

Class 4

10-16

概念明晰

- 类 (class)
- 对象 (object) = 属性 + 方法
- 类是对象的模版，而对象是类的实例化 (instance)。



- 变量 **variable**

- 基本变量类型

- int

- double

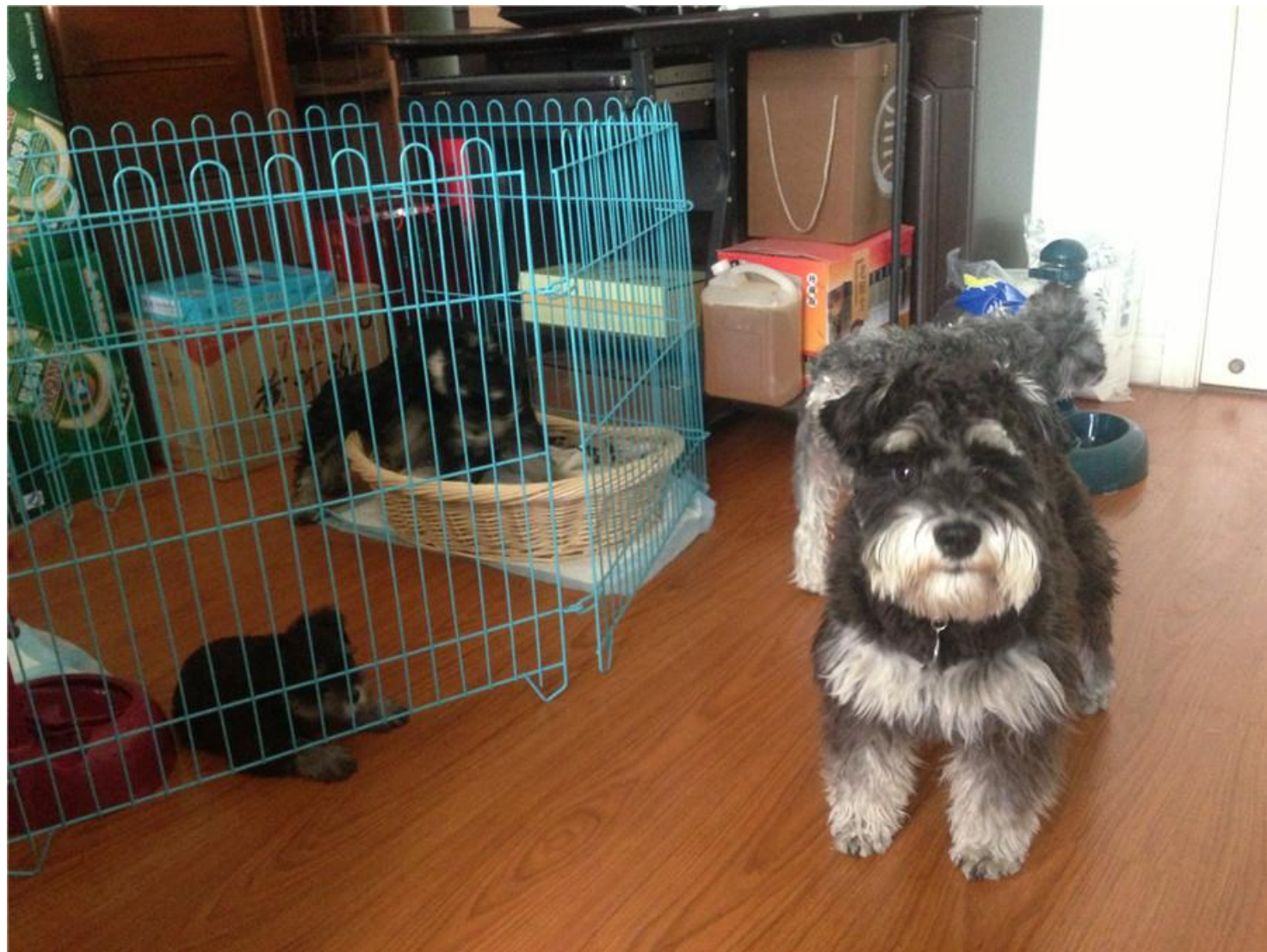
- boolean

- char

- String

- 声明变量

- 强制类型转换**CAST**



NOTE

- 在计算机的世界里， 我们喜欢从零开始
 - **Start with 0**
- `for(int i=0;i< 10;i++){`
- `}`

CAST强制类型转化

- `int <-> double`
 - `int x=3;`
 - `double y=4.2;`
 - `y=x;`
- `x=y;` **WRONG!!!**
 - **`x=(int) y;`**

