

# Java 课程上机报告：

## 上机题目：Java 语言基础

小组名单：

张俊华 16030199025

李金鑫 16030199026

李天浩 16030199027

**上机时间：2017/03/25 8:00-12:00**

**地点：EIII-204**

## 实验目标：

掌握 Java 语法；掌握 Java 程序结构；掌握 Java 编译、调试、运行的方法。

## 实验要求：

1. 练习 PPT 中老师上课要求做的小练习，尝试对小练习中各部分进行修改，并观察修改后的执行效果。（自觉完成不需要交）
2. 编写一个程序，程序提供两种功能：
  - 1) 根据用户输入的年份输出该年日历。
  - 2) 根据用户输入的日期，输出该天星期。

程序应具有良好的人机交互性能，即：程序应向用户提示功能说明，并可根据用户的功能选择，执行对应的功能，并给出带详细描述信息的最终执行结果。

## 一. 题目分析：

本实验要求编写一个程序，实现以下两种功能：

1. 根据用户输入的年份输出该年日历。
2. 根据用户输入的日期，输出该天星期。

其中，某天的星期可以这样计算：

1. 定义某一天是星期几（如：定义 1900 年 1 月 1 日为星期一）
2. 计算输入的日期距离与定义的日期相差几天
3. 相差的天数与 7 取余，便可得星期

查阅相关资料，星期还有公式计算方法：

$$w = (d + 2 * m + 3 * (m + 1) / 5 + y + y / 4 - y / 100 + y / 400) \% 7$$

其中：y 代表年（最后两位）；m 代表月（ $3 \leq m \leq 14$ ）1 月和 2 月当做上一年的 13 月和 14 月；d 代表这个月的第几天；w（ $0 \leq w \leq 6$ ）周一至周日分别对应数字 0~6

本实验决定采用公式计算星期。

输出年历的实现：

可将输出年历拆分成输出 12 个月历，可将月历视为一个二维数组。只需要确定每个月第一天是周几便可推算出每一天在数组中的位置。

## 二. 小组讨论内容：

张俊华：

1. 实验环境：

IntelliJ IDEA 2017.1 Build #IU-171.3780.107

JRE: 1.8.0\_112-release-736-b13 amd64

JVM: OpenJDK 64-Bit Server VM by JetBrains s.r.o

Windows 10 10.0

## 2. 实现过程:

首先实现查询某一天是周几，在 service 包中创建 FetchWeekday 类，实现该功能

fetchWeekday 方法的返回值为 0~6 的整数，对应周一到周日。

```
package service;

/**
 * Created by 张俊华 on 2017/3/18.
 */
public class FetchWeekDay {

    public static int fetchWeekday(int y, int m, int d) {
        /**
         *计算某一天周几
         *@param y 年
         *@param m 月
         *@param d 日
         *@return int w:0~6 对应周一 ~ 周日
         *@exception
         *@author 张俊华 16030199025
         */
        int w;
        y = y % 100; //按照公式，需要取年的最后两位
        switch (m) {
            case 1:
                m = 13;
                y -= 1;
                break;
            case 2:
                m = 14;
                y -= 1;
                break;
        } //按照公式，每年的一月二月当做上一年的 13、14 月

        w = (d + 2 * m + 3 * (m + 1) / 5 + y + y / 4 - y / 100 + y / 400) % 7;
        //计算公式
    }
}
```

