



Written by H.Matsushima 2018/10/13

GraphLibrary 目次

Visual C# 及び Visual Basic などのための グラフィックライブラリ	1
1. GraphLibrary が提供するプロパティ	1
1.1. リビジョン番号を取得.....	1
1.2. ユーザー座標 X 軸最小値	1
1.3. ユーザー座標 X 軸最大値	1
1.4. ユーザー座標 Y 軸最小値	1
1.5. ユーザー座標 Y 軸最大値	1
2. GraphLibrary の実行制御する関数	1
2.1. 描画開始処理	1
2.2. 描画終了処理	1
2.3. ユーザー座標系の定義.....	1
2.4. ユーザー座標系からデバイス座標系へ変換.....	2
2.5. ユーザー座標上の長さからデバイス座標長上の長さへ変換	2
2.6. ユーザー座標系の四角形をデバイス座標系の Rectangle へ変換.....	2
2.7. デバイス座標値からユーザー座標値へ変換.....	2
3. ペンの状態を管理する関数.....	2
3.1. Grsphics オブジェクトの塗りつぶし.....	2
3.2. Graphics オブジェクトの消去（白色で塗りつぶす）	3
3.3. ペンアップ	3
3.4. ペンダウン	3
3.5. 描画するペンの色を設定	3
3.6. 描画するペンの太さを設定	3
3.7. 描画するペンのスタイルを設定	3
3.8. 塗りつぶすブラシのアルファ値を設定	3
3.9. 描画するペンの初期化.....	3
4. ユーザー座標系に図形を描画する関数	3
4.1. ペンを移動する.....	4
4.2. ドットを描く	4
4.3. 直線を描く	4
4.4. 四角形を描く	4
4.5. 四角形を塗りつぶす	4
4.6. 楕円を描く	5
4.7. 楕円を塗りつぶす	5
4.8. 円を描く（半径指定）	5
4.9. 円を塗りつぶす（半径指定）	5
4.10. 円弧を描く	5

4.11.	円弧を描く（半径指定）	6
4.12.	扇型を描く	6
4.13.	扇型を塗りつぶす	6
4.14.	多角形を描く	7
4.15.	多角形を塗りつぶす	7
4.16.	折れ線を描く	7
4.17.	カーディナルスプラインを描画する	7
4.18.	閉じたカーディナルスプラインを描画する	7
4.19.	4つの配列で定義されるベジエスプラインを描画する	7
4.20.	配列から一連のベジエスプラインを描画する	8
5.	デバイス座標系に図形を描画する関数	8
5.1.	直線を描く	8
5.2.	Point 構造体で指定し直線を描く	8
5.3.	座標ペア、幅、および高さで指定された四角形を描画する	8
5.4.	座標ペア、幅、および高さで指定された四角形の内部を塗りつぶす	9
5.5.	座標ペア、幅、および高さで指定された外接する四角形によって定義される楕円を描画する ..	9
5.6.	座標ペア、幅、および高さで指定された外接する四角形によって定義される楕円を塗りつぶす ..	9
5.7.	円を描く（半径指定）	9
5.8.	円を塗りつぶす（半径指定）	9
5.9.	座標ペア、幅、および高さで指定された楕円の一部を表す円弧を描画する	9
5.10.	座標ペア、幅、および高さで指定された楕円と 2本の半径によって定義される扇形を描画する ..	9
5.11.	座標ペア、幅、および高さで指定された楕円と 2本の半径によって定義される扇形の内部を塗りつぶす	10
5.12.	Point 構造体の配列で定義された多角形を描画する	10
5.13.	Point 構造体で指定された点の配列によって定義される多角形の内部を塗りつぶす	10
5.14.	Point 構造体の配列を接続する一連の線分を描画する	10
5.15.	指定した Point 構造体の配列を通過するカーディナルスプラインを描画する	10
5.16.	Point 構造体の配列で定義された、閉じたカーディナルスプラインを描画する	10
5.17.	4つの配列で定義されるベジエスプラインを描画する	10
5.18.	Point 構造体の配列から一連のベジエスプラインを描画する	11
5.19.	Region の内部を塗りつぶす	11
5.20.	GraphicsPath を描画する	11
5.21.	GraphicsPath の内部を塗りつぶす	11
6.	文字列を描画する関数	11
6.1.	省略型で使用するフォントを指定する	11
6.2.	省略型で使用するフォントの色を指定する	12

