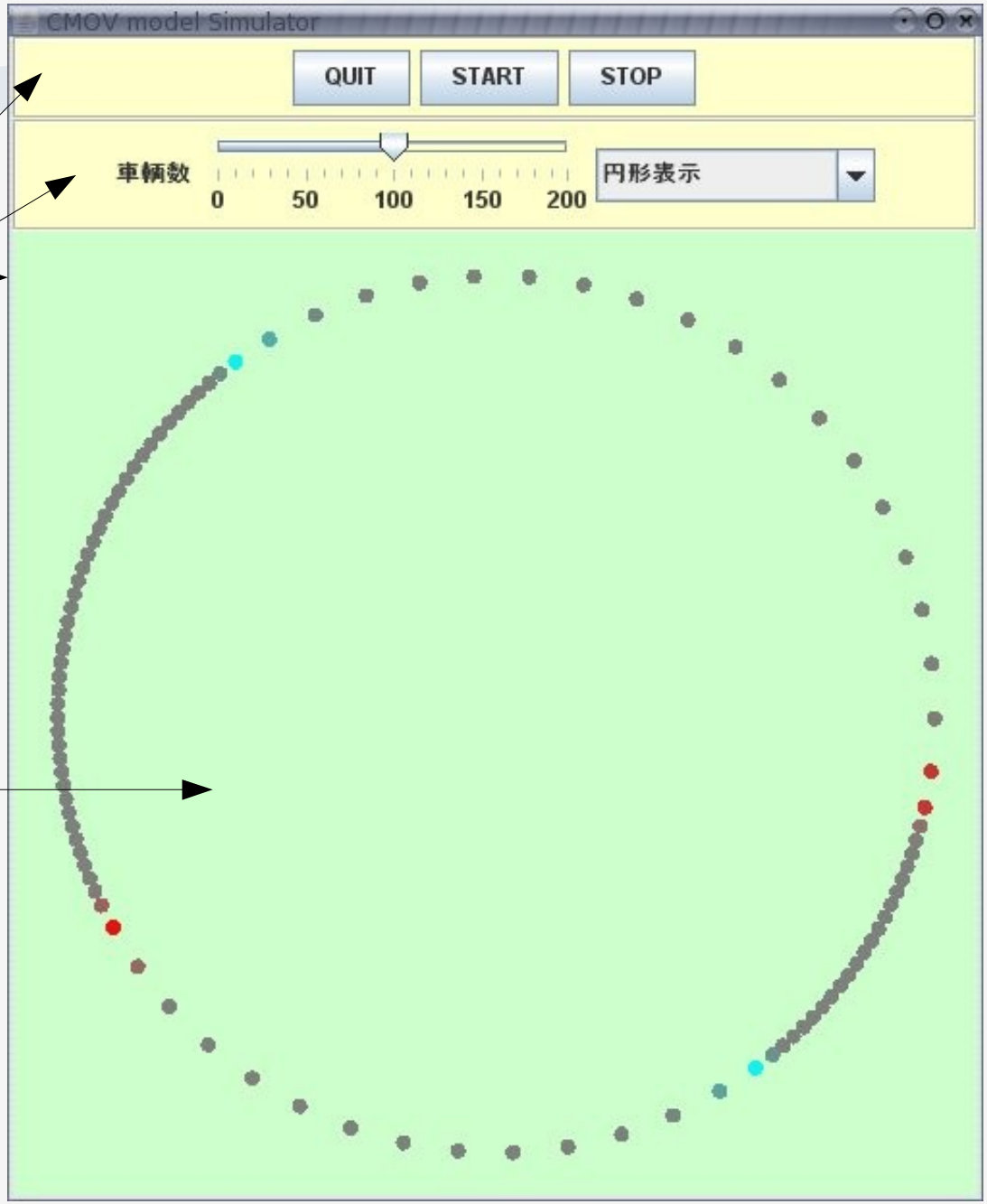
```
public void updatestate() {
    //位置及び速度の保存
    for (int i = 0; i < n; i++) {
        getCars().get(i).saveValues();
    }

    for (int tt = 0; tt < tstep; tt++) {
        //移動準備
        for (int i = 0; i < n - 1; i++) {
            double xp = getCars().get(i + 1).getCurrent().getX();
            getCars().get(i).prepare(xp, dt);
        }
        double xp = getCars().get(0).getCurrent().getX();
        getCars().get(n - 1).prepare(xp, dt);
        //移動実行
        for (Car c : getCars()) {
            c.move();
        }
    }
}
```

