Java Portfolio for my project “Number Analysis”.

Date: June 16, 2023

What is it: This is a program where you can type in an integer n, and after that it will do a bunch of analysis on the integer n and other numbers as well. The output shows different properties of each number from 2 to n, including showing that whether the number is even or odd, perfect number or deficient number, and etc.

Designing process:

I got to do this project again since I actually misunderstood the assignment at first and so I did a “GCD project” instead. This time, I designed different methods for the program to make it work more efficiently and allow me to debug more easily since I can simply fix and check if methods go wrong instead of checking the entire program.

Source code: numberAnalyze.java

import java.util.Scanner;

public class numberAnalyze\_1 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

boolean continueOrNot;

System.out.println("This is a program where you can type in an integer n, and after that it will do a bunch of analysis on the integer n and other numbers as well! Just wait and see.");

while(true) {

System.out.print("Enter an integer (n): ");

int n = scanner.nextInt();

scanner.nextLine();

if (n <= 0) {

System.out.println("Invalid input! Please enter a positive integer.");

continue;

}

System.out.println("Prime Factorization Results:");

for (int i = 2; i <= n; i++) {

int[] factors = primeFactors(i);

if (factors.length > 0) {

System.out.print(i + ": ");

for (int factor : factors) {

System.out.print(factor + " ");

}

System.out.println();

}

}

int sum = sumOfFactors(n);

System.out.println("Number Analysis of "+n+":");

System.out.println("Sum of factors: " + sum);

if (sum < n) {

System.out.println(n + " is a deficient number.");

} else if (sum == n) {

System.out.println(n + " is a perfect number.");

} else {

System.out.println(n + " is an abundant number.");

}

if (n % 2 == 0) {

System.out.println(n + " is an even number.");

} else {

System.out.println(n + " is an odd number.");

}

System.out.print("Type \"quit\" to quit the program, otherwise the program will go on.");

String quit=scanner.nextLine();

if(quit.equalsIgnoreCase("quit")){

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

continueOrNot=false;

System.exit(0);

}

}

}

private static int[] primeFactors(int n) {

int[] factors = new int[n];

int index = 0;

while (n % 2 == 0) {

factors[index++] = 2;

n /= 2;

}

for (int i = 3; i <= Math.sqrt(n); i += 2) {

while (n % i == 0) {

factors[index++] = i;

n /= i;

}

}

if (n > 2) {

factors[index++] = n;

}

// Trim the array to the actual number of factors

int[] trimmedFactors = new int[index];

System.arraycopy(factors, 0, trimmedFactors, 0, index);

return trimmedFactors;

}

private static int sumOfFactors(int n) {

int sum = 1;

for (int i = 2; i <= Math.sqrt(n); i++) {

if (n % i == 0) {

if (i == (n / i)) {

sum += i;

} else {

sum += (i + n / i);

}

}

}

return sum;

}

}//main

Debug:

3 errors

User@Osmonds-MacBook-Air desktop % javac numberAnalyze\_1.java

numberAnalyze\_1.java:53: error: cannot find symbol

if(quit.equalsIgnoreCase("quit")){

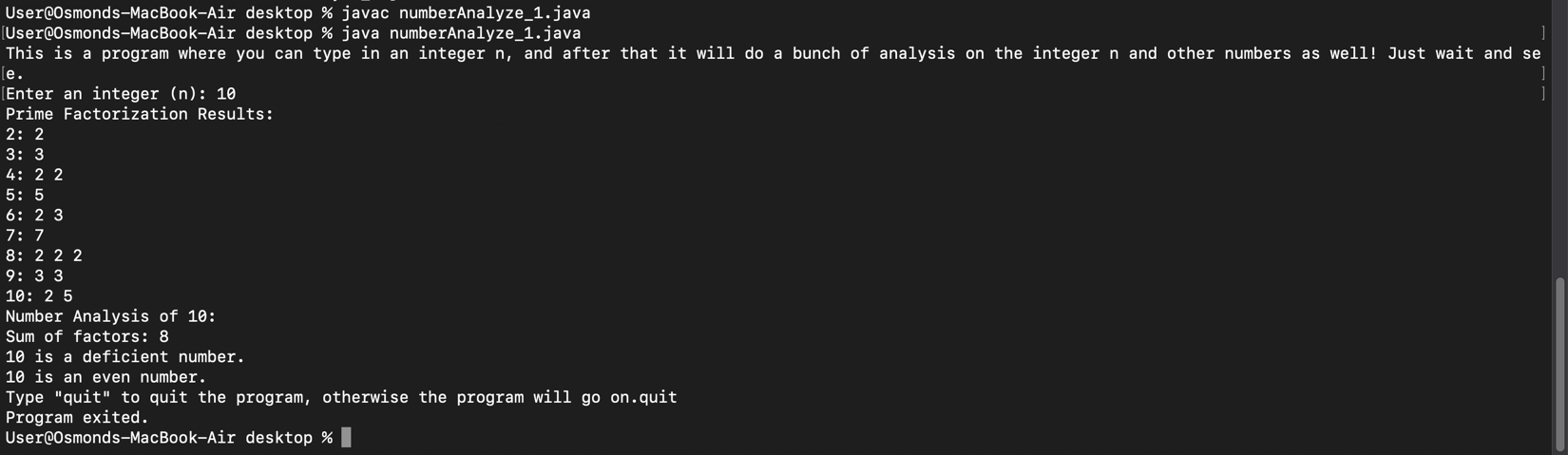
^

symbol: variable quit

location: class numberAnalyze\_1

I put the variable quit inside the loop so that the compiler assumed there wasn’t such a variable.

How it goes:



What I have learned:

1. Application of methods.
2. Module is an efficient tool to use when the program is complex and full of different variables.
3. Use Boolean data type to let users decide whether they want to continue the program or not.
4. Use different arrays to store different data.