6.3.	文字列のアルファ値を設定する	12
6.4.	角度を付け文字列を出力する	12
6.5.	文字列を出力する	12
6.6.	始点位置を指定し角度を付け文字列を出力する（省略型）	12
6.7.	始点位置を指定し文字列を出力する（省略型）	13
6.8.	角度を付け文字列を出力する（省略型）	13
6.9.	文字列を出力する（省略型）	13
7.	座標軸を描画する関数	13
7.1.	座標軸を描くための共通設定	13
7.1.1.	目盛ラベルのフォントを指定する	13
7.1.2.	目盛ラベルのフォントサイズを指定する	14
7.1.3.	軸ラベルのフォントを指定する	14
7.1.4.	軸ラベルのフォントサイズを指定する	14
7.1.5.	座標軸を描画する色を指定する	14
7.1.6.	軸の太さを指定する	14
7.1.7.	目盛線の長さを指定する	14
7.1.8.	軸の設定を初期化する	14
7.2.	X 軸を描画する関数	15
7.2.1.	X 軸の描画	15
7.2.2.	目盛ラベル付き X 軸の描画	15
7.2.3.	軸ラベル、目盛ラベル付き X 軸の描画	15
7.2.4.	軸ラベル、目盛ラベル付き X 軸の描画（オフセット設定）	15
7.2.5.	上側 X 軸の描画	15
7.2.6.	目盛ラベル付き上側 X 軸の描画	16
7.2.7.	軸ラベル、目盛ラベル付き上側 X 軸の描画	16
7.2.8.	軸ラベル、目盛ラベル付き上側 X 軸の描画（オフセット設定）	16
7.2.9.	下側 X 軸の描画	16
7.2.10.	目盛ラベル付き下側 X 軸の描画	16
7.2.11.	軸ラベル、目盛ラベル付き下側 X 軸の描画	17
7.2.12.	軸ラベル、目盛ラベル付き下側 X 軸の描画（オフセット設定）	17
7.3.	Y 軸を描画する関数	17
7.3.1.	Y 軸の描画	17
7.3.2.	目盛ラベル付き Y 軸の描画	17
7.3.3.	軸ラベル、目盛ラベル付き Y 軸の描画	17
7.3.4.	軸ラベル、目盛ラベル付き Y 軸の描画（オフセット設定）	18
7.3.5.	右側 Y 軸の描画	18
7.3.6.	目盛ラベル付き右側 Y 軸の描画	18
7.3.7.	軸ラベル、目盛ラベル付き右側 Y 軸の描画	18

7.3.8.	軸ラベル、目盛ラベル付き右側 Y 軸の描画（オフセット設定）	18
7.3.9.	左側 Y 軸の描画.....	19
7.3.10.	目盛ラベル付き左側 Y 軸の描画	19
7.3.11.	軸ラベル、目盛ラベル付き左側 Y 軸の描画.....	19
7.3.12.	軸ラベル、目盛ラベル付き左側 Y 軸の描画（オフセット設定）	19

Visual C# 及び Visual Basic などのための グラフィックライブラリ

1. GraphLibrary が提供するプロパティ

GraphLibrary は以下の5つのプロパティを公開しています

1.1. リビジョン番号を取得

```
public string Revision { get { return revision; } }
```

1.2. ユーザー座標 X 軸最小値

```
public double getXmin { get { return Xmin; } }
```

1.3. ユーザー座標 X 軸最大値

```
public double getXmax { get { return Xmax; } }
```

1.4. ユーザー座標 Y 軸最小値

```
public double getYmin { get { return Ymin; } }
```

1.5. ユーザー座標 Y 軸最大値

```
public double getYmax { get { return Xmax; } }
```

2. GraphLibrary の実行制御する関数

2.1. 描画開始処理

```
public void PlotBEGIN(Graphics obj, Size size)
```

obj Graphics オブジェクト

size Graphics オブジェクトサイズ

2.2. 描画終了処理

```
public void PlotEND()
```

2.3. ユーザー座標系の定義

```
public void AxisInitialize(double xmin, double ymin, double xmax, double ymax)
```

