**Course**: Principals of Software Development – ENSF 409

**Lab 1**

**Instructor**: M. Moshirpour

**Student Name**: Mitchell Sawatzky

**Date Submitted**: Jan 17, 2016

Exercise B (10 marks)

Clock.java

public class Clock {

//instance variables

private int day;

private int hour;

private int minute;

private int second;

//class constructors

public Clock(int nDay, int nHour, int nMinute, int nSecond) {

//setting uses the internal methods to take advantage of error handling

this.set\_day(nDay);

this.set\_hour(nHour);

this.set\_minute(nMinute);

this.set\_second(nSecond);

}

//default constructor

public Clock() {

this(0, 0, 0, 0); //calls the above constructor with 0 init values

}

//instance methods

public int get\_day() {

return day;

}

public int get\_hour() {

return hour;

}

public int get\_minute() {

return minute;

}

public int get\_second() {

return second;

}

public void set\_day(int nDay) {

day = nDay;

}

public void set\_hour(int nHour) {

if (nHour < 0 || nHour > 23) {

System.err.println("ERROR: new hour out of range: " + nHour);

} else {

hour = nHour;

}

}

public void set\_minute(int nMinute) {

if (nMinute < 0 || nMinute > 59) {

System.err.println("ERROR: new minute out of range: " + nMinute);

} else {

minute = nMinute;

}

}

public void set\_second(int nSecond) {

if (nSecond < 0 || nSecond > 59) {

System.err.println("ERROR: new second out of range: " + nSecond);

} else {

second = nSecond;

}

}

public void increment(int incr) {

int nTime = incr + second; //intermediate time value

if (nTime > 59) {

second = nTime % 60; //seconds leftover from division

nTime = nTime / 60 + minute; //total minutes after seconds got taken away

if (nTime > 59) {

minute = nTime % 60; //minutes leftover from division

nTime = nTime / 60 + hour; //total hours after minutes got taken away

if (nTime > 23) {

hour = nTime % 24; //leftover hours

day += nTime / 24; //days can't tick over

} else {

hour = nTime; //hours didn't dick over

}

} else {

minute = nTime; //minutes didn't tick over

}

} else {

second = nTime; //seconds didn't tick over

}

}

public int calculate\_total\_seconds() {

return (86400 \* day) + //86400 seconds in a day

(3600 \* hour) + //3600 seconds in an hour

(60 \* minute) + //60 seconds in a minute

second;

}

//program start point

public static void main(String[] args) {

// Create elapsed time with the default values of zeros for day, hour,

// minute and second.

Clock t1 = new Clock(); // Default constructor

//Elapsed time is 3 days, 1 hour, 4 mins and 5 secs

Clock t2 = new Clock(3, 1, 4, 5);

t1.set\_hour(23); // sets hour to 23

t1.set\_day(1); // sets day to 1

t1.set\_minute(59); // sets minute to 59

t1.set\_second(16); // sets day to 16

// prints: 1:23:59:16

System.out.println("Expecting 1:23:59:16 - " + t1.get\_day() + ":" + t1.get\_hour() +":" + t1.get\_minute() + ":" + t1.get\_second());

// increments time t1 by 44 seconds:

t1.increment(44);

// prints: 2:0:0:0

System.out.println("Expecting 2:0:0:0 - " + t1.get\_day() + ":" + t1.get\_hour() +":" + t1.get\_minute() + ":" + t1.get\_second());

// prints the total elapsed time in seconds: 172,800

System.out.printf("Expecting 172800 - %d\n", t1.calculate\_total\_seconds());

// prints: 3:1:4:5

System.out.println("Expecting 3:1:4:5 - " + t2.get\_day() + ":" + t2.get\_hour() +":" + t2.get\_minute() + ":" + t2.get\_second());

// increments time t1 by 69 seconds

t2.increment(69);

// prints 3:1:5:14

System.out.println("Expecting 3:1:5:14 - " + t2.get\_day() + ":" + t2.get\_hour() +":" + t2.get\_minute() + ":" + t2.get\_second());

// prints out the total elapsed time in seconds: 263,114

System.out.printf("Expecting 263114 - %d\n", t2.calculate\_total\_seconds());

// attempts to set an illegal minute values

t2.set\_minute(60);

// prints the previous minute value: 5

System.out.printf("Expecting 5 - %d\n", t2.get\_minute());

}

}

Terminal Output

Mitchell@ttys000 23:33 {0} [1]$ java Clock

Expecting 1:23:59:16 - 1:23:59:16

Expecting 2:0:0:0 - 2:0:0:0

Expecting 172800 - 172800

Expecting 3:1:4:5 - 3:1:4:5

Expecting 3:1:5:14 - 3:1:5:14

Expecting 263114 - 263114

ERROR: new minute out of range: 60

Expecting 5 – 5

Exercise C

SinValidator.java

import java.util.Scanner;

public class SinValidator {

private int[] SIN;

private int sumDigit(int x)

{

int result =0;

while(x > 0){

result += x % 10;

x = x /10;

}

return result;

}

public SinValidator(String sin) {

SIN = new int[9];

int i =0;

int counter =0;

while(i < sin.length()){

if(Character.isDigit(sin.charAt(i))){

if(counter < 9)

SIN[counter] =(int) sin.charAt(i) - 48;

counter++;

}

else{

System.err.println("Error: Invalid input by the user");

return;

}

i++;

}

if(counter != 9){

System.err.println("Error: SIN must be 9 digits...");

return;

}

}

public boolean validateSin() {

//note that since we're always adding steps, we can have a running total

int runningTotal = SIN[0] + SIN[2] + SIN[4] + SIN[6]; // Add first, third, fifth, and seventh digits

//iterate through second, fourth, sixth, and eighth digits

for (int i = 1; i < 8; i += 2) {

int intermediateMulti = SIN[i] \* 2; //multiply digit by two

while (intermediateMulti > 0) { //while there are still digits left

runningTotal += intermediateMulti % 10; //add least significant digit

intermediateMulti /= 10; //remove least significant digit

}

}

if (SIN[8] == 10 - (runningTotal % 10)) //check equality of 9th digit and 10 - least significant digit of the running total

return true;

else

return false;

}

public static void main(String[] args) {

// Read user input

String sin;

Scanner scan = new Scanner(System.in);

while (true)

{

System.out.println("Please enter your 9 digit social insurance number"

+ " or enter quit to terminate the program: ");

sin = scan.nextLine();

if(sin.toUpperCase().equals("QUIT"))

break;

SinValidator sv = new SinValidator(sin);

if(sv.validateSin())

System.out.println("Yes this is a valid SIN\n");

else

System.out.println("No this is NOT a valid SIN\n");

}

}

}

Terminal Output

Mitchell@ttys000 00:23 {0} [1]$ java SinValidator

Please enter your 9 digit social insurance number or enter quit to terminate the program:

366497626

Yes this is a valid SIN

Please enter your 9 digit social insurance number or enter quit to terminate the program:

123456789

No this is NOT a valid SIN

Please enter your 9 digit social insurance number or enter quit to terminate the program:

quit