Project 2: Computing Future Investment Value

Active Members: Nevin Ndonwi   Orion Strinni   Adam Johnson

Problem Description:

Create a method that computes future investment value at a given interest rate for a specified number of years. The future investment is determined using the following formula:

$$\texttt{futureInvestmentValue =}$$
$$\texttt{investmentAmount x (1 + monthlyInterestRate)}^{\texttt{numberOfYears*12}}$$

Use the following method header:

```
public static double futureInvestmentValue(
  double investmentAmount, double monthlyInterestRate, int years)
```

For example, futureInvestmentValue(10000, 0.05/12, 5) returns 12833.59.

Create a test program that prompts the user to enter the investment amount (e.g., 1000) and the interest rate (e.g., 9%) and prints a table that displays future value for the years from 1 to 30, as shown below:

```
The amount invested: 1000
Annual interest rate: 9%
Years               Future Value
1                    1093.80
2                    1196.41
...
29                  13467.25
30                  14730.57
```

Create appropriate validation method(s) to validate user input for the investment amount and the interest rate. You may use the validation method I posted in hands-on labs folder and modify the method to validate a double instead of an int.

Analysis:
(Describe the problem including input and output in your own words.)

The user needs to input the initial value and the interest rate and the program needs to output what the future value will be in years for the next 30 years.

Design:
(Describe the major steps for solving the problem.)

Step 1: Gain the interest rate and the principal investment
Step 2: Check if the interest rate and principal investment are numbers. (go back to step 1 if this is not the case)
Step 3: Calculate the future value for the next 30 years and output this information to the user

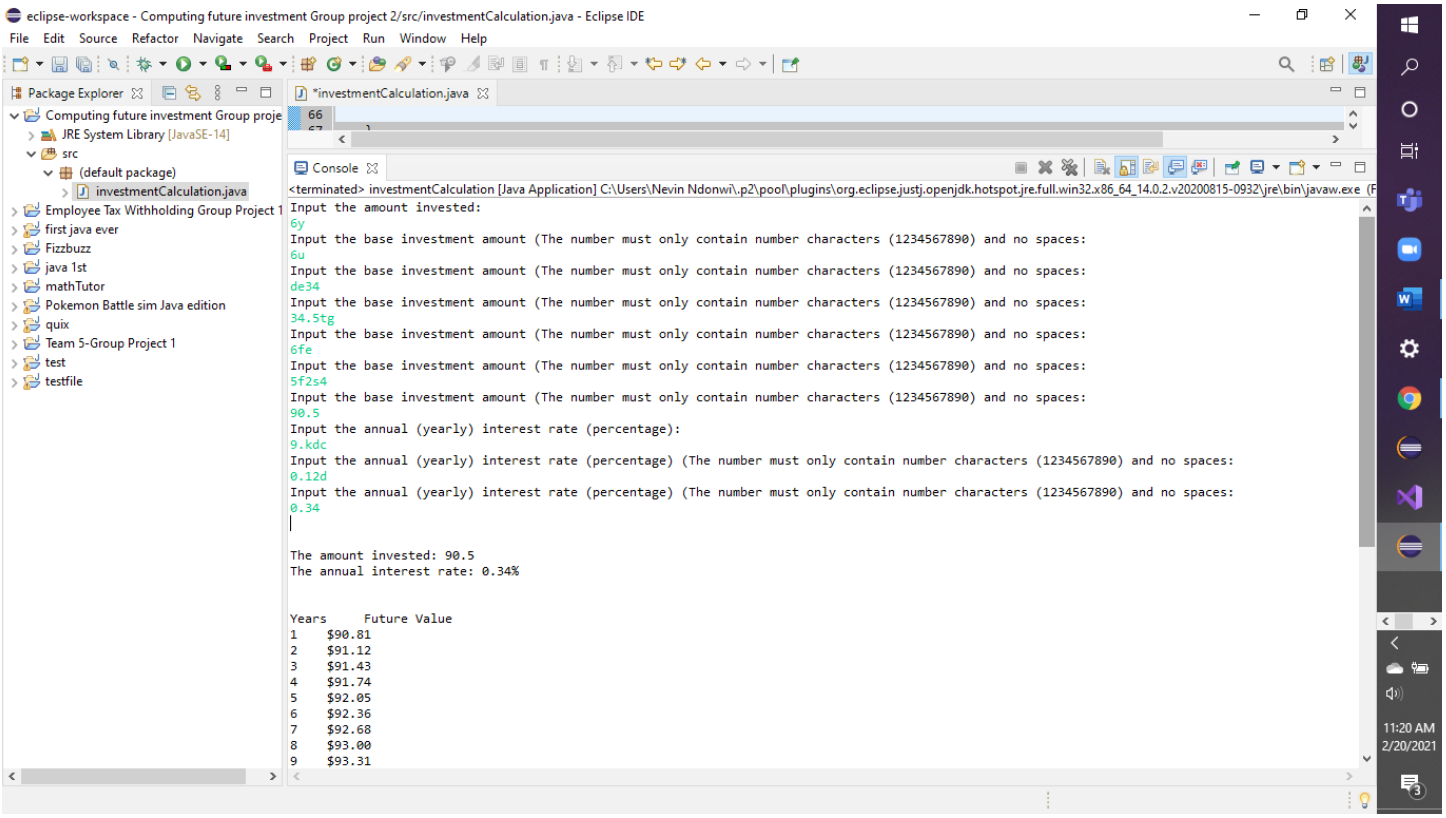
End program.