xmin 左下 X 座標

ymin 左下 Y 座標

xmax 右上 X 座標

ymax 右上 Y 座標

Xmin > Xmax に設定した場合は、左右が反転します。

また、Ymin > Ymax に設定した場合は、上下が反転します。

2.4. ユーザー座標系からデバイス座標系へ変換

```
public Point Transrate(double x, double y)
```

x X 座標

y Y 座標

戻り値 出力先のデバイス座標

2.5. ユーザー座標上の長さからデバイス座標長上の長さへ変換

```
public int Transrate(double length)
```

length ユーザー座標の長さ

戻り値 デバイス座標の長さ

2.6. ユーザー座標系の四角形をデバイス座標系の Rectangle へ変換

```
public Rectangle TransrateRect(double x1, double y1, double x2, double y2)
```

x1 対角始点 X 座標

y1 対角始点 Y 座標

x2 対角終点 X 座標

y2 対角終点 Y 座標

戻り値 デバイス座標系の Rectangle

2.7. デバイス座標値からユーザー座標値へ変換

```
public double Device2User(int delta)
```

delta デバイス座標のドット数

戻り値 ユーザー座標値

デバイス座標の delta ドットが、ユーザー座標でどのくらいになるかを取得します

3. ペンの状態を管理する関数

GraphLibrary は、描画する際にペンを使用します。このペンに色や太さ、スタイルを設定し描画する仕様です。

3.1. Grsphics オブジェクトの塗りつぶし

```
public void Clear(Color color)
```


color 塗りつぶす色

3.2. Graphics オブジェクトの消去（白色で塗りつぶす）

```
public void Clear()
```

3.3. ペンアップ

```
public void PenUp()
```

3.4. ペンダウン

```
public void PenDown()
```

3.5. 描画するペンの色を設定

```
public void SetPenColor(Color color)
```

color ペンの色

初期値 黒

3.6. 描画するペンの太さを設定

```
public void SetPenWidth(int width)
```

width ペンの太さ

初期値 1ドット

3.7. 描画するペンのスタイルを設定

```
public void SetPenStyle(DashStyle style)
```

style ペンのスタイル

初期値 実線

3.8. 塗りつぶすブラシのアルファ値を設定

```
public void SetBrushAlpha(int alpha)
```

alpha アルファ値

初期値 255

3.9. 描画するペンの初期化

```
public void ResetPenParm()
```

4. ユーザー座標系に図形を描画する関数

現在設定されているペンで、ユーザー座標系の X 座標と Y 座標で指定し描画します。

また、塗りつぶしは現在選択されているペンの色と SetBrushAlpha で設定されたアルファ値で塗りつぶします。

4.1. ペンを移動する

```
public void Move(double x, double y)
```

x X 座標

y Y 座標

ペンアップの場合は、ペンの位置を指定された座標に移動します。

また、ペンダウンの場合は、現在位置から指定された座標まで直線で描画します。

4.2. ドットを描く

```
public void DrawDot(double x, double y)
```

x X 座標

y Y 座標

ディスプレイに表示する場合は、解像度にもよりますがペンの太さを 2 以上に設定してください。

4.3. 直線を描く

```
public void DrawLine(double x1, double y1, double x2, double y2)
```

x1 始点 X 座標

y1 始点 Y 座標

x2 終点 X 座標

y2 終点 Y 座標

始点の座標から終点の座標まで直線を描きます。

4.4. 四角形を描く

```
public void DrawRectangle(double x1, double y1, double x2, double y2)
```

x1 対角始点 X 座標

y1 対角始点 Y 座標

x2 対角終点 X 座標

y2 対角終点 Y 座標

ここでは、四角形の対角の対を、対角始点及び対角終点と記載しています。

四角形の、左下と右上、または、左上と右下を指定します。

また、右上と左下、または、右下と左上の順番で指定しても問題ありません。

4.5. 四角形を塗りつぶす

```
public void FillRectangle(double x1, double y1, double x2, double y2)
```