方法Methods

```
public void run(){  
    String s=play();  
}  
  
private String play(){  
    String s=  
        “你是我天边最美的云彩~lalalalala”;  
    return s;  
}
```



Methods

- visibility **type** name(parameters){
 - ...body...
 - return **value**;
- }
- Vis: private or public
- Type: 返回值的类型//无返回值, type=void
- Parameter的语法: (type x1, type x2, type x3,.....)

return的类型

- return为空， void
- Return不为空时， 可以返回基本变量类型， int、 double、 boolean、 char、 String
- 当return为boolean类型时， 作为条件判断方法
- 当然也可以return复杂的变量类型或者是对象

计算x的n次幂

```
import acm.program.*;
public class NPower extends ConsoleProgram{
    public void run() {
        while(true) {
            int i=readInt("请输入一个数值: ");
            int j=readInt("请输入一个N次方值: ");
            println(i+"的"+j+"次方结果="+power(i,j));
        }
    }
    private int power(int i,int j){
        int result=1;
        while(j>0)
        {
            result*=i;
            j--;
        }
        return result;
    }
}
```

Math包

- **import java.lang.Math;**
- 调用方法
- **Math.方法();**

abs(int a)	返回 a的绝对值
cos(double a)	返回 a的余弦值
sin(double a)	返回a的正弦值
tan(double a)	返回a的正切值
max(double a,double b)	返回最大值
min(double a,double b)	最小值
pow(double a, double b)	a的b次方
sqrt(double a)	a的正平方根
round(double a)	a的四舍五入
random()	返回带正号的 double 值，大于或等于 0.0，小于 1.0

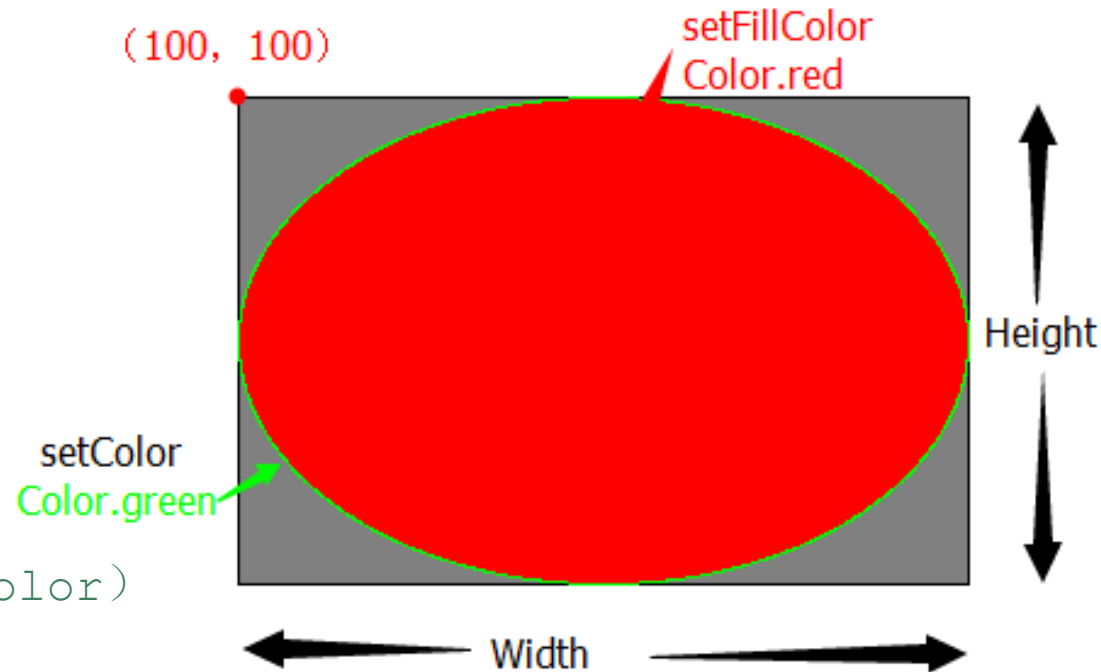
```
import java.awt.Color;
import acm.graphics.*;
import acm.program.*;
//画一个象棋的棋盘
public class Chess extends GraphicsProgram{
    private static final int BeeperWidth=50; //每个格子的长宽都为50
    public void run() {
        for(int i=0;i<8;i++){
            for(int j=0;j<8;j++){
                GRect Beeper=new GRect(BeeperWidth,BeeperWidth);
                if((i+j)%2!=0) //当小方格的坐标和刚好是偶数时
                    Beeper.setFilled(true); //方格实心
                add(Beeper,i*BeeperWidth,j*BeeperWidth);
            }
        }
    }
}
```

