健康管理系統(sort&匯入匯出)1.0版

|  |
| --- |
| BMI |
| -Weight：double  -Height：double  -Birth：String  -Gender：boolean  -Genderstr： String  -KILOGRAMS\_PER\_POUND：final double = 0.45359237  -METERS\_PER\_INCH： final double = 0.0254 |
| +BMI()  +BMI ( double weight, double height, String birth, boolean gender, String genderstr)  +getWeight()：double  +setWeight(double weight)：void  +getHeight()：double  +setHeight(double height)：void  +getBirth()：String  +setBirth (String birth)：void  +getGender()：boolean  +getGenderstr()：String  +setGenderstr (String newGenderstr)：void  +getBMI()：double  +getStatus()：String  +getAge()：int  +getBodyFat()：double |

U10314010陳映羽

|  |
| --- |
| Account |
| -Name：String  -Address：String  -Telephone：String  -Test time：String |
| +Account()  +Account (String name, String address, String telephone, String testtime)  +getName()：String  +setName(String name)：void  +getAddress()：String  +setAddress(String address)：void  +getTelephone()：String  +setTelephone (String telephone)：void  +getTesttime()：String  +setTesttime (String testtime)：void |

|  |
| --- |
| Console |
| +getInt(String)：int  +getStrimg(String)：String  +getGender(String)：boolean  +getDouble(String)：double |

|  |
| --- |
| Bubble Sort |
| +bubbleSort(Account[]list,BMI[]list1)：void |

|  |
| --- |
| TestBMI |
| +main(args：String[ ])：void |

//Design a class named Account

public class Account{

//The name, gender, address, telephone and test time for the account

private String name;

private String address;

private String telephone;

private String testtime;

//Create a default account

public Account(){

}

//Create an account with the specified name, gender, address, telephone, birth and test time

public Account(String name, String address, String telephone, String testtime){

this.name = name;

this.address = address;

this.telephone = telephone;

this.testtime = testtime;

}

//Return name

public String getName(){

return name;

}

//Set name

public void setName(String name){

this.name = name;

}

//Return address

public String getAddress(){

return address;

}

//Set address

public void setAddress(String address){

this.address = address;

}

//Return telephone

public String getTelephone(){

return telephone;

}

//Set telephone

public void setTelephone(String telephone){

this.telephone = telephone;

}

//Return test time

public String getTestTime(){

return testtime;

}

//Set test time

public void setTestTime(String testtime){

this.testtime = testtime;

}

}

//Design a class named BMI

import java.util.Calendar;

public class BMI {

//The weight of the person in pounds

//The height of the person in inches

//The gender of the person,use word(m and f) to distinguish male and female

//The age of the person

private double weight; //in pounds

private double height; //in inches

private String birth;

private boolean gender;

private String genderstr;

public static final double KILOGRAMS\_PER\_POUND = 0.45359237;

public static final double METERS\_PER\_INCH = 0.0254;

//Create a default bmi

public BMI(){

}

//Create a BMI object with the specified age, weight and height,different BMI standard for male and female

public BMI(double weight, double height, String birth, boolean gender, String genderstr){

this.weight = weight;

this.height = height;

this.birth = birth;

this.genderstr = genderstr;

}

//Return weight

public double getWeight(){

return weight \* KILOGRAMS\_PER\_POUND;

}

//Return height

public double getHeight(){

return height \* METERS\_PER\_INCH;

}

//Set weight

public void setWeight(double weight){

this.weight = weight;

}

//Set height

public void setHeight(double height){

this.height = height;

}

//Return birth

public String getBirth(){

return birth;

}

//Set birth

public void setBirth(String birth){

this.birth = birth;

}

//Return gender

public boolean getGender(){

return gender;

}

//Return gender for array

public String getGenderstr(){

return genderstr;

}

//Set gender

public void setGenderstr(String newGenderstr){

genderstr = newGenderstr;

}

//Return BMI

public double getBMI(){

double bmi = weight \* KILOGRAMS\_PER\_POUND /

((height \* METERS\_PER\_INCH) \* (height \* METERS\_PER\_INCH));

return Math.round(bmi \* 100) / 100.0;

}

//Return status

//When gender=1 means male, gender(= 0) means female

public String getStatus(){

double bmi = getBMI();

if (gender){

if(bmi<19)

return"Underweight";

else if(bmi<=25)

return"Normal";

else if(bmi<=30)

return"Overweight";

else

return"Obese";

}

else{

if(bmi<18)

return"Underweight";

else if(bmi<=24)

return"Normal";

else if(bmi<=29)

return"Overweight";

else

return"Obese";