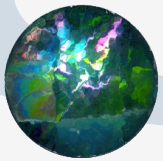

MainFrame
Attributes
private Thread runner = null
private JPanel buttons
private JPanel buttons1
private JPanel buttons2
private JComboBox modeSelect
private JLabel numCarLabel
private JSlider numCarSlider
private JButton quit
private JButton start
private JButton stop
Operations
public MainFrame( )
private void initPanel(CMOVBase panel)
private void initComponents( )
private void quitActionPerformed(ActionEvent evt)
private void startActionPerformed(ActionEvent evt)
private void stopActionPerformed(ActionEvent evt)
private void numCarSliderStateChanged(ChangeEvent evt)
private void modeSelectActionPerformed(ActionEvent evt)
public void main(String args[0..*])
cmovPanel
CMOVBase
Attributes
protected int t
protected int sleep = 100
private double fx
private double fy
protected Image image = null
private boolean running = false
Operations
public CMOVBase( )
public void setCmov(CMOV cmovSystem)
public void setDrawParameter( )
public void showState( )
public void paint(Graphics g)
public void initialize( )
protected void drawLine(Graphics2D g2, double x0, double y0, double x1, double y1)
protected void fillCircle(Graphics2D g2, double x, double y, double r)
public void run( )
public void start( )
public void stop( )
public Color evalColor(double a)
protected void setFx(double fx)
protected void setFy(double fy)
protected double getFx( )
protected double getFy( )
private void initComponents( )
<<datatype>>
JFrame
cmovSystem
CMOV
Attributes
private double leng = 2000.
private double dt = 0.1
private int n
private int tstep = 10
Operations
public CMOV( )
public CMOV(int n)
public void cmovinit( )
public void updatestate( )
public Vector<Car> getCars( )
public CarVariables[0..*] getCarVariables( )
public void setTstep(int tstep)
public double getLeng( )
public void main(String args[0..*])
mode
MODE
Literals
LINEAR
CIRCLE
HV
Attributes
private String modeName = null
Operations
package MODE(String modeName)
public String toString( )
cmovSystem



