

課題 1

Rectangleクラスの拡張

教科書p.44 ソースコード3.7のRectangle クラスを以下の手順で拡張してみましょう.

- (1) 周りの長さを求める `calcAround` メソッドの追加
- (2) 長方形の幅と高さを2倍にする `doublify` メソッドの追加
- (3) 長方形の幅と高さを半分ににする `half` メソッドの追加
(小数点以下は切り捨て)
- (4) 長方形を90度回転させる(幅と高さの値を入替える) `rotate90` メソッドの追加

(1)～(4)を実装するたびに、ソースコード3.3 `Main1`の`main`メソッドを書き換え、実行してみましょう.

- (5) `showState`メソッドの本体を「`system.out.printf`」を使ったものに書き換えてみましょう

課題1のクラス図

::Rectangle	
-xPosition: int -yPosition: int -width: int -height: int	
+Rectangle(xPosition: int, yPosition: int, width: int, height: int) +calcArea(): int +move(xMove: int, yMove: int): void +showState(): void +calcAround(): int +doublify(): void +half(): void +rotate90(): void	

ヒントなど

- `calcAround` は `calcArea` を参考に作れば簡単
- `doublify` も `move` を参考に作ればさほど難しくありません.
メソッド名は「`double`」でもよさそうに思えますが, **Java言語であらかじめ予約されたキーワード(教科書p.50)**なので 使えません
- `half` はあまりややこしく考えず, 2で割れば結構です.
- `rotate90`は少しクセモノです. 単に

```
public void rotate90() {  
    width = height;  
    height = width;  
}
```

としたのではうまくいきません. フィールド `width`, `height` とは別にローカル変数を使う必要があります.
- 書き換える `showState`の本体は「`System.out.printf(...);`」一行のみにしてみましょう.

(1)を確認するMain1のmainメソッド

```
public static void main(String[] args) {  
    Rectangle rectangle1= new Rectangle(2,1,5,6);  
    int area = rectangle1.calcArea();  
    int around = rectangle1.calcAround();  
    rectangle1.printInfo();  
    System.out.printf("の長方形の面積は%d、周りの長さは%dです . ¥n",  
                        area, around);  
}
```

x座標: 2, y座標: 1, 幅: 5, 高さ: 6
の長方形の面積は 30、周りの長さは 22 です.

(1)～(2)を確認するMain1のmainメソッド

```
public static void main(String[] args) {  
    Rectangle rectangle1= new Rectangle(2,1,5,6);  
    int area = rectangle1.calcArea();  
    int around = rectangle1.calcAround();  
    rectangle1.printInfo();  
    System.out.printf("の長方形の面積は%d、周りの長さは%dです . ¥n",  
                        area, around);  
  
    rectangle1.doublify();  
    rectangle1.showState();  
}
```

x座標: 2, y座標: 1, 幅: 5, 高さ: 6

の長方形の面積は 30、周りの長さは 22 です.

x座標: 2, y座標: 1, 幅: 10, 高さ: 12

(1)～(3)を確認するMain1のmainメソッド

```
public static void main(String[] args) {  
    Rectangle rectangle1= new Rectangle(2,1,5,6);  
    int area = rectangle1.calcArea();  
    int around = rectangle1.calcAround();  
    rectangle1.showState();  
    System.out.printf("の長方形の面積は%d、周りの長さは%dです . ¥n",  
                      area, around);  
  
    rectangle1.half();  
    rectangle1.showState();  
    rectangle1.doublify();  
    rectangle1.showState();  
}
```

x座標: 2, y座標: 1, 幅: 5, 高さ: 6

の長方形の面積は 30、周りの長さは 22 です.

x座標: 2, y座標: 1, 幅: 2, 高さ: 3

x座標: 2, y座標: 1, 幅: 4, 高さ: 6

(1)～(4)を確認するMain1のmainメソッド

```
public static void main(String[] args) {  
    Rectangle rectangle1= new Rectangle(2,1,5,6);  
    int area = rectangle1.calcArea();  
    int around = rectangle1.calcAround();  
    rectangle1.showState();  
    System.out.printf("の長方形の面積は%d、周りの長さは%dです . ¥n",  
                      area, around);  
  
    rectangle1.rotate90();  
    rectangle1.half();  
    rectangle1.showState();  
    rectangle1.doublify();  
    rectangle1.showState();  
}
```

x座標: 2, y座標: 1, 幅: 5, 高さ: 6

の長方形の面積は 30、周りの長さは 22 です.

x座標: 2, y座標: 1, 幅: 3, 高さ: 2

x座標: 2, y座標: 1, 幅: 6, 高さ: 4