}

}

public int getAge(){

String strs[] = birth.trim().split("-");

int birthYear = Integer.parseInt(strs[0]);

int birthMonth = Integer.parseInt(strs[1]);

int birthDay = Integer.parseInt(strs[2]);

Calendar cal = Calendar.getInstance();

int yearNow = cal.get(Calendar.YEAR);

int monthNow = cal.get(Calendar.MONTH) + 1;

int dayNow = cal.get(Calendar.DATE);

int yearMinus = yearNow - birthYear;

int monthMinus = monthNow - birthMonth;

int dayMinus = dayNow - birthDay;

int age = yearMinus;

if (yearMinus < 0) {

age = 0;

}

else if (yearMinus == 0) {

if (monthMinus < 0) {

age = 0;

}

else if (monthMinus == 0) {

if (dayMinus < 0) {

age = 0;

}

else if (dayMinus >= 0) {

age = 1;

}

}

else if (monthMinus > 0) {

age = 1;

}

}

else if (yearMinus > 0) {

if (monthMinus < 0) {

}

else if (monthMinus == 0) {

if (dayMinus < 0) {

}

else if (dayMinus >= 0) {

age = age + 1;

}

}

else if (monthMinus > 0) {

age = age + 1;

}

}

return age;

}

//Return body fat

public double getBodyFat(){

double bmi = getBMI();

int age = getAge();

if (getGender()){

double bodyfat = (bmi \* 1.2) + (age \* 0.23) - 5.4 - (1.0 \* 10.8);

return Math.round(bodyfat \* 100) / 100.0;

}

else{

double bodyfat = (bmi \* 1.2) + (age \* 0.23) - 5.4 - (0.0 \* 10.8);

return Math.round(bodyfat \* 100) / 100.0;

}

}

}

//Create a class BubbleSort to sort the data while read and write into the file

public class BubbleSort {

/\*\* Bubble sort method \*/

//Two arrays (account,bmi) need sort

public static void bubbleSort(Account[]list,BMI[]list1) {

//Before sort order, the initial ordering is incorrect

boolean sorted = false;

//To execute sorting

int n = list.length;

while(!sorted){

for (int k = 1; k < list.length; k++){

//The ordering is correct, then stop

sorted = true;

// Array may be sorted and next pass not needed

for (int i = 0; i < n - 1; i++) {

if ((list[i].getName()).compareTo(list[i + 1].getName()) > 0) {

// Swap list[i] with list[i + 1]

Account temp = list[i];

list[i] = list[i + 1];

list[i + 1] = temp;

BMI temp1 = list1[i];

list1[i] = list1[i + 1];

list1[i + 1] = temp1;

//To ordering again

sorted = false;

}

}

}

}

}

}

//Create a class Console about the data need to use scanner

import java.util.Scanner;

public class Console{

private static Scanner input = new Scanner(System.in);

//Prompt the user to read the data or input the data by himself

//Prompt the user for input how many data he want to key in

public static int getInt(String prompt) {

System.out.print(prompt);

int i = input.nextInt(); //Prompt the user to read the data or input the data by himself

input.nextLine(); // discard any other data entered on the line

return i;

}

//Prompt the user for input name

//Prompt the user for input address

//Prompt the user for input telephone

//Prompt the user for input birth

//Prompt the user for input test time

//Prompt the user for which file to read

//Prompt the user for the next action

public static String getString(String prompt){

System.out.print(prompt);

String s = input.next();

input.nextLine(); // discard any other data entered on the line

return s;

}

public static boolean getGender(String prompt){

System.out.print(prompt);

boolean gender = input.next().equals("1");//Prompt the user for input gender

input.nextLine();

return gender;

}

//Prompt the user for input weight

//Prompt the user for input height

public static double getDouble(String prompt){

System.out.print(prompt);

double d = input.nextDouble();

input.nextLine(); // discard any other data entered on the line

return d;

}

}

//User can key in the data, and put it in the file(data.txt), and then after sort the data by bubblesort, put it in the file(BMI.txt).User also can search people's data by name

import java.io.FileReader;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.FileWriter;

import java.io.BufferedWriter;

public class TestBMI{

public static void main(String[] args){

// display a welcome message

System.out.println("Welcome to the Health Management System");

System.out.println();

//Prompt the user for input how many data he want to key in

int n = Console.getInt("Enter how many data you want to key in: ");

//Put the data from scanner into data.txt

try{

FileWriter fw = new FileWriter("data.txt");

