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

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

1. Introduction

######## # ########## #### ###### ## #### ### ## ##### # ####### ####. #### ####### #### ######### ######## ### #### ## #### ### #### ## ## ### #### ###########. ##### #### ## ####### # ###### ### ######## ########.

#####. ##### ###### #### ####### ## ## ## ########## ######### ##### ### ########## ##### ### ## #####. ############ ## ### ########## ## ## ######## ## ##### ## ####### #### ## ###; ## ## ########## ## ## ######## ## ##### ## ####### #### ## ## ##### ###. ### #### ######### ## #### ####### ## ## ####### ######### #### ### ##### ## ### #####.

2. Declarative Binding

#########, ## # ####### #### ### # ###### ## ########### ########### ######## ### #### #### #### # ########## ######### ######## ######. ## ## #### #### #### ####, ### – #### #### ### – ## ##### ### ######## ########: ###### #### ##, ## #### ###.#

##########, #### ### ##### #### #### ######## ## #### ### ####### #### #### #### ### #### ########## ########. ### ##### ##### #### ########: ### #### ####### ## ### ########## ## ### #### #########, ### ##### #### #### #### ### #### #### ###### #### ### ########### ######.

###, ### ### ######## ## ## ######: ### #### ##### #### ### ## #### #### #### # ######## ## #### ### ########### # #### #########? #### ## ### ######? ## ##### ###, ### ###### ## ### #### ######### ######; #### #### ####### ### ##########, ######. ### ## #### ### ########### ######## ## ### #### ######### ## ### ######, ## #### #### ####### ## ##### ###########.

##, ##### #### ## ## #######. ####### 1, ###### #### ########## ####### # #### ######### ######## ### ########### ## ### ### ###### ######## # #########, ## #### ########## ######.

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

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

32–### ########. ## #### ####, #### ## ### ### ### ##### ###### # ######### #### # ####:

```
byte[] in = new byte[] {
  0, 0, 0, 1,
  0, 0, 0, 2,
  0, 0, 0, 3,
  0, 0, 0, 4
};
Codec<Rectangle> codec = Codecs.create(Rectangle.class);
Rectangle rect = Codecs.decode(codec, in);
```

##. ########... ## ## ######### ######, ### ####### ## ### ######## ######## ###. ## ### ##### ### ### #### ##–##, ### ##### ### ### ######## ## ###. (####, ## ##### #########, #####.) ####### ##, ##### #### *not* ###### ######## ###### ### ########## ####### ## ##### #########. ## ##### ## #### ##### #### ## ##### ## ###### ### #1, #1, #2 ### #2 ######, ### #### #### ## ########## ##### ## ## ### ########## ## ### #### #########, #### ####:

2. ##### #### #########

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

###, ### #### ### # ######### #########.

3. Convention over configuration

##, ####### # #### #### ######### ## ### ####### ############## ##### ## ## ###### ####. ########, ##### ####### ## ####### ######### ####### ####. #### ##, ## #### ##### ### #### ##### #### ## ######## ### #### 32 #### ### ######## ## #######, ### ### ##### #### #### ## # ###### ## #### #######. ## #### ## ### ##### #### ### ###### #### #####, ### ###### ###### #### ##### #######?

####. ## ### #### ####### ### ###### ##########, ### ######### #### ##### ## ### ### ####### ####### ## ###### ## #######. ### ## #######, ## #### ###### ### ###### #### #####, ### ###### ## ####### ##### 32 ####. ## ### #### ## ###### ####, ### #### #### ## #### ##### ## ## ##, ##### ##### ###########, ## (#########) ##### ########## ##########.

3. ######## ######## ##### ###

```
class Rectangle {
  @BoundNumber(size="16") private int x1;
  @BoundNumber(size="16") private int y1;
  @BoundNumber(size="16") private int x2;
  @BoundNumber(size="16") private int y2;
}
```

####### 3, ######### ######## ##### ### ###### ####### ### ####### ##### ###### ## #### ##### ######## #### ### ##### (16 ####) ### ######. ??? ########## ### ## ###### #### #####.