x1 対角始点 X 座標

y1 対角始点 Y 座標

x2 対角終点 X 座標

y2 対角終点 Y 座標

対角で指定されつ四角形を塗りつぶします。

4.6. 楕円を描く

```
public void DrawEllipse(double x1, double y1, double x2, double y2)
```

x1 対角始点 X 座標

y1 対角始点 Y 座標

x2 対角終点 X 座標

y2 対角終点 Y 座標

対角で指定された四角形に内接する楕円を描きます。

4.7. 楕円を塗りつぶす

```
public void FillEllipse(double x1, double y1, double x2, double y2)
```

x1 対角始点 X 座標

y1 対角始点 Y 座標

x2 対角終点 X 座標

y2 対角終点 Y 座標

対角で指定された四角形に内接する楕円を塗りつぶします。

4.8. 円を描く（半径指定）

```
public void DrawCircle(double x, double y, double r)
```

x 中心 X 座標

y 中心 Y 座標

r 半径

円の中心及び半径を指定して円を描きます。

4.9. 円を塗りつぶす（半径指定）

```
public void FillCircle(double x, double y, double r)
```

x X 座標

y Y 座標

r 半径

円の中心及び半径を指定して円を塗りつぶします。

4.10. 円弧を描く

```
public void DrawArc(double x1, double y1, double x2, double y2, float start,
float sweep)
```

x1 対角始点 X 座標

y1 対角始点 Y 座標

x2 対角終点 X 座標
 y2 対角終点 Y 座標
 start 開始角度
 sweep スweep角度

対角で指定された四角形に内接した楕円にそって円弧を描きます。

描画される範囲は、開始角度からスweep角度までです。

.NET Framework では円弧にかかる指定を、X 軸の正方向から時計回りの角度で指定します。

私は、この仕様を知った時、目がテンになってしまいました。

右回りの角度系は時計の針とレーダー画面また船舶・航空機などで使われてはいます。

しかし、数学で使用する角度系は X 軸の正方向から反時計回りで使用するのが一般的です。

このため、GraphLibrary では.NET Framework とはあえて異なる、X 軸の正方向から反時計回りで指定します。

4.11. 円弧を描く（半径指定）

```
public void DrawArc(double x, double y, double r, float start, float sweep)
```

x 中心 X 座標
 y 中心 Y 座標
 r 半径
 start 開始角度
 sweep スweep角度

円の中心から半径の円にそって、開始角度からスweep角度まで円弧を描きます。

角度は X 軸の正方向から反時計回りで指定します。

4.12. 扇型を描く

```
public void DrawPie(double x1, double y1, double x2, double y2, float start,
    float sweep)
```

x1 対角始点 X 座標
 y1 対角始点 Y 座標
 x2 対角終点 X 座標
 y2 対角終点 Y 座標
 start 開始角度
 sweep スweep角度

対角で指定された四角形に内接した楕円にそって、扇型を描きます。

描画される範囲は、開始角度からスweep角度までです。

角度は X 軸の正方向から反時計回りで指定します。

4.13. 扇型を塗りつぶす

```
public void FillPie(double x1, double y1, double x2, double y2, float start,
    float sweep)
```

x1 対角始点 X 座標
 y1 対角始点 Y 座標
 x2 対角終点 X 座標
 y2 対角終点 Y 座標
 start 開始角度
 sweep スイープ角度

対角で指定された四角形に内接した楕円にそって、扇型を塗りつぶします。

描画される範囲は、開始角度からスイープ角度までです。

角度は X 軸の正方向から反時計回りで指定します。

4.14. 多角形を描く

```
public void DrawPolygon(double[,] array)
    array    X 座標と Y 座標を収めた二次元配列
```

4.15. 多角形を塗りつぶす

```
public void FillPolygon(double[,] array, FillMode mode)
    array    X 座標と Y 座標を収めた二次元配列
    mode    塗りつぶしモード
```

