German University in Cairo
Media Engineering and Technology
Prof. Dr. Slim Abdennadher
Dr. Wael Abouelsaadat

Introduction to Computer Programming, Spring Term 2016
Practice Assignment 10

Discussion: 20.5.2017 - 25.5.2017

Exercise 10-1 MinRec
Final Exam 2016

Write a recursive method `MinRec` that given an array of integers finds the smallest value in the given array.

Note that you are not allowed to use any additional arrays.

Running the following `main` method:

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

    int[] a = {1,2,3,4};
    int[] b = {5,2};
    int[] c = {6,6};
    System.out.println(MinRec(a));
    System.out.println(MinRec(b));
    System.out.println(MinRec(c));
}
```

will display

```
1
2
6
```

Solution:

```
public static int MinRec(int []a){
     return helper(a,1,a[0]);
}

public static int helper(int []a, int i , int min){
     if(i == a.length)
          return min;
     else if(a[i] < min)
     {
         min = a[i];
         return helper(a,++i, min);
     }
     else
          return helper(a,++i,min);
}
```

Exercise 10-2 MergeRec
Final Exam 2015

Write a recursive method `mergeRec` that given two array of integers displays the elements of the given arrays in an alternating way. Note that the two arrays could be of different length.

Note that you are not allowed to use any additional arrays. Once you execute the following main method

```
public static void main(String[] args) {
  int[] a = {1,8,3,4};
  int[] b = {5,2};
  mergeRec(a,b);
}
```

the following should be displayed:

```
1 5 8 2 3 4
```

Solution:

```
public static void mergeRec(int[]a, int[]b)
{
    helper(a,b,0,0);
}
public static void helper (int[]a, int[]b, int i, int j)
{
    if(a.length==i && b.length==j)
    {
        return;
    }
    if(b.length==j)
    {
        System.out.print(a[i]+" ");
        helper (a,b,++i,j);
    }
    else{
        if(a.length==i)
        {
            System.out.print(b[j] + " ");
            helper (a,b,i,++j);
        }
        else
        {
            System.out.print(a[i] + " " + b[j] +" ");
            helper(a,b,++i,++j);
        }
    }
}
```

Exercise 10-3 Student
To be discussed in Tutorial

Write a class `Subject` that represents a subject's information as it's name and score. Augment your class with a default constructor as well as one that takes two parameters to initialize the corresponding instance variables.

Write a class `Student` that represents a student at the GUC. Each student record object is comprised of four instance variables representing the first name, last name, section number and an array of subjects.

Write a constructor that takes four parameters to initialize the corresponding instance variables.

Augment your class with the following:

a) A method to change a certain score for a certain subject (passing the new score and the subject name).

b) A method to search for the score of a certain subject.

c) A method to calculate the GPA of the student and return it.

d) A method to get the highest score and return it.

e) A method to get the lowest score and return it.

f) A `toString()` method to display all information of a student.

g) A main method to test your program.

Solution:

```
public class Subject
{
        String name;
        double score;

        public Subject()
        {
                name="";
                score=0.0;
        }

        public Subject(String name, double score)
        {
                this.name=name;
                this.score=score;
        }
}


import java.util.*;

public class Student
{
        String fName;
        String lName;
        int sec;
        Subject [] subjects;

        Student(String f,String l,int n,Subject []s)
        {
                fName    = f;
                lName    = l;
                sec      = n;
                subjects = new Subject[s.length];
                for(int i = 0; i < subjects.length; i++)
                        subjects[i] = new Subject(s[i].name,s[i].score);
        }

        public void changeScore(double newScore,String subjectName)
        {
                for(int i=0; i<subjects.length;i++)
                {
                        if(subjects[i].name.equals(subjectName))
                                subjects[i].score = newScore;
                }
        }

        public double searchScore(String subjectName)
        {
                for(int i=0; i<subjects.length;i++)
                        if(subjects[i].name == subjectName)
                                return subjects[i].score;

                return -1;
        }

        public double calculateGPA()
        {
                double sum = 0, GPA;
                for(int i = 0; i < subjects.length; i++)
                        sum += subjects[i].score;

                GPA = sum / subjects.length;
                return GPA;
        }
        public double getMax()
        {
                double max = subjects[0].score;
                for(int i = 1; i < subjects.length; i++)
                        if(subjects[i].score > max)
                                max = subjects[i].score;
                return max;
        }
        public double getMin()
        {
                double min = subjects[0].score;
                for(int i = 1; i < subjects.length; i++)
                        if(subjects[i].score < min)
                                min = subjects[i].score;
                return min;
        }

        public String toString()
        {
                String s = "Name:␣" + fName + "␣" + lName +
                                "\nIn␣Section:␣" + sec + "\nHis␣grades:\n";
                for(int i = 0; i < subjects.length; i++)
                        s = s + "Subject␣" + subjects[i].name +
                            "␣:␣" + subjects[i].score + "\n";
                return s;
        }
```