Coding: (Copy and Paste Source Code here. Format your code using `Courier New 10pts`)
(This project was programmed utilizing eclipse)

```
import java.util.*;


// not done yet
public class investmentCalculation {

      static Scanner scan = new Scanner(System.in); // creates  a scanner object


        public static boolean numberLegitimizer(String test) {


            boolean legit = false;
            String numbers = "1234567890.";

            int counter = 0;

            for (int i = 1; i <= test.length(); i++) { // Checks if a string only contains
number characters 1234567890.


                  for (int o = 1; o <= numbers.length(); o++) {

                    if(test.substring(i - 1,i).equals(numbers.substring(o - 1, o)) ) {

                        counter += 1;
                        break;
                     }





                  }

            }


            if (counter >= test.length() ) {
                  return !legit;  // Returns true:  If the number of number characters is
greater than or equal to the length of the string
            }
            else {

            return legit; // Returns false: if the amount of number characters is less than
the length of the string

            }



      }


      public static double futureInvestmentValue(double investmentAmount, double
monthlyInterestRate, int years) { // actual method that computes the investment



            monthlyInterestRate = monthlyInterestRate/1200; // finds the monthly interest
rate

        double projected  =  investmentAmount * Math.pow(1 + monthlyInterestRate, years *
12); // equation for investment (one has to use exponents)




            return projected;


      }

      public static void main( String[] args) {

            System.out.println("Input the amount invested: ");
            String amounttext = scan.nextLine();

        while(numberLegitimizer(amounttext) != true) { // keeps repeating until the user
enters a number with only number characters 1234567890

                  System.out.println("Input the base investment amount (The number must
only contain number characters (1234567890) and no spaces: ");
                   amounttext = scan.nextLine();



            }



            System.out.println("Input the annual (yearly) interest rate (percentage): ");
            String ratetext = scan.nextLine();

         while(numberLegitimizer(ratetext) != true) { // keeps repeating until the user
enters a number with only number characters 1234567890

                  System.out.println("Input the annual (yearly) interest rate (percentage)
(The number must only contain number characters (1234567890) and no spaces: ");
                   ratetext = scan.nextLine();
            }

            double amount = Double.parseDouble(amounttext); // this converts the string to
double

            double rate = Double.parseDouble(ratetext); // this converts the string to
double

          int years = 1;

          double output = 0;

            System.out.println("  ");
            System.out.println("  ");

            System.out.println("The amount invested: " + amount );
            System.out.println("The annual interest rate: " + rate + "%");

            System.out.println("  ");
            System.out.println("  ");

            System.out.println("Years " + "    " + "Future Value");



            for (int p = 0; p < 30; p++) {

            output = futureInvestmentValue(amount,  rate ,years);

          System.out.printf(p + 1 + "    " + "$%4.2f" + " \n", output);


             // You have to reset the amount to its origonal value to find the value

            years++;


            }




      }
}
```