对象声明

- **GRect Beeper=new Grect (*Width , Height*);**
- ***add(Beeper, x, y);***

- **GRect Beeper=new Grect (*x , y , Width , Height*);**
- ***add(Beeper);***
- **//同样GOval**

```
import java.awt.Color;
import acm.graphics.*;
import acm.program.*;
public class Test extends GraphicsProgram{
    public void run() {
        GRect Beeper=new GRect(100,100,300,200);
        Beeper.setFilled(true);
        //是否填充颜色 (true or false)
        //默认情况下黑色
        Beeper.setFillColor(Color.gray);
        //填充什么颜色 (使用颜色时需要导入包 java.awt.Color)
        add(Beeper);
        GOval oval=new GOval(100,100,300,200);
        //GOval name= new GOval(x轴坐标, y轴坐标, 宽, 高);
        oval.setFilled(true);
        oval.setColor(Color.green);
        //setColor是设置边的颜色
        oval.setFillColor(Color.red);
        add(oval);
    }
}
```



返回对象object

```
private GOval Circle(double x,double y,double r,Color color) {  
    GOval circle= new GOval(x-r, y-r, 2*r,2*r);  
    circle.setFilled(true);  
    circle.setColor(Color);  
    return circle;  
}
```

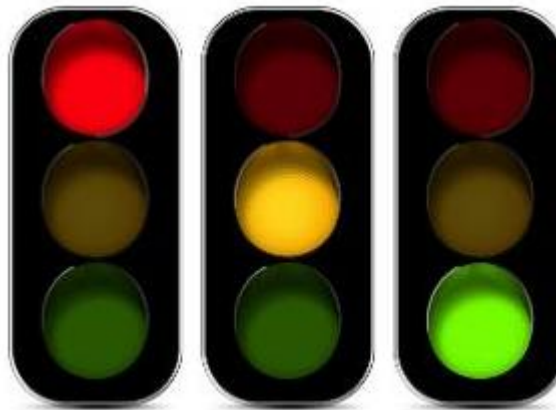
GraphicsProgram的方法（指定给画布的动作）

- add();
- getHeight();
- getWidth();
- pause(x);
- setSize(x,y)
- setLocation(x,y)