4.16. 折れ線を描く

```
public void DrawLines(double[,] array)
    array    X 座標と Y 座標を収めた二次元配列
```

4.17. カーディナルスプラインを描画する

```
public void DrawCurve(double[,] array, float tension)
    array    X 座標と Y 座標を収めた二次元配列
    tension 曲線のテンションを指定する 0.0F 以上の値
```

4.18. 閉じたカーディナルスプラインを描画する

```
public void DrawClosedCurve(double[,] array, float tension)
    array    X 座標と Y 座標を収めた二次元配列
    tension 曲線のテンションを指定する 0.0F 以上の値
```

4.19. 4 つの配列で定義されるベジエスプラインを描画する

```
public void DrawBezier(double[,] array)
    array                                      曲線を決定する複数の点を表す 4 つの X, Y 座標の二次元配列
    array[0,0],array[0,1]                    曲線の始点を表す X, Y 座標の配列
    array[1,0],array[1,1]                    曲線の最初の制御点を表す X, Y 座標の配列
```

array[2,0],array[2,1] 曲線の2番目の制御点を表す X,Y 座標の配列
 array[3,0],array[3,1] 曲線の終点を表す X,Y 座標の配列

4.20. 配列から一連のベジエスプラインを描画する

```
public void DrawBeziers(double[,] array)
```

array 曲線を決定する複数の点を表す X,Y 座標の二次元配列。
 配列内の点の数は、3 の倍数に 1 を加算した数である必要がある。

5. デバイス座標系に図形を描画する関数

現在設定されているペンで、デバイス座標系の X 座標と Y 座標で指定し描画します。

また、塗りつぶしは現在選択されているペンの色と SetBrushAlpha で設定されたアルファ値で塗りつぶします。

DirectXXXX 関数は、デバイス座標に直接描画する関数で、.NET Framework の提供する XXXX 関数に準拠して作成しています。

DirectXXXX 関数を使用し、ユーザー座標系で指定して描画する例

```
gl.DirectDrawLine(gl.Transrate(0, 0), gl.Transrate(1, 1));  
gl.DirectDrawRectangle(gl.TransrateRect(0, 0, 1, 1));  
gl.DirectDrawCircle(gl.Transrate(0, 0), gl.Transrate(1));
```

5.1. 直線を描く

```
public void DirectDrawLine(int x1, int y1, int x2, int y2)
```

x1 始点 X デバイス座標
 y1 始点 Y デバイス座標
 x2 終点 X デバイス座標
 y2 終点 Y デバイス座標

5.2. Point 構造体で指定し直線を描く

```
public void DirectDrawLine(Point Sp, Point Ep)
```

Sp 始点デバイス座標
 Ep 終点デバイス座標

5.3. 座標ペア、幅、および高さで指定された四角形を描画する

```
public void DirectDrawRectangle(Rectangle rect)
```

rect デバイス座標の Rectangle 構造体

5.4. 座標ペア、幅、および高さで指定された四角形の内部を塗りつぶす

```
public void DirectFillRectangle(Rectangle rect)
```

rect デバイス座標の Rectangle 構造体

5.5. 座標ペア、幅、および高さで指定された外接する四角形によって定義される楕円を描画する

```
public void DirectDrawEllipse(Rectangle rect)
```

rect デバイス座標の Rectangle 構造体

5.6. 座標ペア、幅、および高さで指定された外接する四角形によって定義される楕円を塗りつぶす

```
public void DirectFillEllipse(Rectangle rect)
```

rect デバイス座標の Rectangle 構造体

5.7. 円を描く（半径指定）

```
public void DirectDrawCircle(Point center, int r)
```

center 中心座標

r 半径

5.8. 円を塗りつぶす（半径指定）

```
public void DirectFillCircle(Point center, int r)
```

center 中心座標

r 半径

5.9. 座標ペア、幅、および高さで指定された楕円の一部を表す円弧を描画する

```
public void DirectDrawArc(Rectangle rect, float start, float sweep)
```

