Faculty of Exact Sciences examination Programming
Vrije Universiteit February 2nd, 2016 time: 2:45 hours

SOLUTIONS

Problem 1.

a)
```
4 - 6 - 6
3 - 6 - 6
4 - 6 - 6
4 - 6 - 6
4 - 8 - 8
3 - 8 - 8
3 - 8 - 8
2 - 8 - 8
```

b)
```
double calculate (double x, int numberOfTerms) {
    double result = 0.0,
           power = 1.0;
    int sign = 1,
        factor = 2,
        factorial = 1;

    for (int i = 1; i <= numberOfTerms; i++) {
        double term = sign * power / (factor * factorial);
        result += term;

        sign = -sign;
        factor += 1;
        power *= x;
        factorial *= (2*i-1) * (2*i);
    }

    return result;
}
```

c)
```
static final int NUMBER_OF_ROWS = 4,
                 NUMBER_OF_COLUMNS = 8;
char[][] matrix = new char [NUMBER_OF_ROWS][NUMBER_OF_COLUMNS];

boolean oddBigger (int[][] m) {
    int sumEvenNumbers = 0,
        sumOddNumbers = 0;
    for (int i = 0; i < m.length; i++) {
        for (int j = 0; j < m[0].length; j++) {
            if (m[i][j] % 2 == 0) {
                sumEvenNumbers += m[i][j];
            } else {
                sumOddNumbers += m[i][j];
            }
        }
    }
    return sumOddNumbers > sumEvenNumbers;
}
```

d)
```
-3 4
-6 -2
-3 5
5 5
6 11
6 5
-2 8
```

Problem 2.

a)
```
class Library {
    static final int MAX_NUMBER_OF_BOOKCASES = 50;

    Bookcase[] bookcaseArray;
    int numberOfBookcases;

    Library () {
        bookcaseArray = new Bookcase[MAX_NUMBER_OF_BOOKCASES];
        numberOfBookcases = 0;
    }

    void add (Bookcase bookcase) {
        bookcaseArray[numberOfBookcases] = bookcase;
        numberOfBookcases += 1;
    }
}
```

b) Add to the class Bookcase.

```
static final int POPULAR_LIMIT = 100;

int numberOfLentOutBooks () {
    int result = 0;
    for (int i=0; i<bookArray.length; i++) {
        if (bookArray[i].lentOut) {
            result += 1;
        }
    }
    return result;
}

boolean isPopularGenre () {
    return numberOfLentOutBooks() > POPULAR_LIMIT;
}
```

Add to the class Library.

```
Library popularGenres () {
    Library result = new Library();

    for (int i=0; i<numberOfBookcases; i++) {
        if (bookcaseArray[i].isPopularGenre()) {
            result.add(bookcaseArray[i]);
        }
    }

    return result;
}
```

c) Add to the class Book.

```
static final int BIG_BOOK_PAGE_LIMIT = 500;

boolean isBigBook () {
    return numberOfPages > BIG_BOOK_PAGE_LIMIT;
}
```

Add to the class Bookcase.

```
static final double BIG_BOOK_PERCENTAGE = 22.5;

double percentageBigBooks () {
    int numberOfBigBooks = 0;
    for (int i=0; i<bookArray.length; i++) {
        if (bookArray[i].isBigBook()) {
            numberOfBigBooks += 1
        }
    }
    return 100 * (double)numberOfBigBooks/bookArray.length;
}

boolean isBigBookGenre () {
    return percentageBigBooks() >= BIG_BOOK_PERCENTAGE;
}
```

Add to the class Library.

```
int bigBookGenres () {
    int result = 0;

    for (int i=0; i<numberOfBookcases; i++) {
        if (bookcaseArray[i].isBigBookGenre()) {
            result += 1;
        }
    }

    return result;
}
```

d) Add to the class Library.

```
int bigPopularBookGenres () {
    return popularGenres().bigBookGenres()
}
```

Problem 3.

a)
```
String omit (String s) {
    if (s.length() <= 1) {
        return s;
    }

    char firstChar = s.charAt(0);
    String rest = s.substring(2);
    return firstChar + omit(rest);
}
```

b)
```
String clean(String s) {
    if (s.length() <= 1) {
        return s;
    }

    char firstChar  = s.charAt(0),
         secondChar = s.charAt(1);
    String rest = s.substring(1);

    if (firstChar == secondChar) {
        firstChar = '-';
    }
    return firstChar + clean(rest);

}
```