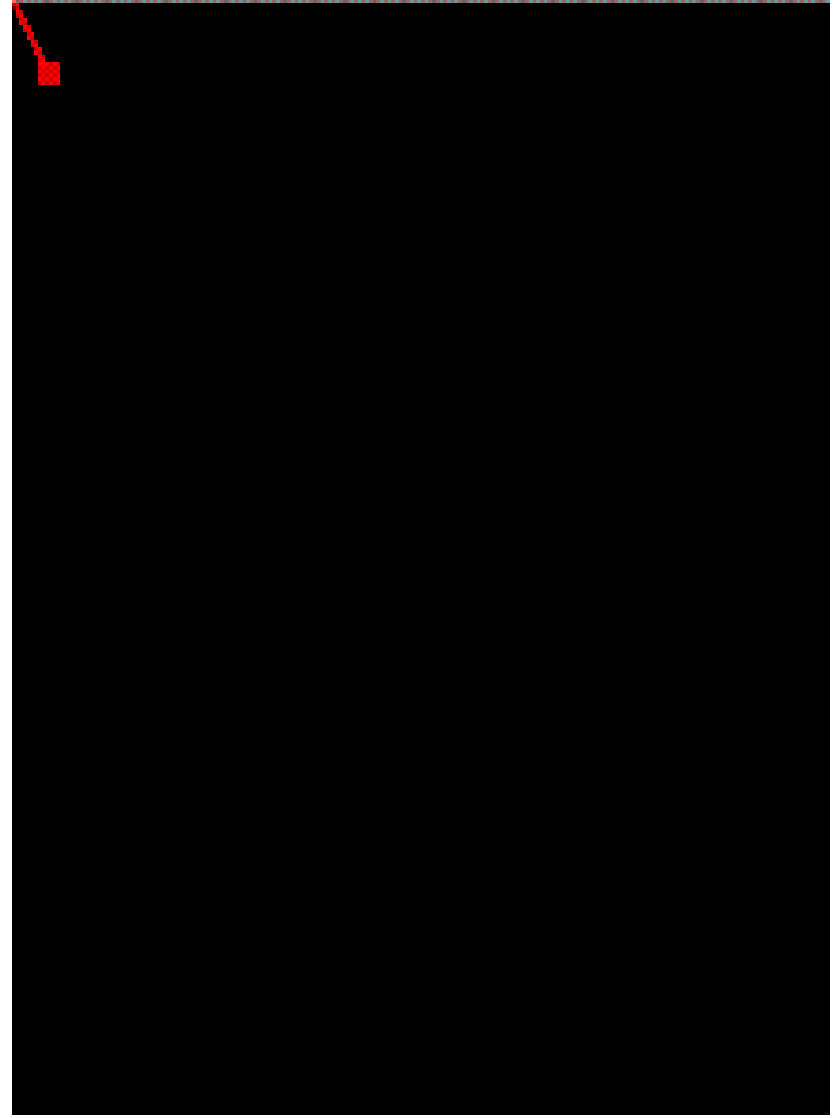
```
int h=getHeight(); //返回的画布宽和高都是整数型
int w=getWidth();
double h2=Beeper.getHeight(); //矩形的宽和高是浮点型
double w2=Beeper.getWidth();
println(h+" "+w);
println(h2+" "+w2);
```


Homework_TrafficLight

利用GraphicsProgram类编写一个红绿灯动态程序



反弹球 BlanceBall



参数传递（栈框架stack frame）

- 传递的不是参数本身，而是参数的复制品

```
public class NPower extends ConsoleProgram{  
    public void run() {  
        while(true) {  
            int i=readInt("请输入一个数值: ")  
            addOne(i);  
        }  
    }  
    private int addOne(int n) {  
        int i=n+1;  
        return i;  
    }  
}
```

```
public void run() {  
    ...  
    private int addOne(int n) {  
        int i=n+1;  
        return i;  
    }  
}
```

n

10

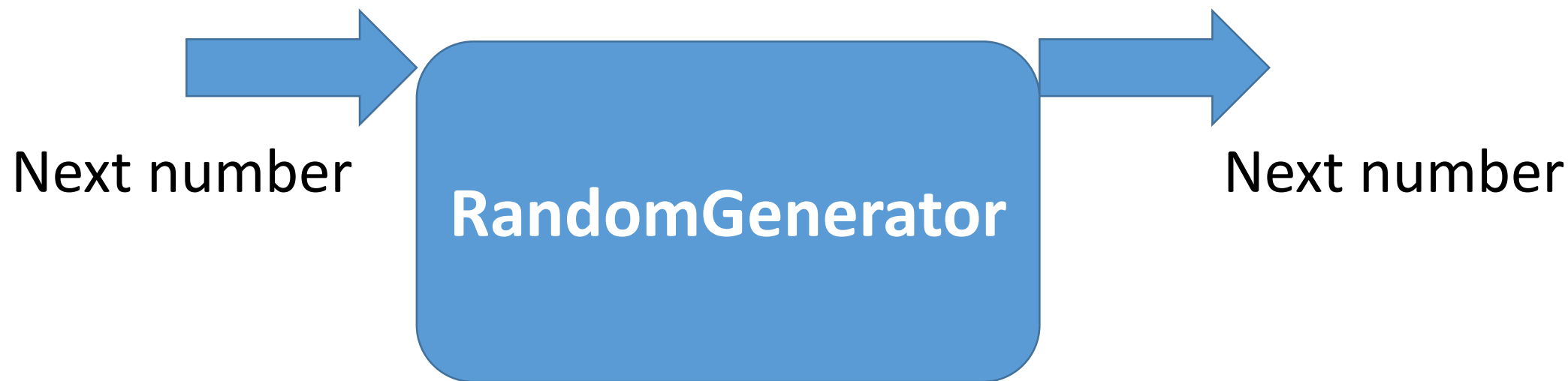
i

11

随机让生活变得更有趣



随机生成器



```
Import acm.util.*;
```

```
private RandomGenerator rgen=RandomGenerator.getInstance();
```