####### 4. ######## ######## ##### ###### ###### #### #####

```
class Rectangle {
  @BoundNumber(byteOrder=LittleEndian) private int x1;
  @BoundNumber(byteOrder=LittleEndian) private int y1;
  @BoundNumber(byteOrder=LittleEndian) private int x2;
  @BoundNumber(byteOrder=LittleEndian) private int y2;
}
```

4. More than just numbers

#######. ## #######, ## ####### ######## #### #### #### ####. #### ### #### ########:

```
// Will just read one bit, interpreting 1 as true, and 0 als false
@Bound boolean visible;

// Reads a String from a fixed number of bytes
@BoundString(size="10") String value;

// Reads a bit from the in, and interprets it as an enum value,
// interpreting the number as its ordinal value.
@BoundNumber(size="1") Type type;
```

##############; ## ###### ######### ## # ####### ######### #### ##### ##### ## #### ### ###, ### ######### ## ## #### ## #### #### ### ##### 1, ### ##### #########.

#######. #### #### (## #### ####) ### #### ######### ####### *the number of bytes* ### ### ### ###### ## ####. ## #### ####, ### ##### #### #### ## ########### ## ##–#####.

####–#### #### ##### ## ##### ## ### ### ##. #####, ## ###### ## ## ######## ########## #### ## #### ####### #### ######: ### ############ ########## ######### ### # ####### ####### ##### ##### ## ## ####. #### ####### ##### #### ##### ## ########## ## ### ####### ##### ## ### ## ### ###### ######.

5. Composite content

######: ## ### #####, ## ### #### ##### # ######## ## ##########. ### #### ## – ## #### ###### – ### #### #### #### ######–##### ########## ########### ## #### ########? ### ##### ### #### ## #### #### ####?

2, ###### #### ######### ########## ## ## #######. #### ## ## #### ## ## #### # #### ### ###### ##### ## ####? ##### #### ## ### ########?

```
@Bound int fillRed;
@Bound int fillGreen;
```

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

8. Lazy loading lists

#######; ##### #### ######## ##### (### ####.####.####) ####, ## ### ##### #### ########. ##### ## ####### # ########## #### ### ### ## ###### #####.

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

```
class PseudoMondriaan {
  @BoundList(size="20", type=Rectangle.class)
  private List<Rectangle> rectangles;
}
```

######. ## ### ##### ####, ### #### ## #### ######### ######## ## ##### ## #######. ## ####, ## #### #### ### ###### ## #### ######## ## ### ###### #### #### ## 20 ##### ### #### ## ###### #########. ### ## #### #### #### – ## ##### ## ### ### ##### ####### – ## #### #### ## #### #### ### ##### ### ######## (### ###### ## #### ## # ###### ######### #######, ########## ## ###).

##. ##### ### ##### ##### ## ##### #### ##### ## #### ######, ######### #### ### #### ###### #### ## ########## ## #######. ## ##### #### #####, ##### #### ############## #### ## ########## #####, #### ###### ### #### ####### ##### ### ####.

9. Expressions

###, ### ######## ######## # ###### ## ##### ### ### #### ######### ## ########## ### ############ ####### ###### ###### ####### ## #######. ## ##### ###### # ########, ### ### ######## ## ### ##### ## #### ### ## #### #### ### ##### ####### ###### ## ##### ##########.

##, ##### ########## ###### #### #### #### ####### ######: # ### ## ########## ## ##### ##### ### ## #### ## ###########, ####### ## ####### ######. ### #### ######### ## ########## ### ############ ### #### ######## ## ##### ## ##### ##### ########## ### ########.

```
class PseudoMondriaan {
  @Bound int numberOfRectangles;
  @BoundList(size="numberOfRectangles", type=Rectangle.class)
    List<Rectangle> rectangles;
}
```

#######. ## ###### ## # ######## ###### ##################, ### #### ### #### ## ##### ### #### ######## ## ####### ########: ## #######, ##### #### ### ## ####### ######### #### ##### ########## ####### ######. (##### ### ##### ##### ## ##########, ### ##### ##### #### #### ###### ####### ### ###.)