```
    public static void main(String[] args)
    {
        Subject [] s = new Subject[3];
        Scanner sc  = new Scanner(System.in);

        for(int i=0;i<3;i++)
            s[i]=new Subject();

        System.out.println(
            "Please enter the name and grades for the 3 subjects:");

        for(int i=0;i<3;i++)
        {
            s[i].name=sc.next();
            s[i].score=sc.nextDouble();
        }

        Student a = new Student("Ali", "Mansour", 15, s);
        System.out.println(a);
        a.changeScore(100,"CSEN");
        System.out.println(a);
        System.out.println("His highest score: " + a.getMax());
        System.out.println("His lowest score: " + a.getMin());
        System.out.println("His GPA: " + a.calculateGPA());
    }
}
```

Exercise 10-4 Polynomial

You have to design a class `Polynomial` that represents polynomials with integer coefficients. A polynomial can be represented as a list of coefficients. For example,

- a polynomial of the form $4x^2 + 10x - 7$ can be represented as `{4,10,-7}`
- a polynomial of the form $3x^4 + 10x^2$ can be represented as `{3,0,10,0,0}`

a) Define a class `Polynomial` with the attributes defined above.

b) Augment your class with two constructors. One constructor is without parameters. The second one is a constructor that takes as parameter an array of coefficients and copies its the elements into the instance variable.

c) Augment your class with an instance method `degree` that returns the power of the highest non-zero term.

d) Augment your class with two methods for addition in both static and instance forms. The addition method should return a new polynomial. For simplicity, assume that both polynomials have the same degree.

e) Augment your class with a `toString()` method that returns a string representation of the polynomial (usex as the dummy variable). The method`toString` takes no parameters and returns a String representation of the polynomial. For example, you might return something like `"4x^2 + 10x^1 + -7"` as the result for $4x^2 + 10x - 7$. **Note**: If you want to be more clever about `toString`, you can watch the signs and omit terms with a zero coefficient.

f) Augment your class with a main method that constructs at least two polynomials and tests all methods defined above.

Solution:

```
public class Polynomial
{
        int [] coeff;
        public Polynomial(int[] coeff)
        {
                this.coeff = new int[coeff.length];
                for(int i = 0; i<this.coeff.length; i++)
                        this.coeff[i] = coeff[i];
        }
        public int degree()
        {
                for(int i = 0; i < coeff.length; i++)
                        if(coeff[i] != 0)
                                break;
                return coeff.length-i-1;
        }
        public Polynomial add(Polynomial p)
        {
                int[] result = new int [this.coeff.length];
                for(int i = 0; i < result.length; i++)
                        result[i] = this.coeff[i] + p.coeff[i];

                return new Polynomial(result);
        }
        public String toString()
        {
                String s = ""; int  j =coeff.length-1;
                for(int i = 0; i<coeff.length; i++)
                {
                        if(i == coeff.length-1)
                        {
                                if(coeff[i] > 0)
                                        s = s + "+␣"+ coeff[i];
                                else if(coeff[i] < 0)
                                        s = s + "-␣" + (-1 * coeff[i]);
                        }
                        else if( i == 0)
                        {
                                if(coeff[i] > 0)
                                        s = s + coeff[i] + "x^" + j+ "␣";
                                else if(coeff[i] < 0)
                                        s = s + coeff[i] + "x^" + j+ "␣";
                        }
                        else
                        {
                                if(coeff[i] > 0)
                                        s = s + "+␣" + coeff[i] + "x^" + j + "␣";
                                else if(coeff[i] < 0)
                                        s = s + "-␣" + (-1 * coeff[i])
                                            + "x^" + j + "␣";
                        }
                        j--;
                }
                return s;
        }
        public static void main(String[] args)
        {
                int[] c = {7,0,-2,7};
                Polynomial p = new Polynomial(c);
                int[] c1 = {2,-2,0,7};
                Polynomial p1 = new Polynomial(c1);
                System.out.println(p);
                System.out.println(p.add(p1));
        }
}
```

Exercise 10-5 Vacation Days
Final Exam 2016

A date is defined as a particularly day, month, and year.

a) Implement a class `Date` to define a date. Assume that an object of class `Date` has the following attributes:

- a day (where: 1 stands for Sunday, 7 stands for Saturday)
- a month (where: 1 stands for January, 12 stands for December)
- a year

The class should have an additional class variable which is an array of integers representing the number of days in every month. Assume that the first element correspond to the number of days in January and the last element corresponds to the number of days in December.