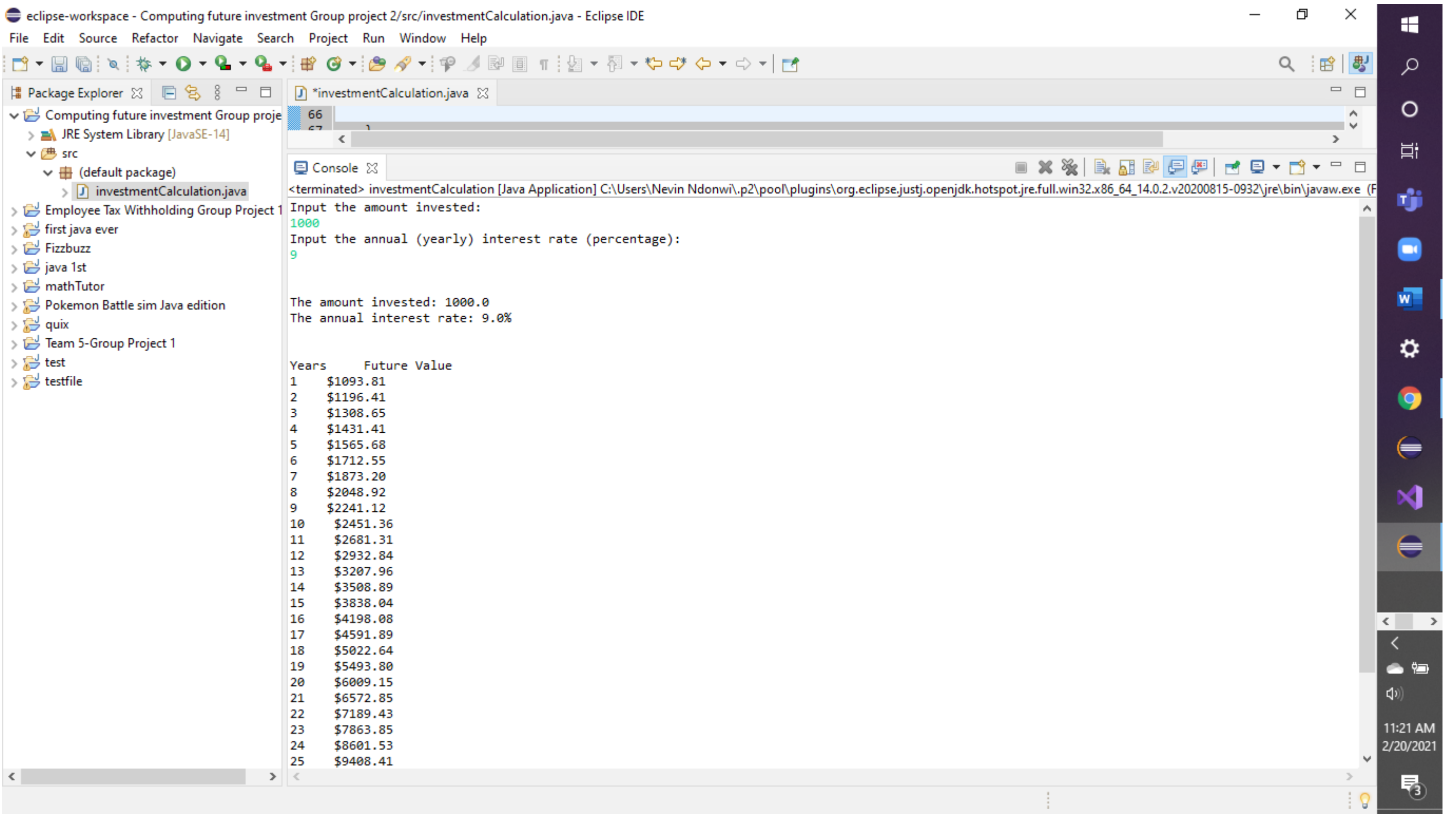
Testing: (Describe how you test this program including screen captures)

Testing if the code accepts non-number characters:
(It doesn’t) (A while loop is employed so that the user can re-input the variable)

