10 インタフェース

10.1　インタフェースのソースリスト

Main070View.java に次のソースを追加してください。(網掛け部分)

ファイル名：　src/jp/edu/mie/ Main070View.java


```
package jp.edu.mie;
・・・
public class Main070View extends View
{
    ・・・
    protected void onDraw(Canvas canvas)
    {
        ・・・
        shapes[14] = new Point();
        shapes[15] = new Star();
        shapes[16] = new StarFill();


        HasGetAreaMethod[] closeShapes = new HasGetAreaMethod[17];//①
        closeShapes[0] = new RectangleFill();
        closeShapes[1] = new CircleFill();
        closeShapes[2] = new TriangleFill();

        HasGetLengthMethod[] closeShapesLength = new HasGetLengthMethod[17];
        closeShapesLength[0] = new RectangleFill();
        closeShapesLength[1] = new CircleFill();

        for(int i = 0; i < shapes.length; i++)
        {
            if(shapes[i] == null){ break; }
            shapes[i].draw(canvas, paint);
        }
        paint.reset();
        paint.setTextSize(24);
        for(int i = 0; i < closeShapes.length; i++)
        {
            if(closeShapes[i] == null){ break; }
            closeShapes[i].getArea(canvas, paint);
        }

        for(int i = 0; i < closeShapesLength.length; i++)
        {
            if(closeShapesLength[i] == null){ break; }
            closeShapesLength[i].getLength(canvas, paint);
        }
    }
}
interface HasGetAreaMethod
{
```

```
    int pixel = 41;  //1cm 当たりのピクセル
    public abstract void getArea(Canvas canvas, Paint paint);  //面積
}
interface HasGetLengthMethod
{
    int pixel = 41;  //1cm 当たりのピクセル
    public abstract void getLength(Canvas canvas, Paint paint);  //長さ
}
abstract class Shape
{
    abstract void draw(Canvas canvas, Paint paint);
}
・・・
//四角形
class Rectangle extends Shape
{
    ・・・
}
//②四角形の塗りつぶし
class RectangleFill extends Shape
                            implements HasGetAreaMethod, HasGetLengthMethod
{
    int x = 110;
    int y = 100;
    int incx = 80;
    int incy = 80;
    void draw(Canvas canvas, Paint paint)
    {
        ・・・
    }
    public void getArea(Canvas canvas, Paint paint)  //③
    {
        double area = (double)incx / HasGetAreaMethod.pixel *
                                    (double)incy / HasGetAreaMethod.pixel;
        canvas.drawText("四角形の面積は " + area + " ｃ㎡です。", 240, y +
                                                        incy / 2, paint);
    }
    public void getLength(Canvas canvas, Paint paint)
    {
        double length = ((double)incx + incy) / HasGetLengthMethod.pixel;
        canvas.drawText("四角形の辺の長さは " + length + " cm です。",
                                                    240, y + incy, paint);
    }
}
・・・
class RoundedRectangleFill extends Shape  //角丸四角形の塗りつぶし
{
    ・・・
}
//円
```

```
class Circle extends Shape
{
    ・・・
}
//円の塗りつぶし
class CircleFill extends Shape
                          implements HasGetAreaMethod, HasGetLengthMethod
{
    ・・・
    void draw(Canvas canvas, Paint paint)
    {
        ・・・
    }
    public void getArea(Canvas canvas, Paint paint)
    {
        double area = Math.PI * Math.pow(((double)r /
                                              HasGetAreaMethod.pixel), 2);
        canvas.drawText("円の面積は " + area + " c ㎡です。", 240, y, paint);
    }
    public void getLength(Canvas canvas, Paint paint)
    {
        double length = ((double)2 * r * Math.PI) / HasGetLengthMethod.pixel;
        canvas.drawText("円周は " + length + " cm です。", 240, y + 40, paint);
    }
}
class Triangle extends Shape  //三角形
{
    ・・・
}
//三角形の塗りつぶし
class TriangleFill extends Shape  implements HasGetAreaMethod
{
    void draw(Canvas canvas, Paint paint)
    {
        ・・・
    }
    public void getArea(Canvas canvas, Paint paint)
    {
        int base = 80;
        double area = Math.pow(((double)base / pixel), 2) *
                                                  (double)Math.tan(60);
        canvas.drawText("三角形の面積は " + area +
                                          " c ㎡です。", 240, 450, paint);
    }
}
```

10.2　インタフェースのソースリストの解説

①HasGetAreaMethod[] closeShapes = new HasGetAreaMethod[17];

インタフェースのメリットは、ポリモーフィズムの仕組みを使えることです。

②class RectangleFill extends Shape implements HasGetAreaMethod, HasGetLengthMethod

クラスはインタフェースを２つ以上実装することができます。

③public void getArea(Canvas canvas, Paint paint)

インタフェースを実装するクラスは､インタフェースで宣言されたメソッドをすべて実装しなくてはなりません。

10.3　クラスと比較したときのインタフェースの特徴

（1）インスタンスを作れない。

（2）インタフェースで宣言したメソッドは、暗黙的に public abstract 修飾子が付いているものとして扱われる。

（3）インタフェースで宣言したフィールドは、暗黙的に public static final 修飾子が付いているものとして扱われる。(つまり定数として扱われる)

（4）ポリモーフィズムを活用できる。

（5）２つ以上のインタフェースを実装するクラスを宣言できる。

（6）大規模なプログラムを作成するときに役立つ。

10.4 抽象クラスとの比較

（1）インタフェースのフィールドはすべて定数で、メソッドはすべて抽象メソッドとなる。

(2)抽象クラスのように値を変更できるフィールドや処理が定義されているメソッドのようなメンバをもたせることはできない。