CS101- Algorithms and Programming I

Lab 10

Lab Objectives: Arrays

- ❏ For all labs in CS 101, your solutions must conform to these CS101 style guidelines (rules!)
- ❏ Remember to include javadoc comments for each class and method.
- ❏ Create a Lab10 workspace (i.e. the folder H:\private\cs101\lab10).

Part a:

Create a new project and name it Lab10a. This part tests your ability to define, fill and use a fixed sized array of primitive types. Therefore, you should try to make a link between what this question asks and what you have already learnt about arrays. In this question, you are asked to write a Java program that given necessary parameters calculates the value of the generic polynomial expression shown below:

$$R = (a_n)x^n + (a_{n-1})x^{n-1} + \ldots + a_1x^1 + a_0$$

In each term of the above mentioned expression, “x” is called the variable and “a” is called coefficient. Your program must prompt the user for the value of n and then ask the values of the coefficients accordingly (the number of times that the program asks for the value of the “a” is equal to n + 1). Another parameter that is necessary to calculate the result of the expression is “x”. This value must be asked at the end. Your program will then print the specified expression together with the result in the format given below:

an(x ^ n) + an-1(x ^ (n - 1)) + … + a1x1 + a0 = R

For example for n = 3, a_3 = 6, a_2 = 1, a_1 = 3, a_0 = 5 and x = 2, your program must show the final result as:

6(2 ^ 3) + 1(2 ^ 2) + 3(2 ^ 1) + 5 = 63

Your program must iterate and ask different values of x and print the results until user enters 0.

The program must have a method that takes the array of coefficients and the value of x and then it returns the value of R. Name your method calculatePolynomial.
Write another method that takes the array of coefficients and values of x and R as parameters and prints the expression in the defined format. Name your method printPolynomial.

You can define all the values as integers.

SAMPLE OUTPUT (user input is shown in red):

```
Start of Lab10a

Please enter the value of N: 3
Please enter the coefficient a3: 8
Please enter the coefficient a2: 6
Please enter the coefficient a1: 2
Please enter the coefficient a0: 5

Please enter the X value: 2
************************
for X = 2
8(2 ^ 3) + 6(2 ^ 2) + 2(2 ^ 1) + 5 =  97

Please enter the X value: 4
************************
for X = 4
8(4 ^ 3) + 6(4 ^ 2) + 2(4 ^ 1) + 5 =  621

Please enter the X value:  9
************************
for X = 9
8(9 ^ 3) + 6(9 ^ 2) + 2(9 ^ 1) + 5 =  6341

Please enter the X value:  0
************************
--- End of Lab10a ---
```

Part b: Create a new project and name it Lab10b. This project will contain three classes named House, Site, and SiteApplication. The latter one will contain the main method.

1. Write a class, **House.java** that has the following data and methods.
   Data Members:
   - **String ownerName** – the name of the registered owner.
   - **String address** – house address (block and unit).
   - **int residents** – number of residents living in the unit.
   - **double serviceChargePerResident**– monthly charge per resident.

   Methods:
   - Constructor that initializes all data members to values passed as parameters.
   - *Getters* and *Setters* for **all** data members.
   - **calculateTotalServiceFee ()** – calculates and returns the total monthly fee for all residents.

   toString() – returns a String representation of a House, formatted as in the sample shown below.

   Owner: Owner Name

   Address

   Residents: number of residents Monthly Fee: (total calculated service fee TL)

2. Write a class, **Site.java**, that has the following data and methods:
   Data Members:
   - **MAX_HOUSES:** stores the constant maximum number of houses allowed in the site (10)
   - **houseList:** an ***array*** of Houses on the Site
   - **houseCount:** stores the number of houses in the Site.
   - **siteName:** full name of the Site.
   - **residents:** total number of residents living on the site.

   Methods:
   - **Constructor** that initializes the Site name to value passed as parameters and initializes the houseList.
   - **addHouse():** takes the information about a house (owner name, address, residents, service fee per person), and if the Site is not full, adds the House to the list and updates the number of residents each time a house is added, using the appropriate method. The method should return a boolean value, indicating whether or not the house was successfully added to the list.
   - **updateTotalResidents ():** calculates the total residents in the Site and sets the residents data member using the calculated value.
   - **viewHouses():** displays all Houses on the Site, see sample run.
   - **searchHouseByFee ():** takes a min and max service charge as a parameter and returns an ***ArrayList*** containing all houses whose total service fee is between the given values.

3. Write an application class, **SiteApplication.java** that does the following:
    - Creates a new Site, using name input from the user.
    - Implements the menu functionality shown in the sample run below.

Sample output (user input is shown in red):

```
Enter name of site: Jasmin Garden Apartments
1-Add House
2-View Houses
3-Search Houses by Fee
4-Exit
Enter choice: 1

Enter Owner Name: Alona Akar
Enter Address: B-Block Apartment 502
Enter number of residents: 3
Enter service charge: 175
House Successfully added to Site

1-Add House
2-View Houses
3-Search Houses by Fee
4-Exit
Enter choice: 1

Enter Owner Name: Melinda Watts
Enter Address: A-Block Apartment 316
Enter number of residents: 2
Enter service charge: 125
House Successfully added to Site

1-Add House
2-View Houses
3-Search Houses by Fee
4-Exit
Enter choice: 1

Enter Owner Name: Suleyman Zengin
Enter Address: B-Block Apartment 418
Enter number of residents: 5
Enter service charge: 100
House Successfully added to Site

1-Add House
```

```
2-View Houses
3-Search Houses by Fee
4-Exit
Enter choice: 1

Enter Owner Name: Mert Yayla
Enter Address: C-Block Apartment 210
Enter number of residents: 2
Enter service charge: 225
House Successfully added to Site

1-Add House
2-View Houses
3-Search Houses by Fee
4-Exit
Enter choice: 1

Enter Owner Name: Nil Karahan
Enter Address: B-Block Apartment 101
Enter number of residents: 2
Enter service charge: 180
House Successfully added to Site

1-Add House
2-View Houses
3-Search Houses by Fee
4-Exit
Enter choice: 5
Invalid Choice

1-Add House
2-View Houses
3-Search Houses by Fee
4-Exit
Enter choice: 2

List of Houses in Jasmin Garden Apartments
------------------------

Owner: Alona Akar
B-Block Apartment 502
Residents: 3  Monthly Fee: (525.0 TL)


Owner: Melinda Watts
```

```
A-Block Apartment 316
Residents: 2  Monthly Fee: (250.0 TL)

Owner: Suleyman Zengin
B-Block Apartment 418
Residents: 5  Monthly Fee: (500.0 TL)

Owner: Mert Yayla
C-Block Apartment 210
Residents: 2  Monthly Fee: (450.0 TL)

Owner: Nil Karahan
B-Block Apartment 101
Residents: 2  Monthly Fee: (360.0 TL)
Total Residents on Site: 14.0

1-Add House
2-View Houses
3-Search Houses by Fee
4-Exit
Enter choice: 3

Enter minimum and maximum fee: 300 400

Houses on Site with fee between 300.00 and 400.00 TL:
[Owner: Nil Karahan
B-Block Apartment 101
Residents: 2  Monthly Fee: (360.0 TL)]

1-Add House
2-View Houses
3-Search Houses by Fee
4-Exit
Enter choice: 4
--- End of Lab10b ---
```