Mod 45Project

Chris Schmidlin

Rasmussen College

Author Note

This paper is being submitted on Wednesday, July 31, 2019, for Anastasia Rashtchian’s COP2268C Java Programming course.

Mod 5 Project

import java.util.Scanner;

import java.io.\*;

import java.io.FileNotFoundException;

interface BMI{

public void getInfo();

public void calcBMI();

public void createFile();

public void consolePrint();

}

public class BMICalc implements BMI{

String file = "BMI.csv";

int BMI = 0;

Scanner in = new Scanner(System.in);

String name = "";

String birthdate = "";

int weight = 0;

int height = 0;

public void getInfo(){

System.out.print("Name: ");

name = in.nextLine();

System.out.print("Birthdate: ");

birthdate = in.nextLine();

System.out.print("Weight: ");

weight = in.nextInt();

System.out.print("Height: ");

height = in.nextInt();

}

public void calcBMI(){

try{

BMI = 703 \* weight / (height\*height);

} catch (Exception e){

BMI = -1;

System.out.println("The exception is " + e.getMessage());

System.out.println("The exception code is " + e.toString());

} finally {

if (BMI == -1){

System.out.println("Please start over.");

System.exit(0);

}

}

}

public void createFile(){

try{

FileWriter fw = new FileWriter(file, true);

//Create headers in the CSV file

fw.append("Name" + ",");

fw.append("Birthday" + ",");

fw.append("Weight" + ",");

fw.append("Height" + ",");

fw.append("BMI");

fw.append("\n");

fw.append("\n");

fw.flush();

} catch (Exception e){

System.out.println("The exception is " + e.getMessage());

System.out.println("The exception code is " + e.toString());

}

}

public void appendToFile(){

try {

FileWriter fw = new FileWriter(file, true);

//Input info

fw.append(name + ",");

fw.append(birthdate + ",");

fw.append(weight + ",");

fw.append(height + ",");

fw.append(BMI + ",");

if (BMI < 18.5){

fw.append("BMI Score is Underweight" + ",");

} else if (BMI >= 18.5 && BMI <= 24.9){

fw.append("BMI Score is Normal" + ",");

} else if (BMI >= 25 && BMI <= 29.9){

fw.append("BMI Score is Overweight" + ",");

} else {

fw.append("BMI Score is Obese" + ",");

}

if (BMI < 18.5){

fw.append("Insurance payment category is low.");

} else if (BMI >= 18.5 && BMI <= 24.9){

fw.append("Insurance payment category is low.");

} else if (BMI >= 25 && BMI <= 29.9){

fw.append("Insurance payment category is high.");

} else {

fw.append("Insurance payment category is highest.");

}

fw.append("\n");

fw.append("\n");

fw.flush();

//fw.close();

} catch (Exception e){

System.out.println("The exception is " + e.getMessage());

System.out.println("The exception code is " + e.toString());

}

}

public void consolePrint(){

//Printing out the user's info

System.out.println(name);

System.out.println("Birthdate: " + birthdate);

System.out.println("Weight: " + weight);

System.out.println("Height: " + height);

System.out.println("BMI: " + BMI);

//Printing out the user's BMI Score

if (BMI < 18.5){

System.out.println("BMI Score is Underweight");

} else if (BMI >= 18.5 && BMI <= 24.9){

System.out.println("BMI Score is Normal");

} else if (BMI >= 25 && BMI <= 29.9){

System.out.println("BMI Score is Overweight");

} else {

System.out.println("BMI Score is Obese");

}

//Printing out the user's insurance payment category based on the BMI

if (BMI < 18.5){

System.out.println("Insurance payment category is low.");

} else if (BMI >= 18.5 && BMI <= 24.9){

System.out.println("Insurance payment category is low.");

} else if (BMI >= 25 && BMI <= 29.9){

System.out.println("Insurance payment category is high.");

} else {

System.out.println("Insurance payment category is highest.");

}

}

public void runProgram(){

char token = 's';

createFile();

do {

 getInfo();

calcBMI();

consolePrint();

appendToFile();

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

token = in.next().charAt(0);

in.nextLine();

} while (token != 'q');

}

}

public class main {

public static void main(String[] args){

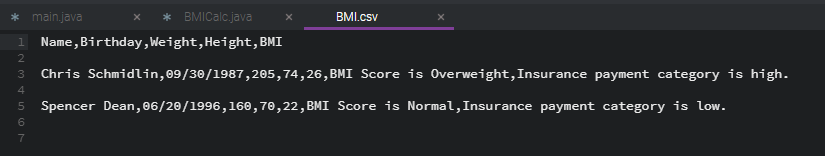
BMICalc patient = new BMICalc();

patient.runProgram();

}

}

^ Inputting the information into the console



^ The output