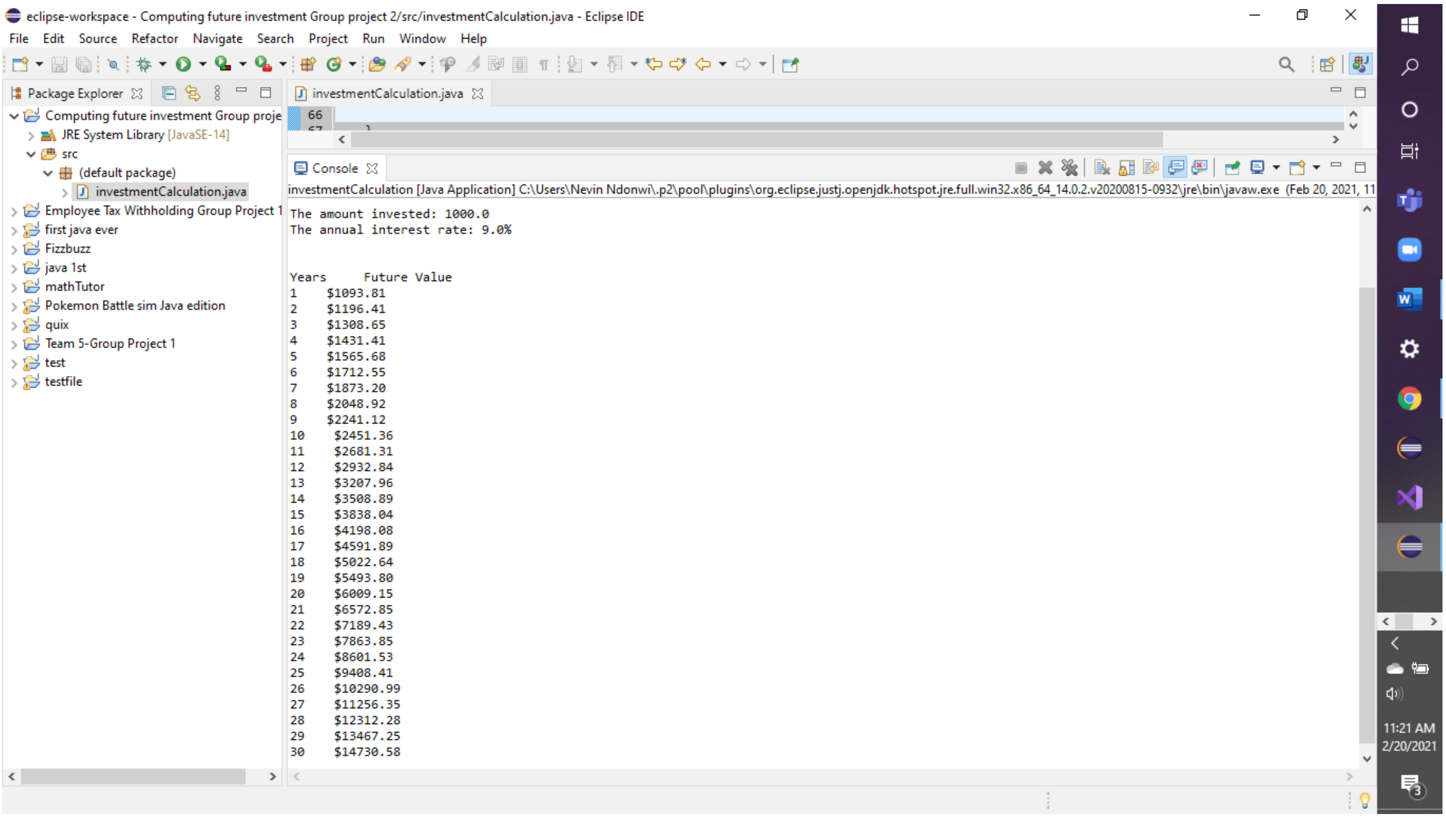

eclipse-workspace - Computing future investment Group project 2/src/investmentCalculation.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Computing future investment Group proje
JRE System Library [JavaSE-14]
src
(default package)
investmentCalculation.java
Employee Tax Withholding Group Project 1
first java ever
Fizzbuzz
java 1st
mathTutor
Pokemon Battle sim Java edition
quix
Team 5-Group Project 1
test
testfile
*investmentCalculation.java
Console
11:20 AM
2/20/2021

```
<terminated> investmentCalculation [Java Application] C:\Users\Nevin Ndonwi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_14.0.2.v20200815-0932\jre\bin\javaw.exe (F
Input the amount invested:
6y
Input the base investment amount (The number must only contain number characters (1234567890) and no spaces:
6u
Input the base investment amount (The number must only contain number characters (1234567890) and no spaces:
de34
Input the base investment amount (The number must only contain number characters (1234567890) and no spaces:
34.5tg
Input the base investment amount (The number must only contain number characters (1234567890) and no spaces:
6fe
Input the base investment amount (The number must only contain number characters (1234567890) and no spaces:
5f2s4
Input the base investment amount (The number must only contain number characters (1234567890) and no spaces:
90.5
Input the annual (yearly) interest rate (percentage):
9.kdc
Input the annual (yearly) interest rate (percentage) (The number must only contain number characters (1234567890) and no spaces:
0.12d
Input the annual (yearly) interest rate (percentage) (The number must only contain number characters (1234567890) and no spaces:
0.34


The amount invested: 90.5
The annual interest rate: 0.34%


Years     Future Value
1    $90.81
2    $91.12
3    $91.43
4    $91.74
5    $92.05
6    $92.36
7    $92.68
8    $93.00
9    $93.31
```

Testing the code for the correct monetary value within the nearest penny: (uncertainty of +- 0.01)
(It does)


eclipse-workspace - Computing future investment Group project 2/src/investmentCalculation.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Computing future investment Group proje
JRE System Library [JavaSE-14]
src
(default package)
investmentCalculation.java