BufferedWriter bw = new BufferedWriter(fw);

//Use loop to input the amount of data that user want to key in

for(int i = 0; i < n; i++){

//Prompt the user for input name, gender, age, address, telephone, weight and height

String name = Console.getString("Enter your name: ");

String address = Console.getString("Enter your address: ");

String telephone = Console.getString("Enter your telephone: ");

String birth = Console.getString("(YYYY-MM-DD)" + "Enter your birth: ");

boolean gender = Console.getGender("(1-male, 0-female)" + "Enter your gender: ");

double height = Console.getDouble("Enter your height: ");

double weight = Console.getDouble("Enter your weight: ");

String testtime = Console.getString("(YYYY-MM-DD)" + "Enter the time you test: ");

bw.write(name + "," + address + "," + telephone + "," + birth);

if (gender){

bw.write(",1");

}

else{

bw.write(",0");

}

bw.write("," + height + "," + weight + "," + testtime + "\r\n");

bw.flush();

}

bw.close();

}

catch(IOException e){

e.printStackTrace();

}

//Use account and bmi array to manage data

Account[] account = new Account[n];

BMI[] bmi = new BMI[n];

//Read the data from data.txt

//The data in array is by lines, one line have people's name, address, telephone, birthday, gender, height, weight and testtime

//Use "," to seperate the data in one line

try{

int row = 0;

FileReader fr = new FileReader("data.txt");

BufferedReader br = new BufferedReader(fr);

String line;

while ((line=br.readLine()) != null){

String[] str = line.split(",");

account[row] = new Account();

bmi[row] = new BMI();

account[row].setName(str[0]);

account[row].setAddress(str[1]);

account[row].setTelephone(str[2]);

bmi[row].setBirth(str[3]);

bmi[row].setGenderstr(str[4]);

bmi[row].setHeight(Double.parseDouble(str[5]));

bmi[row].setWeight(Double.parseDouble(str[6]));

account[row].setTestTime(str[7]);

row ++;

}

}

catch(IOException e){

e.printStackTrace();

}

//Write the data into BMI.txt

//Use bubblesort to sort the data

try{

FileWriter fw = new FileWriter("BMI.txt");

BufferedWriter bw = new BufferedWriter(fw);

BubbleSort.bubbleSort(account, bmi);

for(int i = 0; i < n; i++){

String g = null;

if(bmi[i].getGenderstr() != null && bmi[i].getGenderstr().equals("0")){

g = "female";

}

else{

g = "male";

}

bw.write("Name:" + account[i].getName() + " Address:" + account[i].getAddress() + " Telephone:" + account[i].getTelephone() + " Birthday:" + bmi[i].getBirth() + " Gender:" + g + " Height:" + bmi[i].getHeight() + " Weight:" + bmi[i].getWeight() + " BMI:" + bmi[i].getBMI() + " Body Status:" + bmi[i].getStatus() + " Body Fat:" + bmi[i].getBodyFat() + " Age:" + bmi[i].getAge() + " Test time:" + account[i].getTestTime() + "\r\n");

bw.flush();

}

bw.close();

}

catch(IOException e){

e.printStackTrace();

}

//Use while loop to ask user what is the next action

Boolean value = true;

while(value){

//Here are three options that user can choose

String answer = Console.getString("Continue? Next.End.Search(y/n/s):");

if(answer.equals("n")) {

System.out.println("Thanks for your using!");

value = false;

}

else if(answer.equals("s")){

//If user want to search the data, then just key in the person's name to get information

String tempName = Console.getString("Enter person's name:");

for(int i = 0; i < n; i++){

if(!account[i].getName().equals(tempName)){

System.out.println("UNKNOWN");

}

else{

String ge = null;

if(bmi[i].getGenderstr() != null && bmi[i].getGenderstr().equals("0")){

ge = "female";

}

else{

ge = "male";

}

System.out.println("Name:" + account[i].getName() + " Address:" + account[i].getAddress() + " Telephone:" + account[i].getTelephone() + " Birthday:" + bmi[i].getBirth() + " Gender:" + ge + " Height:" + bmi[i].getHeight() + " Weight:" + bmi[i].getWeight() + " BMI:" + bmi[i].getBMI() + " Body Status:" + bmi[i].getStatus() + " Body Fat:" + bmi[i].getBodyFat() + " Age:" + bmi[i].getAge() + " Test time:" + account[i].getTestTime());

value = true;

}

}

}

else if(answer.equals("y")){

value = false;

}

else{

System.out.println("Please enter a valid choice.");

}

}

}

}