Faculty of Exact Sciences examination Programming
Vrije Universiteit October 20th, 2014 time: 2:45 hours

SOLUTIONS

Problem 1.

a)
```
A-Z-A
B-Z-A
B-Y-A
B-X-A
A-X-A
A-Z-A
B-X-B
```

b)
```
double eToThePower (double x, int numberOfTerms) {
    int factorial = 1;
    double power = 1.,
           term = power / factorial,
           result = term;

    for (int i = 1; i < numberOfTerms; i++) {
        factorial *= i;
        power *= x,
        term = power / factorial;
        result += term;
    }

    return result;
}
```

c)
```
static final int NUMBER_OF_ROWS    = 7,
                 NUMBER_OF_COLUMNS = 4;
char[][] matrix = new char[NUMBER_OF_ROWS][NUMBER_OF_COLUMNS];

boolean special (int[][] m) {
    for (int i = 0; i < m.length; i++) }
        for (int j = 0; j < m[0].length; j++) {
            if (m[i][j] != (j+1)*(i+1)) {
                return false;
            }
        }
    }

    return true;
}
```

d)
```
7, 11
16, 11
7, 0
6, 5
8, 6
```

Problem 2.

a)
```
class PotatoDistributionCentre {
    static final int MAX_NUMBER_OF_ELEMENTS = 200;
    PileOfPotatoes[] potatoArray;
    Int numberOfElements;

    PotatoDistributionCentre () {
        potatoArray = new PileOfPotatoes[MAX_NUMBER_OF_ELEMENTS];
        numberOfElements = 0;
    }

    void add (PileofPotatoes p) {
        potatoArray[numberOfElements] = p;
        numberOfElements += 1;
    }
}
```

b) Add to the class PileOfPotatoes

```
static final double FLOURY_BORDER = 19.5; // percent

boolean isFloury () {
      return starchContent > FLOURY_BORDER;
}
```

Add to the class PotatoDistributionCentre

```
PotatoDistributionCentre floury () {
    PotatoDistributionCentre result = new PotatoDistributionCentre();

    for (int i = 0; i < numberOfElements; i++) {
        if (potatoArray[i].isFloury()) {
            result.add(potatoArray[i]);
        }
    }

    return result;
}
```

c)
```
PotatoDistributionCentre flourySpecies (String s) {
    return floury().species(s);
}
```

d) Add to the class PileOfPotatoes

```
static final int UNSELLABLE_BORDER = 24; // months

boolean isUnsellable () {
      return monthsInStorage > UNSELLABLE_BORDER;
}
```

Add to the class PotatoDistributionCentre

```
void cleanup () {
    for (int i = 0; i < numberOfElements; i++) {
        if (potatoArray[i].isUnsellable()) {
            numberOfElements -= 1;
            potatoArray[i] = potatoArray[numberOfElements];
            i -= 1;
        }
    }
}
```

Problem 3.

a)
```
void dashes (String s) {
    if (s.length() < 2) {
        return s;
    }

    return s.charAt(0) + "-" + dashes(s.substring(1));
}
```

b) This can best be done with back tracking

```
void permutations (char[] r, int length) {
    if (length < 2) {
        println(r);
        return;
    }

    for (int i = 0; i < length; i++) {
        swap(r, i, length-1);
        permutations(r, length-1);
        swap(r, i, length-1);
    }
}
```