```
{31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
```

The class `Date` should implement the following:

1. a constructor that takes three parameters as input
2. `public int getDay()`
   returns the date's day.
3. `public int getMonth()`
   returns the date's month.
4. `public int getYear()`
   returns the date's year.
5. `public String toString()`
   return a String of the form

   `DD/MM/YYYY`
6. `public void updateDate()`
   to add one day to the date. **Note:** You should handle all cases when the date is updated.

Solution:

```
public class Date {

int day;
int month;
int year;
static int[] monthDays = {31,28,31,30,31,31,30,31,30,31,30,31};

public Date(int day, int month, int year) {
     this.day = day;
     this.month = month;
     this.year = year;
}

public int getDay() {
     return day;
}
public int getMonth() {
     return month;
}
public int getYear() {
     return year;
}

public String toString()
{
     String s = "";
     if(day<10)
     s+="0"+day+"/";
     else
     s+=day+"/";
     if(month<10)
     s+="0"+month+"/";
     else
     s+=month+"/";
     s+=year;
     return s;
}

public void updateDate()
{
     if(day==monthDays[month-1])
     {
     day=1;
     if(month==12)
     {
     month = 1;
     year++;
     }
     else
     month++;
     }
     else
     day++;
}
}
```

b) **Hint: Read all the question then start to solve:**

Assume that a user would like to keep track of his/her vacation days. Implement a class called `VacationDays` that has two attributes

- `dateList` which is an array of `Date` objects
- `numEntries` which corresponds to the number of `Date` instances that are actually in the list (the remaning slots will be `null`).

The class `VacationDays` should implement the following:

1. a constructor that takes an array `d` of dates as input and initializes the corresponding instance variables. The array instance variable should be of the same length as the array `d`. Assume that the first `numEntries` of `d` are `Date` instances and the remaning elements are null. You are required to do deep cloning in the constructor. Do not copy references of the objects. **Hint:** You are required to determine the value of `numEntries`.
2. Implement a boolean method `inOrder()` which determines if the elements of `dateList` are in order.
3. Implement a `toString` method that displays the dates in a specific format. For example, if the list consists of two dates `new Date(11,5,2016)` and `new Date(14, 7, 2016)`:

   ```
   You have vacation days in:
   day 11 in month 5 in year 2016
   day 14 in month 7 in year 2016
   ```
4. Implement a `main` method to test your classes.
   - Create an array `dates` of `Date` objects of size 10. The list should consist of at least three `Date` instances.
   - Create a `VacationDays` instance using the array `dates` and the corresponding constructor.
   - Check whether the dates are in order.
   - Display the information of the `VacationDays` object.
   - Update the date that appears in the first position of the array of days of the `VacationDays` object and display only this date using the corresponding methods from Part a).

Solution:

```
public class VacationDays {

Date []dateList;
int numEntries;

public VacationDays(Date []d)
{
     dateList = new Date[d.length];
     for(int i=0;i<d.length&&d[i]!=null;i++)
     {
     dateList[i] = new
     Date(d[i].getDay(),d[i].getMonth(),d[i].getYear());
     numEntries++;
     }
}

public boolean inOrder()
{
     for(int i=0;i<dateList.length-1 && dateList[i]!=null &&
     dateList[i+1]!=null ;i++)
     {
     if(dateList[i].getYear()>dateList[i+1].getYear())
     return false;
     if(dateList[i].getMonth()>dateList[i+1].getMonth()&&
     dateList[i].getYear()>=dateList[i+1].getYear())
     return false;
     if(dateList[i].getDay()>dateList[i+1].getDay()&&
     dateList[i].getMonth()>=dateList[i+1].getMonth())
     return false;
     }
     return true;
}

public String toString()
{
     System.out .println("You have vacation days in:");
     for(int i=0;i<dateList.length&&dateList[i]!=null;i++)
     {
     System.out .println("day "+dateList[i].getDay()+
     " month"+dateList[i].getMonth()+
     " year "+dateList[i].getYear());
     }
     return "";
}

public static void main(String []args)
{
     Date[]d = new Date[3];
     d[0] = new Date(11,5,2003);
     d[1] = new Date(1,3,2006);
     d[2] = new Date(13,6,2004);
     VacationDays v = new VacationDays(d);
     System.out.println(v.inOrder());
     System.out.println(v);
```

```
    v.dateList[0].updateDate();
    System.out.println(v.dateList[0]);
}
}
```