###, ### ### ######### ######### #### ####### ########## #### ####. ### ## ##### ## ## ####, ### ### ### # ##### ## ########## ## ####### #########. #### ### #### ########:

```
@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

#####. ##### ## # ###### ##### ########## ########, ### ## ########## ###### ## *not* ##### ## ######## #### ## ##### ## ## ####, ### ### #### ######.

###, ##### ## ### #### ##### ### ## #### ######## ## ########. ##### #### *not* ##### ### ## #### #######, ########*it wants to capture the meaning of the expression explicitly*. ### ## ##### ## ########## ## ######## ### ######## ## #### ########## ## ## ##### ######## ########## ######## ######## ## ##### ######. ##, ##### ##### ## #### ####, ### ### ###### ## ######### ## ######.

###########. ## #### ##### ## ######## ##### ########### ## #### # ### #### ## #### ## #### ## ######## *human-readable* ########### #### ## ##########. ##, ## ##### ##### ## ### ##### ###########:

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

... #### ##### ##### ## ## ######## ## ######## #### #### ####:

- ##### ## ### ##### ##### ### ###### ## ### #####
- ### ### ## # ## ### ##### ## 2 ### # ## ### ##### ## ### ###### # ## ### ##### ## ###

########, ## #### ### ### #### ## ########.

11. Conditionals

#########. ########, ## ##### ## ## #### ## #### #### ### ######## ## ######## #######, ## #### ## ######### # ###### ## ##### ######## ##########, ## ##### ## ######## #### ############.

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

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

###, ##### ### #### #### ##### ## ### ########. ##### #### ### #### # ##### ###### ### ## ######## #### ### #### # ###### ######, ### #### ## ######## #### ########## ########## ######## ### ## ######## ### ### ###### ### #####.

```
class Shape {
  @Bound Color fillColor;
  @Bound Color borderColor;
  @Bound boolean dashed;
  @If("dashed")
  @Bound int spaceBetweenDashes;
}
```

12. Complex references

####### 9, ############# ####### ######### #### ##### #### ### ##### ### ## ##### ## ##### ##### ########## ## ### #### ##### ####. ## #### ######## ### ### ######## (## ###### ########## ## ######## ##########) ### ### #### ######## (## ###### ##### ## ####–#### ##########). ####### 8, ###### ########### ##### #### ###### ##########.

8. #####

```
a.b.c
a.b[0].d
a.b[e].d
a.b[e * 21].d
```

####### ##### #### ##### ###### ### ## ## ## ## ##### ## ### *outer* #######. ####### 9, ###### ####### ########## ###### ## #### ## ##### ## ### bitsPerColorComponent ######## ## ### #####, #### ###### ### ######## ######.

9. ##### #######

```
public class Image {
  @Bound int bitsPerColorComponent;
  @Bound int nrColors;
  @BoundList(size="nrColors") RgbColor[] colors;
  ...
  public class RgbColor {
    @BoundNumber(size="outer.bitsPerColorComponent") red;
    @BoundNumber(size="outer.bitsPerColorComponent") green;
    @BoundNumber(size="outer.bitsPerColorComponent") blue;
  }
}
```

13. Documentation

####### ######## ## #### ####### ####### ####### ## ### #### #### ##### #### ## #########
########### ####### ######### ###### ## #### ##########, ## ##### ## ## #### ## ########
#############. ## #### ## ### ##### #### ### #### ##### #####, ### ##### ###### #### #### ### ### #####.

#########. ## #### ######## #########, ### ### ### ########## ### ### ######## ### ######## # ########## #### ######, ##### ####### ### ########## ### ######## ####; #### ## #### #### ########## ## #### ############ ## ####. ## ### ###, ########## ###### ## ## ### ## ####. ####### ### ####### ## #### #### ### ####### ### ####, ### ### ############ ### ###### ### ## #####.

####: ## ### ## ## ####### # ######### ## ##### ### ####### ### ######## ###### ####, ### ### ########, ######## *and documentation* ### ####.

##, ### #### ## ####? ####, ## ### ######## ### ##### ####### ## #### #######, ## ########## # ##### ### #### # ##### ##### ### ###### #####, #### ####:

```
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. ######