rect デバイス座標の Rectangle 構造体

start 開始角度

sweep スイープ角度

描画される範囲は、開始角度からスイープ角度までです。

角度は X 軸の正方向から反時計回りで指定します。

5.10. 座標ペア、幅、および高さで指定された楕円と 2 本の半径によって定義される扇形を描画する

```
public void DirectDrawPie(Rectangle rect, float start, float sweep)
```

rect デバイス座標の Rectangle 構造体

start 開始角度

sweep スイープ角度

描画される範囲は、開始角度からスイープ角度までです。

角度は X 軸の正方向から反時計回りで指定します。

5.11. 座標ペア、幅、および高さで指定された楕円と 2 本の半径によって定義される扇形の内部を塗りつぶす

```
public void DirectFillPie(Rectangle rect, float start, float sweep)
```

rect デバイス座標の Rectangle 構造体

start 開始角度

sweep スweep 角度

描画される範囲は、開始角度からスweep 角度までです。

角度は X 軸の正方向から反時計回りで指定します。

5.12. Point 構造体の配列で定義された多角形を描画する

```
public void DirectDrawPolygon(Point[] points)
```

points 多角形の頂点を表す Point 構造体の配列

5.13. Point 構造体で指定された点の配列によって定義される多角形の内部を塗りつぶす

```
public void DirectFillPolygon(Point[] points, FillMode mode)
```

points 多角形の頂点を表す Point 構造体の配列

mode 塗りつぶしモード

5.14. Point 構造体の配列を接続する一連の線分を描画する

```
public void DirectDrawLines(Point[] points)
```

points 多角形の頂点を表す Point 構造体の配列

5.15. 指定した Point 構造体の配列を通過するカーディナルスプラインを描画する

```
public void DirectDrawCurve(Point[] points, float tension)
```

points スプラインを定義する Point 構造体の配列

tension 曲線のテンションを指定する 0.0F 以上の値

5.16. Point 構造体の配列で定義された、閉じたカーディナルスプラインを描画する

```
public void DirectDrawClosedCurve(Point[] points, float tension)
```

points スプラインを定義する Point 構造体の配列

tension 曲線のテンションを指定する 0.0F 以上の値

5.17. 4 つの配列で定義されるベジエスプラインを描画する

```
public void DirectDrawBezier(Point point1, Point point2, Point point3, Point point4)
```

point1 曲線の開始点を表す Point 構造体

point2 曲線の最初の制御点を表す Point 構造体

point3 曲線の 2 番目の制御点を表す Point 構造体

point4 曲線の終了点を表す Point 構造体

5.18. Point 構造体の配列から一連のベジエスプラインを描画する

```
public void DirectDrawBeziers(Point[] points)
```

points 曲線を決定する複数の点を表す Point 構造体の配列。

配列内の点の数は、3 の倍数に 1 を加算した数である必要がある。

5.19. Region の内部を塗りつぶす

```
public void DirectFillRegion(Region region)
```

region 四角形とパスで構成されるグラフィックス形状の内部を示す

5.20. GraphicsPath を描画する

```
public void DirectDrawPath(GraphicsPath path)
```

path 接続された一連の直線と曲線を表す

5.21. GraphicsPath の内部を塗りつぶす

```
public void DirectFillPath(GraphicsPath path)
```

path 接続された一連の直線と曲線を表す

6. 文字列を描画する関数

文字列を描画する各関数に出てくる、始点位置指定とは、x、y 座標に対し以下のような位置関係になる。

- 0: 左上寄せ
- 1: 中央上寄せ
- 2: 右上寄せ
- 3: 左中央寄せ
- 4: 中央中央寄せ
- 5: 右中央寄せ
- 6: 左下寄せ
- 7: 中央下寄せ
- 8: 右下寄せ

6.1. 省略型で使用するフォントを指定する

```
public void SetStringFont(string font)
```

font フォント名

初期値 "MS Pゴシック"

6.2. 省略型で使用するフォントの色を指定する

```
public void SetStringColor(Color color)
    color    フォントの色
    初期値   Color.Black
```

6.3. 文字列のアルファ値を設定する

```
public void SetStringAlpha(int alpha)
    alpha    アルファ値
    初期値   255
```