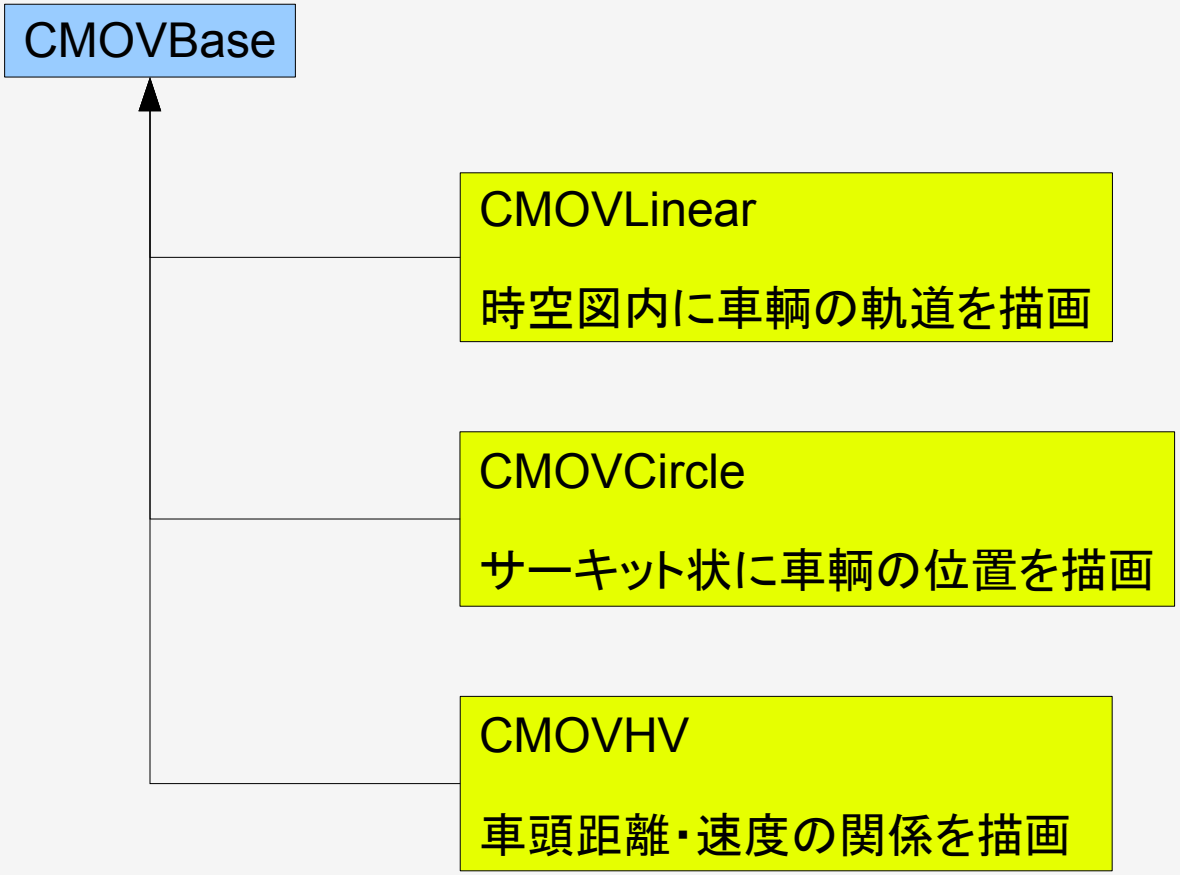
GUIの概要

- 全体のフレーム
  - ボタンパネル
    - ボタン
    - ラベル
    - スライダー
    - コンボボックス
- 描画用パネル

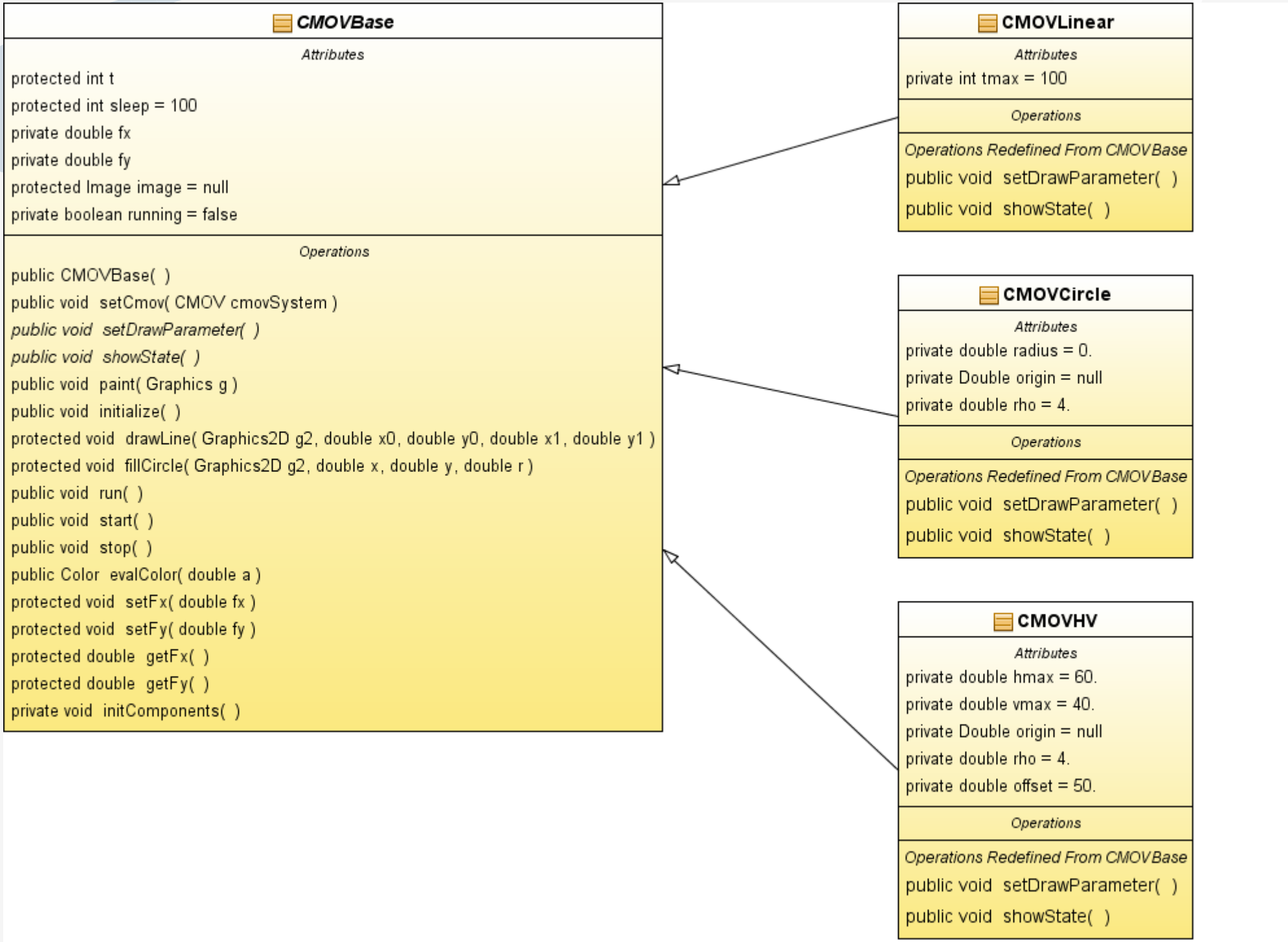




描画パネルの階層



MainFrameクラスから
すべてCMOVBaseの
インスタンスに見せる


CMOVBase
Attributes
protected int t
protected int sleep = 100
private double fx
private double fy
protected Image image = null
private boolean running = false
Operations
public CMOVBase( )
public void setCmov(CMOV cmovSystem)
public void setDrawParameter( )
public void showState( )
public void paint(Graphics g)
public void initialize( )
protected void drawLine(Graphics2D g2, double x0, double y0, double x1, double y1)
protected void fillCircle(Graphics2D g2, double x, double y, double r)
public void run( )
public void start( )
public void stop( )
public Color evalColor(double a)
protected void setFx(double fx)
protected void setFy(double fy)
protected double getFx( )
protected double getFy( )
private void initComponents( )
CMOVLinear
Attributes
private int tmax = 100
Operations
Operations Redefined From CMOVBase
public void setDrawParameter( )
public void showState( )
CMOVCircle
Attributes
private double radius = 0.
private Double origin = null
private double rho = 4.
Operations
Operations Redefined From CMOVBase
public void setDrawParameter( )
public void showState( )
CMOVHV
Attributes
private double hmax = 60.
private double vmax = 40.
private Double origin = null
private double rho = 4.
private double offset = 50.
Operations
Operations Redefined From CMOVBase
public void setDrawParameter( )
public void showState( )