Employee Tax Withholding Group Project 1
first java ever
Fizzbuzz
java 1st
mathTutor
Pokemon Battle sim Java edition
quix
Team 5-Group Project 1
test
testfile
*investmentCalculation.java
Console
11:21 AM
2/20/2021

```
<terminated> investmentCalculation [Java Application] C:\Users\Nevin Ndonwi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_14.0.2.v20200815-0932\jre\bin\javaw.exe (F
Input the amount invested:
1000
Input the annual (yearly) interest rate (percentage):
9


The amount invested: 1000.0
The annual interest rate: 9.0%


Years    Future Value
1    $1093.81
2    $1196.41
3    $1308.65
4    $1431.41
5    $1565.68
6    $1712.55
7    $1873.20
8    $2048.92
9    $2241.12
10    $2451.36
11    $2681.31
12    $2932.84
13    $3207.96
14    $3508.89
15    $3838.04
16    $4198.08
17    $4591.89
18    $5022.64
19    $5493.80
20    $6009.15
21    $6572.85
22    $7189.43
23    $7863.85
24    $8601.53
25    $9408.41
```


eclipse-workspace - Computing future investment Group project 2/src/investmentCalculation.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Computing future investment Group proje
JRE System Library [JavaSE-14]
src
(default package)
investmentCalculation.java
Employee Tax Withholding Group Project 1
first java ever
Fizzbuzz
java 1st
mathTutor
Pokemon Battle sim Java edition
quix
Team 5-Group Project 1
test
testfile
investmentCalculation.java
Console
11:21 AM
2/20/2021

```
investmentCalculation [Java Application] C:\Users\Nevin Ndonwi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_14.0.2.v20200815-0932\jre\bin\javaw.exe (Feb 20, 2021, 11
The amount invested: 1000.0
The annual interest rate: 9.0%


Years    Future Value
1    $1093.81
2    $1196.41
3    $1308.65
4    $1431.41
5    $1565.68
6    $1712.55
7    $1873.20
8    $2048.92
9    $2241.12
10    $2451.36
11    $2681.31
12    $2932.84
13    $3207.96
14    $3508.89
15    $3838.04
16    $4198.08
17    $4591.89
18    $5022.64
19    $5493.80
20    $6009.15
21    $6572.85
22    $7189.43
23    $7863.85
24    $8601.53
25    $9408.41
26    $10290.99
27    $11256.35
28    $12312.28
29    $13467.25
30    $14730.58
```

Submit the following items:

1. Submit this Word file – rename it project 2 solution file in Bb

2. Submit your Java project folder in compressed file format in Blackboard