6.4. 角度を付け文字列を出力する

```
public void DrawString(double x, double y, int pos, string font, int size,
    Color color, string msg, float angle)
    x        X 座標
    y        Y 座標
    pos      始点位置指定
    font     フォント名
    size     フォントサイズ
    color    フォントの色
    msg      出力文字列
    angle    傾斜角度
```

6.5. 文字列を出力する

```
public void DrawString(double x, double y, int pos, string font,
int size, Color color, string msg)
    x        X 座標
    y        Y 座標
    pos      始点位置指定
    font     フォント名
    size     フォントサイズ
    color    フォントの色
    msg      出力文字列
```

6.6. 始点位置を指定し角度を付け文字列を出力する（省略型）

```
public void DrawString(double x, double y, int pos, int size, string msg, float angle)
    x        X 座標
    y        Y 座標
```

pos	始点位置指定
size	フォントサイズ
msg	出力文字列
angle	傾斜角度

6.7. 始点位置を指定し文字列を出力する（省略型）

```
public void DrawString(double x, double y, int pos, int size, string msg)
```

x	X 座標
y	Y 座標
pos	始点位置指定
size	フォントサイズ
msg	出力文字列

6.8. 角度を付け文字列を出力する（省略型）

```
public void DrawString(double x, double y, int size, string msg, float angle)
```

x	X 座標
y	Y 座標
size	フォントサイズ
msg	出力文字列
angle	傾斜角度

6.9. 文字列を出力する（省略型）

```
public void DrawString(double x, double y, int size, string msg)
```

x	X 座標
y	Y 座標
size	フォントサイズ
msg	出力文字列

7. 座標軸を描画する関数

座標軸の描画は、以下の6種類に分類される。

X 軸を描く AxisX, AxisXu(アップ), AxisXd(ダウン)

Y 軸を描く AxisY, AxisYr(ライト), AxisYl(レフト)

7.1. 座標軸を描くための共通設定

7.1.1. 目盛ラベルのフォントを指定する

```
public void SetScaleFont(string font)
```

font フォント名
初期値 "MS UI Gothic"

7.1.2. 目盛ラベルのフォントサイズを指定する

```
public void SetScaleFontSize(int point)
```

point フォントサイズ
初期値 8 ポイント

7.1.3. 軸ラベルのフォントを指定する

```
public void SetLabelFont(string font)
```

font フォント名
初期値 "MS UI Gothic"

7.1.4. 軸ラベルのフォントサイズを指定する

```
public void SetLabelFontSize(int point)
```

point フォントサイズ
初期値 10 ポイント

7.1.5. 座標軸を描画する色を指定する

```
public void SetAxisColor(Color color)
```

color 描画する色
初期値 Color.Black

7.1.6. 軸の太さを指定する

```
public void SetAxisWidth(int width)
```

width 軸の太さ
初期値 1 ドット

7.1.7. 目盛線の長さを指定する

```
public void SetAxisScaleMark(int length)
```

length 目盛線の長さ
初期値 3 ドット

7.1.8. 軸の設定を初期化する

```
public void ResetAxisParm()
```

7.2. X 軸を描画する関数

7.2.1. X 軸の描画

```
public void AxisX(double min, double max, double step)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
```

7.2.2. 目盛ラベル付き X 軸の描画

```
public void AxisX(double min, double max, double step, double lstep)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
```

7.2.3. 軸ラベル、目盛ラベル付き X 軸の描画

```
public void AxisX(double min, double max, double step, double lstep, string label)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
    label    軸ラベル
```

7.2.4. 軸ラベル、目盛ラベル付き X 軸の描画（オフセット設定）

```
public void AxisX(double min, double max, double step, double lstep, string label,
    int off)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
    label    軸ラベル
    off     軸からのオフセット
```

7.2.5. 上側 X 軸の描画

```
public void AxisXu(double min, double max, double step)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
```

7.2.6. 目盛ラベル付き上側 X 軸の描画

```
public void AxisXu(double min, double max, double step, double lstep)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
```

7.2.7. 軸ラベル、目盛ラベル付き上側 X 軸の描画

```
public void AxisXu(double min, double max, double step, double lstep, string label)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
    label    軸ラベル
```

