

Week 7

Objects, Arrays and Methods



1

```
import java.awt.*;
public class Circle {

    // instance variables
    private int xCentre, yCentre;
    private int radius;
    private Color colour;
    private boolean isSolid;

    // constructor
    public Circle (int x, int y,
                  int r, Color c, boolean b) {
        xCentre = x;
        yCentre = y;
        radius = r;
        colour = c;
        isSolid = b;
    }
}
```

2

```
public int getXCentre ( ) {
    return xCentre;
}
public int getYCentre ( ) {
    return yCentre;
}
public void setCentre (int x, int y) {
    xCentre = x;
    yCentre = y;
}
public int getRadius ( ) {
    return radius;
}
public void setRadius (int r) {
    radius = r;
}
}
```

3

```
public Color getColour ( ) {
    return colour;
}

public void setColour (Color c) {
    colour = c;
}

public boolean getIsSolid ( ) {
    return isSolid;
}

public void setIsSolid (boolean b) {
    isSolid = b;
}
}
```

4

```
import java.awt.*;

public class DrawingCircles {

    public static void main (String [ ] args) {
        int num = 24;
        Circle [ ] circles = new Circle [num];
        for (int i=0; i<num; i++) {
            circles[i] =
                new Circle(244, 232,
                          10*i, Color.red, false);
        }
        CirclesWindow.create(circles);
    }
}
```

5

```
import java.awt.*;

public class DrawingCircles {

    private static Color [ ] colours = {Color.red,
        Color.blue, Color.green, Color.magenta,
        Color.yellow, Color.pink, Color.orange,
        Color.cyan, Color.black, Color.gray};

    public static void main (String [ ] args) {
        int num = 35;
        Circle [ ] circles = new Circle [num];
        for (int i=0; i<num; i++) {
            int posn =
                (int)(colours.length*Math.random( ));
            circles[i] = new Circle(244, 232,
                10*(num-i-1), colours[posn], true);
        }
        CirclesWindow.create(circles);
    }
}
```

6

```
public static void cycleColours (Circle [ ] c) {
    for (int i=0; i<c.length-1; i++) {
        c[i].setColour(c[i+1].getColour( ));
    }
    c[c.length-1].setColour(c[0].getColour( ));
}
```

7

```
public static void explode (Circle [ ] c) {
    int num = c.length;
    while (true) {
        Delay.milliseconds(10);
        for (int i=0; i<num; i++) {
            c[i].setRadius(c[i].getRadius( )+2);
            if (c[i].getRadius( ) > 100) {
                int x = (int)(400*Math.random( ))+50;
                int y = (int)(400*Math.random( ))+50;
                c[i].setCentre(x, y);
                int posn =
                    (int)(colours.length*Math.random( ));
                c[i].setColour(colours[posn]);
                c[i].setRadius(0);
            }
        }
        CirclesWindow.redraw(c);
    }
}
```

8

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Rabbits live in the above environment, moving freely between the fields. Rabbits have a fixed lifetime but are able to breed. If there are too many rabbits in the environment the available food is consumed and all the rabbits die.

9

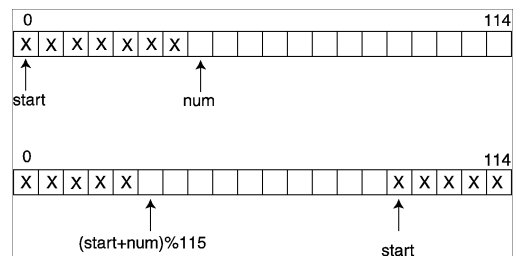
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

At each stage each rabbit moves to a vertical, horizontal or diagonal field with equal probability. After a given number of moves a rabbit dies of old age. If at any stage there are 2 or more rabbits in a field a new rabbit is born into that field; the new rabbit moves in the same way and has the same lifetime. If the population of rabbits reaches 100 or more they all die of starvation.

10

	n%4			
	0	1	2	3
0				
1	4	5	6	
2	8	9	10	
3	12	13	14	

11



12

```

public class Rabbit {

    // instance variables
    private int posn;
    private int age = 0;
    private int lifetime;

    //constructor methods
    public Rabbit (int lt) {
        posn = (int)(16*Math.random( ));
        lifetime = lt;
    }

    public Rabbit (int lt, int n) {
        posn = n;
        lifetime = lt;
    }
}

```

13

```

    public void move ( ) {
        if (age < lifetime) {
            int next = (int)(16*Math.random( ));
            while (Math.abs(posn%4-next%4)>1
                || Math.abs(posn/4-next/4)>1
                || next == posn)
                next = (int)(16*Math.random( ));
            posn = next;
            age++;
        }
    }

    public int getPosn ( ) {
        return posn;
    }

    public int getAge ( ) {
        return age;
    }
}

```

14

```

import genesis.*;

public class Simulation {

    private static int num; // class variables
    private static int lifetime;
    private static Rabbit [ ] rabbits = new Rabbit [115];
    private static int start;

    public static void removeDeadRabbits ( ) {
        for (int i=start; i<start+num; i++) {
            if (rabbits[i%115].getAge( ) == lifetime) {
                start = (start+1)%115;
                num = num-1;
            }
        }
    }
}

```

15

```

    public static void main (String [ ] args) {
        num = DialogBox.requestInt("number of rabbits:");
        lifetime = DialogBox.requestInt("rabbit lifetime:");
        start = 0;
        for (int i=0; i<num; i++) {
            rabbits[i] = new Rabbit(lifetime);
        }
        RabbitWindow.display(rabbits, 0, num);
        while (num > 0 && num < 100) {
            Delay.milliseconds(1000);
            for (int i=start; i<start+num; i++) {
                rabbits[i%115].move( );
            }
            removeDeadRabbits( );
            mateRabbits( );
            RabbitWindow.display(rabbits, start, num);
        }
    }
}

```

16

```

    public static void mateRabbits ( ) {
        int [ ] numbersAt = new int [16];
        for (int i=0; i<16; i++) numbersAt[i] = 0;
        for (int i=start; i<start+num; i++) {
            numbersAt[rabbits[i%115].getPosn( )]++;
        }
        for (int i=0; i<16; i++) {
            if (numbersAt[i] > 1) {
                rabbits[(start+num)%115]=new
Rabbit(lifetime,i);
                num++;
            }
        }
    }
}
}

```

17