变量

局部变量 **Local variable**

- 在方法中声明
- 在方法中可见
- 生命随着方法的结束而结束
- 做局部计算时使用局部变量

实例变量 **instance variable**

- 在类中声明，而非方法
- 对整个对象中可用
- 对象消失，变量消失
- 对对象的状态进行计算时，采用实例变量
- 声明实例变量时，要声明它的可视范围 **visibility** (**private or public**)

随机变量

```
import acm.program.*;
import acm.util.*;

public class Ramdom extends ConsoleProgram{
    public void run() {

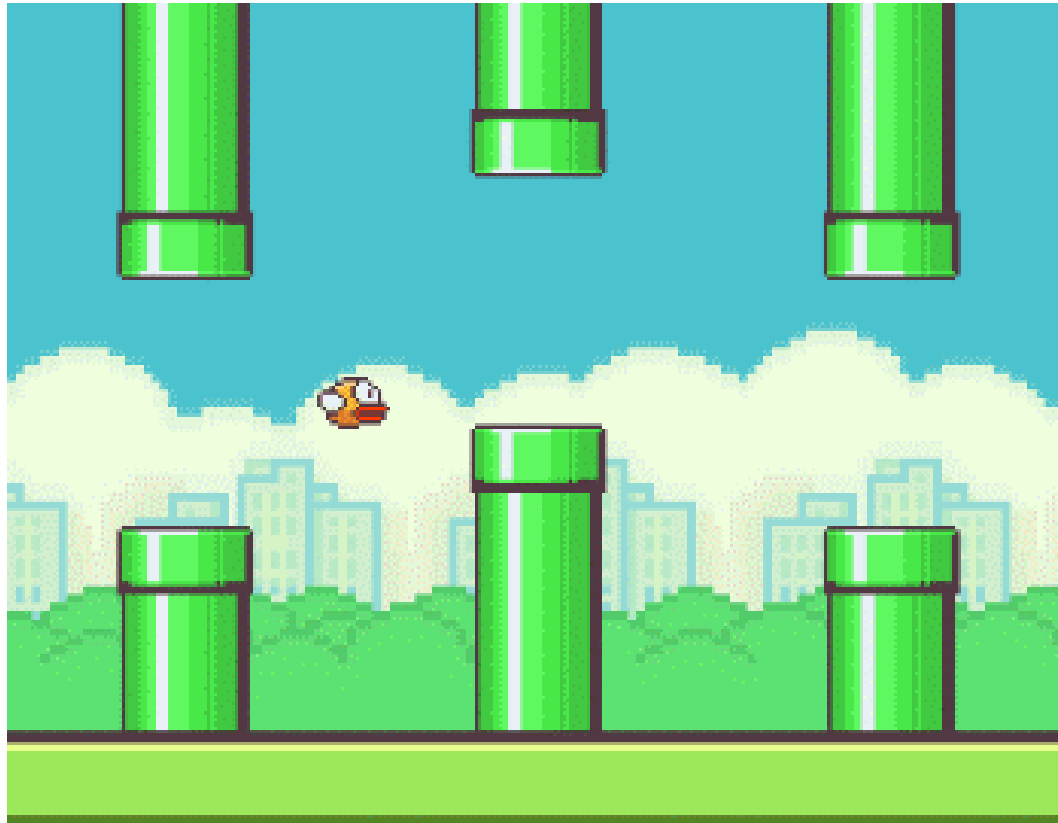
    }
    //私用实例变量
    private RandomGenerator rgen= RandomGenerator.getInstance();

}
```

