Lecture Notes: ESC 101

Date: 26/02/2008

```
// Program to Display calendar of a given month

import java.util.Scanner;

class Year
{
	int year;
	Year() {
		this.year=2000;
	}

	Year(int y) {
		year=(y>1900)? y:1900;  //if year less than 1900, set it to 1900
	}

	Year(Year y) {  //copy constructor
	// used during assignment operation between two year elements

		this.year = y.year;
	}

	boolean isLeap(int year)
	{
		if((year % 400)==0) return true;//if divisible by 400 return true
		if((year % 100)==0) return false;//if divisible by 100 but not by 400
		if((year % 4)==0) return true;
		return false;
	}

	int weekDay(){
		int date  = Date.MON;     //First day of week
		int i;
		//To find the day on 1st Jan of Current year
		for(i=1900;i<year;i++)
		{
			date = (date + 365+ (isLeap(i)?1 : 0)) % 7;
		}
		return date;
	}
```

```
        public String toString(){
                return "" + year;  //return year as string

        }
}

class Month{

        public final static int JAN = 1;
        public final static int FEB = 2;
        public final static int MAR = 3;
        public final static int APR = 4;
        public final static int MAY = 5;
        public final static int JUN = 6;
        public final static int JUL = 7;
        public final static int AUG = 8;
        public final static int SEP = 9;
        public final static int OCT = 10;
        public final static int NOV = 11;
        public final static int DEC = 12;


        public final static String monthShortName[] =
{"JAN","FEB","MAR","APR","MAY","JUN","JUL,"AUG","SEP","OCT","NOV","DEC"};

        public final static String monthFullName[] =
{"JANUARY","FEBRUARY","MARCH","APRIL","MAY","JUNE","JULY","AUGUST","SEPTEMBE
R","OCTOBER","NOVENBER","DECEMBER"};

        private int month;  //int as private member
        private Year year; //object of type Year as private member

        Month(int month,int year){

                this.month = ((month < JAN) || (month > DEC)) ? JAN:month ;
                //Assign this.month with month in parameter otherwise with JAN if
                month in parameter is less than 0(JAN) or greater 12(DEC)
                this.year = new Year(year);
                }

        Month(int month){

                this.month  = ((month < JAN) || (month > DEC ))?JAN:month;
                this.year = new Year();
         }
```

```
Month(){

        this.month = JAN;
        this.year = new Year();

}

Month(Month m){
        this.month = m.month;
        this.year = new Year(m.year);

}



//find the no of days in the month.


private int lastDay(int month){

int i = this.year.year;

        switch(month){ //using switch case on month to find the no: of days in
the month.

                case MAR:
                case JUN:
                case SEP:
                case NOV:
                        return 30;
                case FEB:
                        return ((year.isLeap(i))?29:28);

                default :
                        return 31;

        }

}

int lastDay(){

return lastDay(month); //calling the lastday fun. with current month as
parameter to it.

}
```

```
        //To find the day on 1st of Current month

        int weekDay(){

                int i;
                int first = year.weekDay();

                for(i = JAN; i<month; i++){

                        first = (first + lastDay(i)) % 7;
                }

                return first;

        }


        public String toString(){

                return monthFullName[month - 1] + " " + year;

        }

        public int[][] calender()
        {

        int[][] calArray = new int[5][7]; // A month can have at most 5 weeks and 7
        days per week... 4 entries will remain vacant

        int i = weekDay();// Day of the 1st of the current month
        int j;

        for(j=0; j<lastDay(); j++){

                int t = i+j;
                calArray[(t/7)%5][t%7] = j+1;

        }

                return calArray;
        }

}
```

```
class Date{

        public final static int SUN=0;
        public final static int MON=1;
        public final static int TUE=2;
        public final static int WED=3;
        public final static int THU=4;
        public final static int FRI=5;
        public final static int SAT=6;

        public final static String dayShortName[] = {
"SUN","MON","TUE","WED","THU", "FRI","SAT"};
        public final static String dayFullName[] =
{"SUNDAY","MONDAY","TUESDAY","WEDNESDAY","THURSDAY", "FRIDAY","SATURDAY"};

        private int day;
        Month month;

        Date(int day,int month,int year)
        {
                this.month =new Month(month,year);
                this.day=(day>this.month.lastDay()||day<1)?1:day;
        }

        Date(int day,int month)
        {
                this.month =new Month(month);
                this.day=(day>this.month.lastDay()||day<1)?1:day;
        }
        Date(int day)
        {
                this.month =new Month();
                this.day=(day>this.month.lastDay()||day<1)?1:day;
        }
        Date()
        {
                this.month=new Month();
                this.day=1;
        }

        Date (Date d)
        {
                this.month=new Month(d.month);
                this.day=d.day;
        }
```

```
		//To find the day on current date

		int weekDay()
		{
			return (this.month.weekDay()+this.day-1)%7;
		}
		public String toString()
		{
			return ""+day+" "+month;
		}
}
```

class Calender{

```
	public static void main(String a[]){

		Scanner s = new Scanner (System.in);

		System.out.println(" Enter day (1-31) month (1-12) and year
(>1900)");

		int day=s.nextInt();
		int month=s.nextInt();
		int year=s.nextInt();

		Date d = new Date(day,month,year);
		System.out.println(d+" is / was a "+
Date.dayFullName[d.weekDay()]+"\n\n");
		int [][]cal = d.month.calender();

		System.out.println(d.month);
		int i=0, j=0;
		for(j=0;j<7;j++)
		{
			System.out.print(" "+Date.dayShortName[j]);
		}

		System.out.println(" ");

		for(i=0;i<5;i++)
		{
			for(j=0;j<7;j++)
			{
				if(cal[i][j]==0)
				{
					System.out.print("  ");
```

```
                        }
                        else if( cal[i][j]<10)
                        {
                                System.out.print(" "+cal[i][j]);
                        }
                        else
                        {
                                System.out.print(" "+cal[i][j]);
                        }
                }
                System.out.println("");
        }
    }
}
```

Multi- Dimensional Arrays:

- Once an array is created its size cannot be changed.
- Box is abstraction of memory.
- Unit of memory is Byte.

  Byte is a data type that occupies 1 memory location and Short is a data type that occupies 2 memory locations etc.

  There is a mechanism by which array can be represented in memory.

  Eg. Byte array[5](1-D array of 5 Bytes) will occupy 5 memory locations.

2 D Array of Bytes:

Consider an array, Byte array[5][7].

Its elements will be stored in following way:

0,0	0,1	0,2	0,3	0,4	0,5	0,6
1,0	1,1	1,2	1,3	1,4	1,5	1,6
2,0	2,1	2,2	2,3	2,4	2,5	2,6
3,0	3,1	3,2	3,3	3,4	3,5	3,6
4,0	4,1	4,2	4,3	4,4	4,5	4,6

First memory location will contain array element [0][0]. Next memory location will contain element [0][1] and so on. Here elements are stored Row-wise. This way of storing elements is referred as “Row Major Order”. Java uses Row major order.

There is another way of storing these elements in which first memory location contains element [0][0] and next memory location contains element[1][0] i.e. here elements are stored Column-wise. This way of storing is referred as “Column Major Order”.