

#####

[illegible][illegible]

```
class Rectangle {
private int x1;
private int y1;
private int x2;
private int y2;
}
```

1





#####

```
@Bound int fillBlue;
@Bound int borderRed;
@Bound int borderGreen;
@Bound int borderBlue;
```

... ## ##### ## ##### ##?

```
@Bound RgbColor fillColor;
@Bound RgbColor borderColor;
```

#####. ## ## ##### ## ##### 5, #####, ##### *can* #####  
#####. #####.

##### 5. #####

```
class RgbColor {
    @Bound int red;
    @Bound int green;
    @Bound int blue;
}
```

##### – ##### # ##### ##### ## – ##### ##  
#####. #####. ##### 6 #####  
#####. #####, ##### ## ## ##### #####  
##### ## ##### ## #####.

##### 6, ##### ##### ## ##  
#####.

##### 6. #####

```
class Rectangle {
    @Bound private RgbColor fillColor;
    @Bound private RgbColor borderColor;
    @Bound private int x1;
    @Bound private int y1;
    @Bound private int x2;
    @Bound private int y2;
}
```

## 6. Inheritance

##### (##### 5, #####), ## ## ## ##  
#####. ##### ## ## #####  
#####. #####, ## *not* #####. ## #####, # #####  
#####.

##### 6, ##### ##### ## ##. ## ##, #####  
#####. ## ## ## #####. ## ## ## ##  
##### ## ##### ## #####?

```
## ##### ## ##### ##### #####, ##### #####. ##### ##, ##### ##
allways ##### ## ##### ## #####. ##### ## ## ##### ##### ## #
##### #####, ## ## ## ## ##### —##### ## ## ##### #####.
```

##### 7. #####

```
class Shape {
    @Bound fillColor;
    @Bound lineColor;
}
```

```
class Rectangle extends Shape {
    @Bound private int x1;
    @Bound private int y1;
    @Bound private int x2;
    @Bound private int y2;
}
```

```
class Circle extends Shape {
    @Bound int centerX;
    @Bound int centerY;
    @Bound int radius;
}
```

```
##### # ##### # ##### # #####, ## ### ##### ##### ##### ## ##### #####.
## ### # #####, ## ### ##### ##### ## ##### ## ##### ##### ##, ## ## ## ##
##### ## ## #####.
```

## 7. Lists

### #####—#### ## ##### ##### ## ## ##### ##### ## ## ##### #####, ### ###  
#####. ## ##### # ##### #####.

#####  
#####

```
#####  #####  ###  ####  #####  ##  #####  #  #####  -#####  #####  ##  #####  #, #####  ##  #####  #####  #####
#####  ##  #####  #####, ###  #####  #####  #####  #####  #####  #####:
```

```
class PseudoMondriaan {
    @BoundList(type=Rectangle.class, size="20")
    Rectangle[] rectangles;
}
```

```
### #### ##### ##### ##### ##### - ##### ## #### # #####, ###
20 ##### ##### ## ## #####, ## #### ## ## ## #####
#####.
```

```
###, ## ## ## ##### ##### ## ## ## ##### ## ##### #### ##### #####
##### 20 #####. #####, ### ##### ##### ##### ##### ##### #####
##### ## #####. #### ## #### ## ##### 9, #####.
```



#####

```
@BoundList(size="width * height") byte[] pixels;
@BoundNumber(size="nrBits * 2") int value;
// Clearly pointless, but you know...
@BoundString(size="x * (y + z) / 23 ^ t");
```

## 10. Limbo

#####  
#####  
#####  
#####

#####  
#####  
#####  
#####

#####  
#####  
#####

width \* height  
 $a^2 + b^2 == c^2$

... #####

- #####
- #####

#####

## 11. Conditionals

#####  
#####  
#####

#####  
#####  
#####

#####  
#####  
#####

#####  
#####  
#####





#####

#####  
#####  
#####  
#####  
#####  
#####

#####  
#####  
#####  
#####  
#####  
#####

##,###  
#####  
#####  
#####  
#####  
#####

Codec<Rectangle> codec = Codecs.create(Rectangle.class);  
Rectangle rect = Codecs.decode(codec, in);

#####

Codec<Rectangle> codec = Codecs.create(Rectangle.class);  
Codecs.document(codec, DocumentType.Html, new File(...);

##### 7, #####

##### 1. #####