调用本方法，即可获得每个月一日的星期，再得到每个月有几天，递推即可获得本月的月历。本程序中，将月历用数组保存。具体源码如下：

```
package service;

/**
 * Created by 张俊华 on 2017/3/18.
 */
public class MonthCalendar {

    private static int getMonthDay(int year, int month) {
        /**
         * 计算某年某月有几天
         * @param year 年
         * @param month 月
         * @return 某月的天数
         * @exception
         * @author 张俊华 16030199025
         */
        if ((month == 1) || (month == 3) || (month == 5) || (month == 7) ||
            (month == 8) || (month == 10) || (month == 12)) {
            return 31; // 1 3 5 7 8 10 12 月, 每个月有 31 天
        } else if (month == 2) {
            if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) {
                return 29; // 闰年
            } else return 28; // 非闰年
        } else {
            return 30;
        }
    }

    public static StringBuffer[] getMonthCalender(int year, int month) {
        /**
         * 获得一个月历数组
         * @param year 年
         * @param month 月
         * @return StringBuffer[] outMonth: 月历数组, 每天已经用\t分隔
         * @exception
         * @author 张俊华 16030199025
         */
    }
}
```

```

int date = 1;
int week = 0;
StringBuffer outWeek = new StringBuffer();
//初始化周字符串
StringBuffer[] outMonth = new StringBuffer[6];
// 初始化月数组
int firstWeekday = FetchWeekDay.fetchWeekday(year, month, 1);
//这个月第一天是星期几
int dayInThisMonth = MonthCalendar.getMonthDay(year, month);
//这个月有几天
for (int i = 0; i < firstWeekday; i++) {
    outWeek.append(" \t");
} //用空格和制表符对齐第一天的星期
for (int i = firstWeekday; i < 7; i++) {
    outWeek.append("\t").append(date);
    date++;
} //将第一周剩余的日子加入到第一周的字符串
outMonth[week++] = outWeek;
//将第一周加入到月数组
while (date <= dayInThisMonth) {
    outWeek = new StringBuffer();
    for (int i = 0; i < 7; i++) {
        outWeek.append("\t").append(date);
        date++;
        if (date > dayInThisMonth) {
            for (; i < 6; i++) {
                outWeek.append("\t ");
            } //最后一周补齐
            break; //输出到每个月最后一天
        }
    } //按顺序将每一天加入字符串
    outMonth[week++] = outWeek;
}
if (week < 6) {
    outMonth[5] = new StringBuffer("\t \t \t \t \t \t \t ");
}
return outMonth;
}

```

该方法直接返回一个 StringBuffer 类型的数组，按顺序打印即可得到完整月历。  
日历的各种功能实现均在 action 包中实现。

在本题目的基本要求之外，还实现了年历自定义行列数打印，只需要将列数当做参数传入。

main class 为整个程序的入口，在 main 类里实现对各种功能的调用。

```
package action;

import service.MonthCalendar;
import ui.UI;

/**
 * Created by 张俊华 on 2017/3/28.
 */
public class PrintYearCalendar {
    public static void main(int LINE) {
        /**
         *控制台输入任意年份，实现 m*n 年历的绘制，m 必须能被 12 整除
         *@param LINE
         *@return
         *@exception
         *@author 张俊华 16030199025
         */
        UI ui = new UI();
        int year = ui.scanYear();
        //初始化 ui 及 scanner
        StringBuffer yearCalendar[][] = new StringBuffer[12][6]; //年历数组
        for (int month = 1; month <= 12; month++) {
            yearCalendar[month-1] = MonthCalendar.getMonthCalender(year, month);
        } //调用 getMonthCalender 给年历数组赋值
        int month = 0;
        StringBuffer head = new StringBuffer(); //年历表头
        for (int i = 0; i < LINE; i++) {
            head.append("\t 一\t 二\t 三\t 四\t 五\t 六\t 日").append("\t\t");
        } //生成对应长度的表头
        for (int row = 0; row < 12/LINE; row++) {
            System.out.println(head);
            for (int r = 0; r < 6; r++) {
                for (int line = 0; line < LINE; line++) {
                    System.out.print(yearCalendar[month][r]);
                    System.out.print("\t\t");
                    month++;
                } //输出每一行
            }
        }
    }
}
```

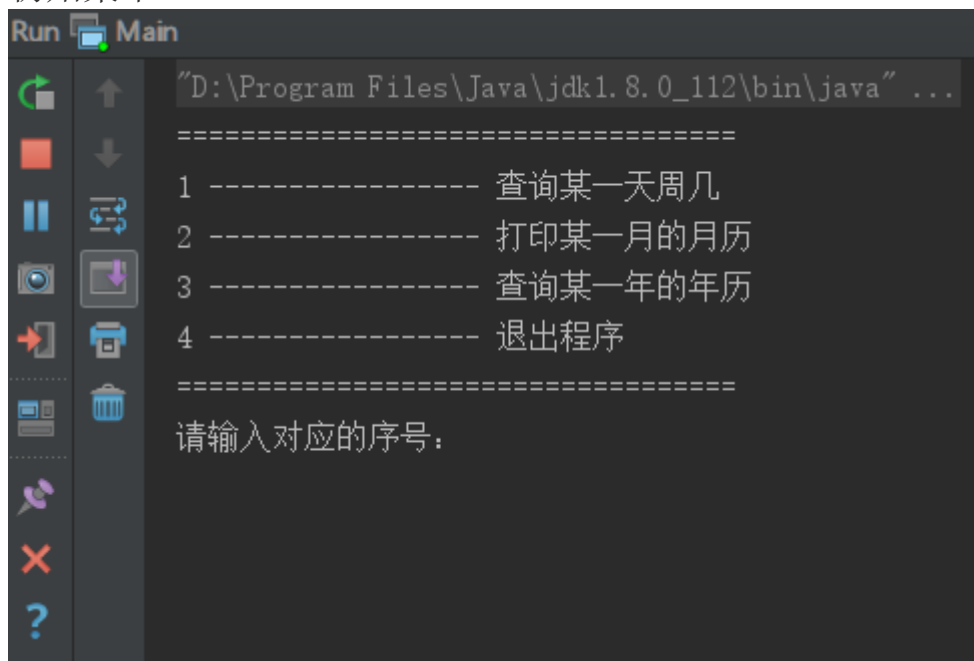
```

        month-=LINE; //下一行月份归位
        System.out.println();
    }
    month+=LINE; //月份递进
}
}

