

Point2D

Einfalt dæmi, gagnatag fyrir tvívíð hnit.

```
public class Point2D {  
    public double x,y;  
    public Point2D() {  
        x = 0.0;  
        y = 0.0;  
    }  
}
```

Point2D

Einfalt dæmi, gagnatag fyrir tvívíð hnit.

```
public class Point2D {  
    public double x,y;  
    public Point2D() {  
        x = 0.0;  
        y = 0.0;  
    }  
}
```

Vandamál: aðrir partar forrits geta breytt hlutnum

Point2D

```
public class Point2D {
    public double x,y;
    public Point2D() {
        x = 0.0;
        y = 0.0;
    }
}

...
Point2D p = new Point2D();
Point2D r = p;
r.y = 3.0; // nú er p líka breyttur!
```

Point2D

```
public class Point2D {  
    private double x,y;  
    public Point2D(double x, double y) {  
        this.x = x;  
        this.y = y;  
    }  
    public double getX() { return x;}  
    public double getY() { return y;}  
    public void setX(double x) { this.x = x;}  
    public void setY(double y) { this.y = y;}  
}
```

Vandamálið leyst?

Point2D

```
public class Point2D {  
    private double x,y;  
    Point2D(double x, double y) {  
        this.x = x;  
        this.y = y;  
    }  
    public double getX() { return x;}  
    public double getY() { return y;}  
    public void setX(double x) { this.x = x;}  
    public void setY(double y) { this.y = y;}  
}
```

Vandamálið leyst?

```
Point2D p = new Point2D(1.0,3.0);  
r = p;  
r.setX(3.0); // alveg eins og áður!
```

Point2D

Af hverju er þetta vandamál?

Ef við skilgreinum p og höldum að hluturinn sé einhverju ástandi þá geta aðrir hlutar forrits breytt því án þess að við tökum eftir því!

```
Point2D p = new Point2D(x,y);
```

```
// athugum að p sé í [0,1] x [0,1]
```

```
fall_af_hinu_illa(p) // breytir gildinu á p, p  
gæti verið
```

```
// hvað sem er
```

```
... // nú er p ekki í löglegu ástandi
```

Point2D

Lausnin er að láta Point2D vera óbreytanlegan (Immutable).

```
public class Point2d {  
    private final double x,y;  
    public Point2D(double x, double y) {  
        this.x = x;  
        this.y = y;  
    }  
    public double getX() { return x;}  
    public double getY() { return y;}  
}
```

Nú er ekki hægt að breyta p, við getum bara búið til nýja hluti.

String klasinn

String klasinn sem geymir strengi í Java er óbreytanlegur.

```
String s = "Hello";  
String r = s;  
for (int i = 0; i < 10; i++) {  
    s += i;  
}
```

```
System.out.println(s);  
System.out.println(r);
```

Hvað prentast út?

Immutable

Óbreytanlegir hlutir kosta aðeins meira

- Þurfum að búa til nýjan hlut í hvert skipti sem við viljum breyta einhverju
- Nýir hlutir verða til í kös (heap) og kosta minni
- Ruslasafnarinn fær nóg að gera

- Auðveldara að skrifa rétt forrit með óbreytanlegum hlutum
- Margir partar forrits geta deilt sama óbreytanlega hlut (þ.e. tilvísun) án þess að hafa áhyggjur.

Rectangle

Notum Point2D til að búa til ferhyrningaklasa

```
public class Rectangle {  
    // tilviksbreytur ...  
  
    public Rectangle(double x0, double y0,  
                    double x1, double y1) { ... }  
    public Rectangle(Point2D p1, Point2D p2) { ... }  
  
    public double getWidth() { ... }  
    public double getHeight() { ... }  
    public Point2D getPos() { ... }  
}
```

Hvernig eigum við að geyma gildin? Hvaða tilviksbreytur?