7.2.8. 軸ラベル、目盛ラベル付き上側 X 軸の描画 (オフセット設定)

```
public void AxisXu(double min, double max, double step, double lstep, string label,
    int off)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
    label    軸ラベル
    off      軸からのオフセット
```

7.2.9. 下側 X 軸の描画

```
public void AxisXd(double min, double max, double step)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
```

7.2.10. 目盛ラベル付き下側 X 軸の描画

```
public void AxisXd(double min, double max, double step, double lstep)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
```

7.2.11. 軸ラベル、目盛ラベル付き下側 X 軸の描画

```
public void AxisXd(double min, double max, double step, double lstep, string label)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
    label    軸ラベル
```

7.2.12. 軸ラベル、目盛ラベル付き下側 X 軸の描画（オフセット設定）

```
public void AxisXd(double min, double max, double step, double lstep, string label,
    int off)
    min      X 軸最小値
    max      X 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
    label    軸ラベル
    off      軸からのオフセット
```

7.3. Y 軸を描画する関数

7.3.1. Y 軸の描画

```
public void AxisY(double min, double max, double step)
    min      Y 軸最小値
    max      Y 軸最大値
    step     目盛間隔
```

7.3.2. 目盛ラベル付き Y 軸の描画

```
public void AxisY(double min, double max, double step, double lstep)
    min      Y 軸最小値
    max      Y 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
```

7.3.3. 軸ラベル、目盛ラベル付き Y 軸の描画

```
public void AxisY(double min, double max, double step, double lstep, string label)
    min      Y 軸最小値
    max      Y 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
```

label 軸ラベル

7.3.4. 軸ラベル、目盛ラベル付き Y 軸の描画（オフセット設定）

```
public void AxisY(double min, double max, double step, double lstep, string label,
    int off)
    min      Y 軸最小値
    max      Y 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
    label    軸ラベル
    off      軸からのオフセット
```

7.3.5. 右側 Y 軸の描画

```
public void AxisYr(double min, double max, double step)
    min      Y 軸最小値
    max      Y 軸最大値
    step     目盛間隔
```

7.3.6. 目盛ラベル付き右側 Y 軸の描画

```
public void AxisYr(double min, double max, double step, double lstep)
    min      Y 軸最小値
    max      Y 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
```

7.3.7. 軸ラベル、目盛ラベル付き右側 Y 軸の描画

```
public void AxisYr(double min, double max, double step, double lstep, string label)
    min      Y 軸最小値
    max      Y 軸最大値
    step     目盛間隔
    lstep    目盛ラベル間隔
    label    軸ラベル
```

7.3.8. 軸ラベル、目盛ラベル付き右側 Y 軸の描画（オフセット設定）

```
public void AxisYr(double min, double max, double step, double lstep, string label,
    int off)
    min      Y 軸最小値
    max      Y 軸最大値
```


step 目盛間隔
 lstep 目盛ラベル間隔
 label 軸ラベル
 off 軸からのオフセット

7.3.9. 左側 Y 軸の描画

```
public void AxisYl(double min, double max, double step)
    min     Y 軸最小値
    max     Y 軸最大値
    step    目盛間隔
```

7.3.10. 目盛ラベル付き左側 Y 軸の描画

```
public void AxisYl(double min, double max, double step, double lstep)
    min     Y 軸最小値
    max     Y 軸最大値
    step    目盛間隔
    lstep   目盛ラベル間隔
```

7.3.11. 軸ラベル、目盛ラベル付き左側 Y 軸の描画

```
public void AxisYl(double min, double max, double step, double lstep, string label)
    min     Y 軸最小値
    max     Y 軸最大値
    step    目盛間隔
    lstep   目盛ラベル間隔
    label   軸ラベル
```

7.3.12. 軸ラベル、目盛ラベル付き左側 Y 軸の描画（オフセット設定）

```
public void AxisYl(double min, double max, double step, double lstep, string label,
    int off)
    min     Y 軸最小値
    max     Y 軸最大値
    step    目盛間隔
    lstep   目盛ラベル間隔
    label   軸ラベル
    off      軸からのオフセット
```