



Review - Working With the Java API

I. Scott MacKenzie

1



DaysToSummer

```
import java.util.*;

public class DaysToSummer
{
    public static void main(String[] args)
    {
        Calendar today = new GregorianCalendar();

        Calendar summerStart =
            new GregorianCalendar(today.get(Calendar.YEAR), Calendar.JUNE, 21);

        int days = summerStart.get(Calendar.DAY_OF_YEAR) -
            today.get(Calendar.DAY_OF_YEAR);

        if (days > 0)
            System.out.println(days + " day(s) to summer");
        else if (days == 0)
            System.out.println("Summer begins today!");
        else
            System.out.println("Summer began " + -days + " day(s) ago");
    }
}
```

DaysToSummer.java

Demo

2



DemoDateTime

```
import java.util.*;
import java.text.*;

public class DemoDateTime
{
    public static void main(String[] args)
    {
        // get current date and time
        Date currentTime = new Date();

        // create SimpleDateFormat object with default format
        SimpleDateFormat sdf = new SimpleDateFormat();

        // print current date and time using default format
        System.out.println(sdf.format(currentTime));

        // set a new format
        sdf.applyPattern("'Date:' MMM d, yyyy 'Time:' hh:mm:ss a zzz");

        // print again using new format
        System.out.println(sdf.format(currentTime));

        // set a format for day of week
        sdf.applyPattern("'Today is' EEEE");

        // print day of the week using new format pattern
        System.out.println(sdf.format(currentTime));
    }
}
```

DemoDateTime.java

Demo

3



DemoClock

```
import java.util.*;

public class DemoClock
{
    public static void main(String[] args)
    {
        Calendar today = new GregorianCalendar();
        int seconds = today.get(Calendar.SECOND);
        while (true)
        {
            today.setTime(new Date());
            int newSeconds = today.get(Calendar.SECOND);
            if (newSeconds != seconds)
            {
                System.out.print(today.getTime() + "\r");
                seconds = newSeconds;
            }
        }
    }
}
```

DemoClock.java

Demo

4

RandomBits

```
import java.util.*;

public class RandomBits
{
    public static void main(String[] args)
    {
        long t1 = System.currentTimeMillis();
        int temp = 0;
        StringBuffer sb = new StringBuffer("          "); // 10 spaces
        Random r = new Random();
        while(true)
        {
            int t2 = (int)(System.currentTimeMillis() - t1) / 100;
            if (t2 != temp)
            {
                int i = r.nextInt(10);
                String bit = r.nextBoolean() ? "1" : "0";
                sb.replace(i, i + 1, bit);
                System.out.println(sb);
                temp = t2;
            }
        }
    }
}
```

RandomBits.java

Demo

RandomQuilt

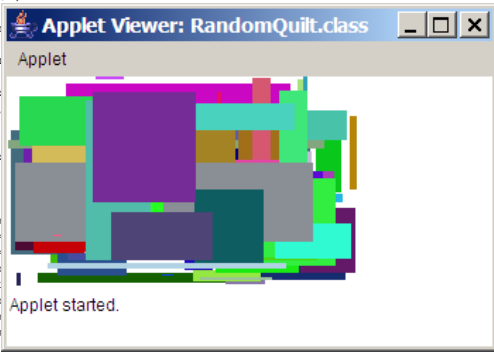
```
import java.awt.*;
import java.applet.*;

public class RandomQuilt
{
    public void paint(Graphics g)
    {
        long t1 = System.currentTimeMillis();
        int temp = 0;
        while (true)
        {
            // update:
            int t2 = (int)(System.currentTimeMillis() - t1) / 100;
            if (t2 != temp)
            {
                // generate random color and position
                int x = r.nextInt(width);
                int y = r.nextInt(height);
                int width = r.nextInt(100);
                int height = r.nextInt(100);
                int red = r.nextInt(256);
                int green = r.nextInt(256);
                int blue = r.nextInt(256);

                // set random color
                g.setColor(new Color(red, green, blue));

                // draw filled rectangle: random position, random proportions
                g.fillRect(x, y, width, height);

                temp = t2;
            }
        }
    }
}
```



RandomQuilt.java

Demo