```

### 三. 程序结果：

初始菜单：



The screenshot shows a Java IDE window titled "Run Main". The command line shows the execution of a Java program. The output displays a menu with four options, separated by dashed lines. The prompt "请输入对应的序号：" (Please enter the corresponding serial number:) is shown at the bottom.

```

"D:\Program Files\Java\jdk1.8.0_112\bin\java" ...
=====
1 ----- 查询某一天周几
2 ----- 打印某一月的月历
3 ----- 查询某一年的年历
4 ----- 退出程序
=====
请输入对应的序号：

```

查询某一天星期几示例：

```

=====
1 ----- 查询某一天周几
2 ----- 打印某一月的月历
3 ----- 查询某一年的年历
4 ----- 退出程序
=====
请输入对应的序号： 1
请输入需要查询的年份： 2017
请输入需要查询的月份： 4
请输入需要查询的日期： 1
2017年4月1日是星期六

```

打印年历示例：

```
请输入对应的序号：3
请输入需要查询的年份：2017

一 二 三 四 五 六 日      一 二 三 四 五 六 日      一 二 三 四 五 六 日
      1                    1 2 3 4 5                    1 2 3 4 5
2 3 4 5 6 7 8      6 7 8 9 10 11 12      6 7 8 9 10 11 12
9 10 11 12 13 14 15      13 14 15 16 17 18 19      13 14 15 16 17 18 19
16 17 18 19 20 21 22      20 21 22 23 24 25 26      20 21 22 23 24 25 26
23 24 25 26 27 28 29      27 28                    27 28 29 30 31
30 31

一 二 三 四 五 六 日      一 二 三 四 五 六 日      一 二 三 四 五 六 日
      1 2                    1 2 3 4 5 6 7                    1 2 3 4
3 4 5 6 7 8 9      8 9 10 11 12 13 14      5 6 7 8 9 10 11
10 11 12 13 14 15 16      15 16 17 18 19 20 21      12 13 14 15 16 17 18
17 18 19 20 21 22 23      22 23 24 25 26 27 28      19 20 21 22 23 24 25
24 25 26 27 28 29 30      29 30 31                    26 27 28 29 30

一 二 三 四 五 六 日      一 二 三 四 五 六 日      一 二 三 四 五 六 日
      1 2                    1 2 3 4 5 6                    1 2 3
3 4 5 6 7 8 9      7 8 9 10 11 12 13      4 5 6 7 8 9 10
10 11 12 13 14 15 16      14 15 16 17 18 19 20      11 12 13 14 15 16 17
17 18 19 20 21 22 23      21 22 23 24 25 26 27      18 19 20 21 22 23 24
24 25 26 27 28 29 30      28 29 30 31                    25 26 27 28 29 30
31

一 二 三 四 五 六 日      一 二 三 四 五 六 日      一 二 三 四 五 六 日
      1                    1 2 3 4 5                    1 2 3
2 3 4 5 6 7 8      6 7 8 9 10 11 12      4 5 6 7 8 9 10
9 10 11 12 13 14 15      13 14 15 16 17 18 19      11 12 13 14 15 16 17
16 17 18 19 20 21 22      20 21 22 23 24 25 26      18 19 20 21 22 23 24
23 24 25 26 27 28 29      27 28 29 30                    25 26 27 28 29 30 31
30 31
```

## 四. 程序源码：

本程序已经在 github 开源。项目地址：

<https://github.com/SincereXIA/miniJavaCalendar>