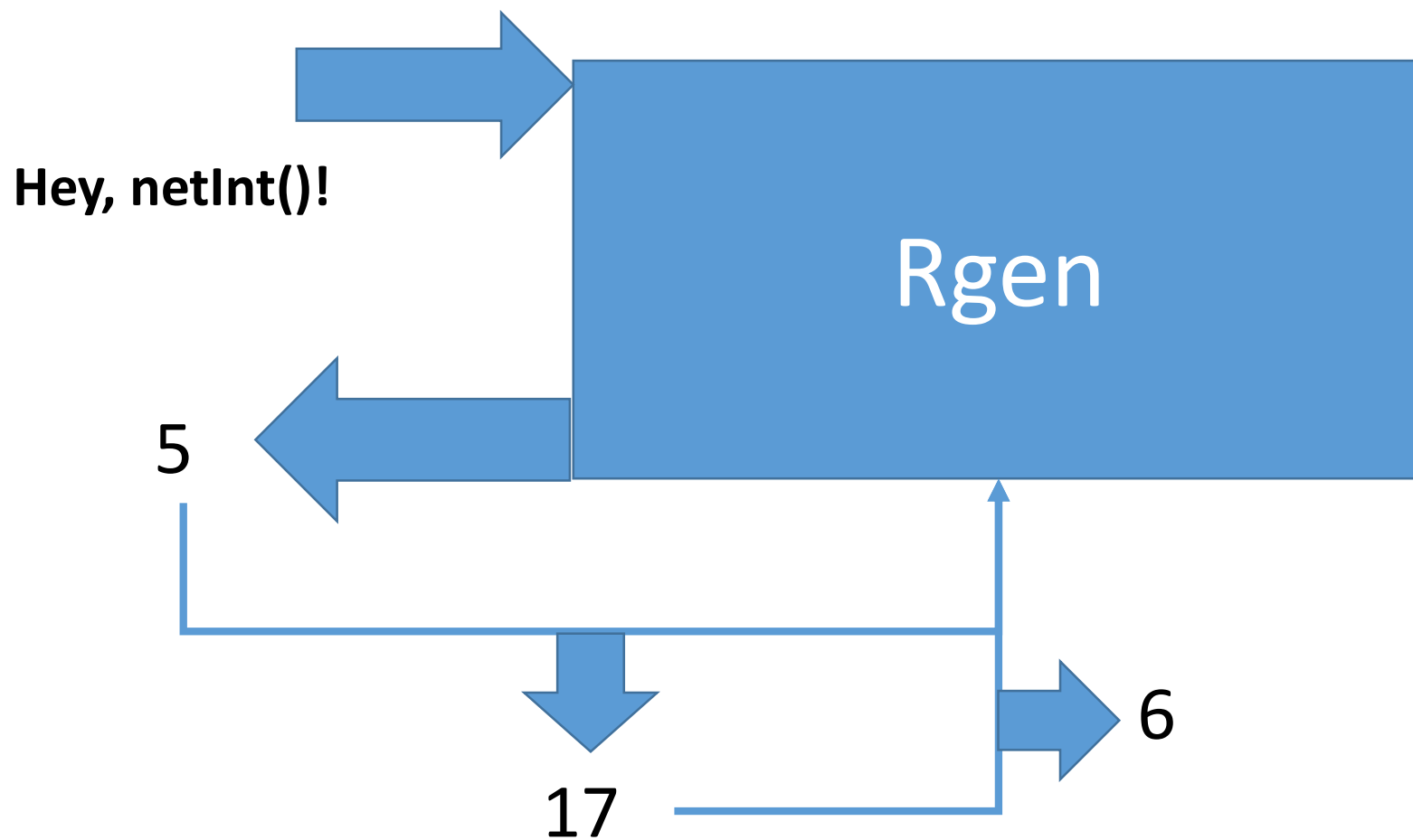

生成随机变量的方法

<code>int nextInt(int low, int high)</code>	随机返回low和high之间的一个整数
<code>int nextInt(int n)</code>	随机返回0和n-1之间的一个整数
<code>double nextDouble(double low, double high)</code>	随机返回low和high之间的一个浮点数
<code>double nextDouble()</code>	随机返回0和1之间的一个浮点数
<code>boolean nextBoolean()</code>	随机返回一个布尔值，真假各占50%概率
<code>boolean nextBoolean(double p)</code>	随机返回一个布尔值，为真的概率为p， $0 \leq p \leq 1$
<code>Color nextColor()</code>	随机返回一个颜色

Flappy bird



随机数的生成方式



给随机生成器一个种子Seed

- rgen.setSeed(int a);

```
import acm.program.*;
import acm.util.*;
public class Random extends ConsoleProgram{
    public void run() {
        rgen.setSeed(3);
        for(int i=0;i<10;i++){
            println(rgen.nextInt(1,10));
        }
    }
    //私用实例变量
    private RandomGenerator rgen= RandomGenerator.getInstance();
}
```

homework

- 利用RandomGenerator
- 在画布（500，500）的范围内，随机的位置
- 在(0,5000)的随机间隔时间画出半径大小在（0-50）之间的圆
- 颜色随机变化



See you next week !