НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №1.2

з дисципліни

“Програмування мобільних систем”

Виконав:

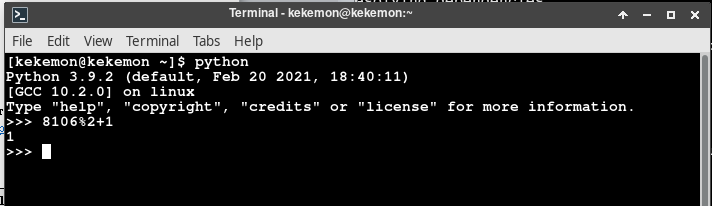
студент групи IВ-81

ЗК 8106

Бухтій О.В.

Київ 2021

1. **Визначення варіанту:**



Посилання на репозиторій: <https://github.com/KekemonBS/PMS/tree/main/LAB_2>

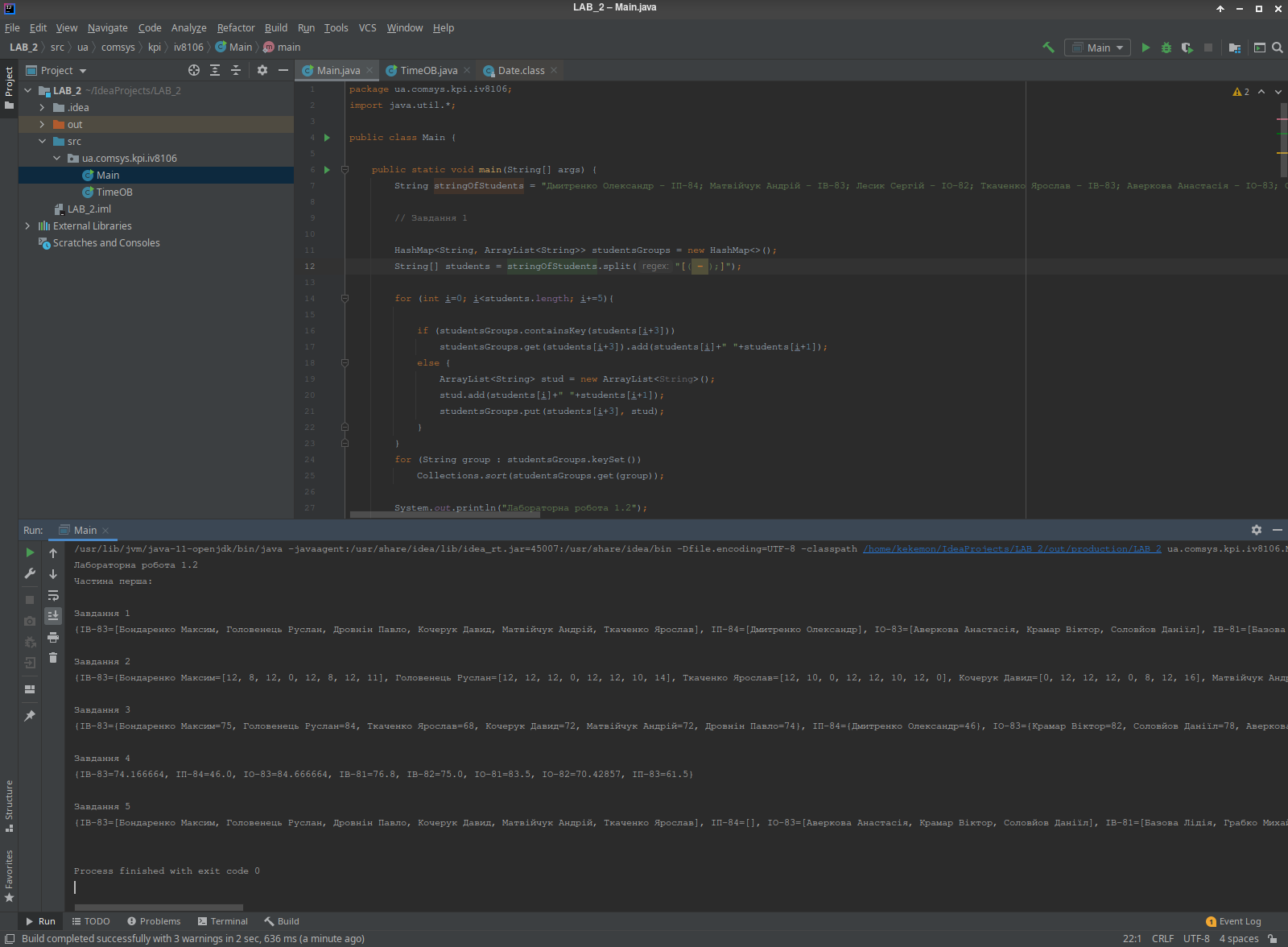
**Умова:**

|  |
| --- |
| **Варіант 1** |
| 1. Створіть клас TimeXY, який представляє час, де *X* перша літера вашого імені, *Y* перша літера вашого прізвища. 2. Додайте три властивості типу UInt в клас TimeXY, для представлення годин, хвилин та секунд. 3. Додайте методи ініціалізації:    1. з нульовими значеннями за замовчанням;    2. з заданим набором значень (години, хвилини, секунди) (перевірте вхідні значення – години ∈ [0, 23], хвилини ∈ [0, 59], секунди ∈ [0, 59]);    3. з заданим об'єктом типу Date. 4. Додайте методи, що повертають:    1. значення типу String у форматі “*hh*:*mm*:*ss* *ZZ*”, де *hh* – години, *mm* – хвилини, *ss* – секунди, *ZZ* – “AM” або “PM” (12-годинний формат), наприклад, “12:00:00 AM”;    2. об'єкт типу TimeXY, що представляє суму (за відповідними властивостями) поточного об'єкта та об'єкта, що отриманий як вхідний параметр;    3. об'єкт типу TimeXY, що представляє різницю (за відповідними властивостями) поточного об'єкта та об'єкта, що отриманий як вхідний параметр. 5. Додайте методи класу, що повертають:    1. об'єкт типу TimeXY, що представляє суму (за відповідними властивостями) двох об'єктів, що отримані як вхідні параметри;    2. об'єкт типу TimeXY, що представляє різницю (за відповідними властивостями) двох об'єктів, що отримані як вхідні параметри; 6. Створіть декілька об'єктів типу TimeXY за допомогою різних ініціалізаторів (методи з кроку 5). 7. Продемонструйте використання методів з кроків 6 та 7 (виведіть результати). 8. Переконайтеся, що сума та різниця об'єктів обчислюється правильно:    1. сума відповідних властивостей не може бути більшою ніж 59 (23 для годин) (23:59:59 + 12:00:01 = 12:00:00);    2. різниця відповідних властивостей не може бути меншою ніж 0 (00:00:00 - 00:00:01 = 23:59:59). |

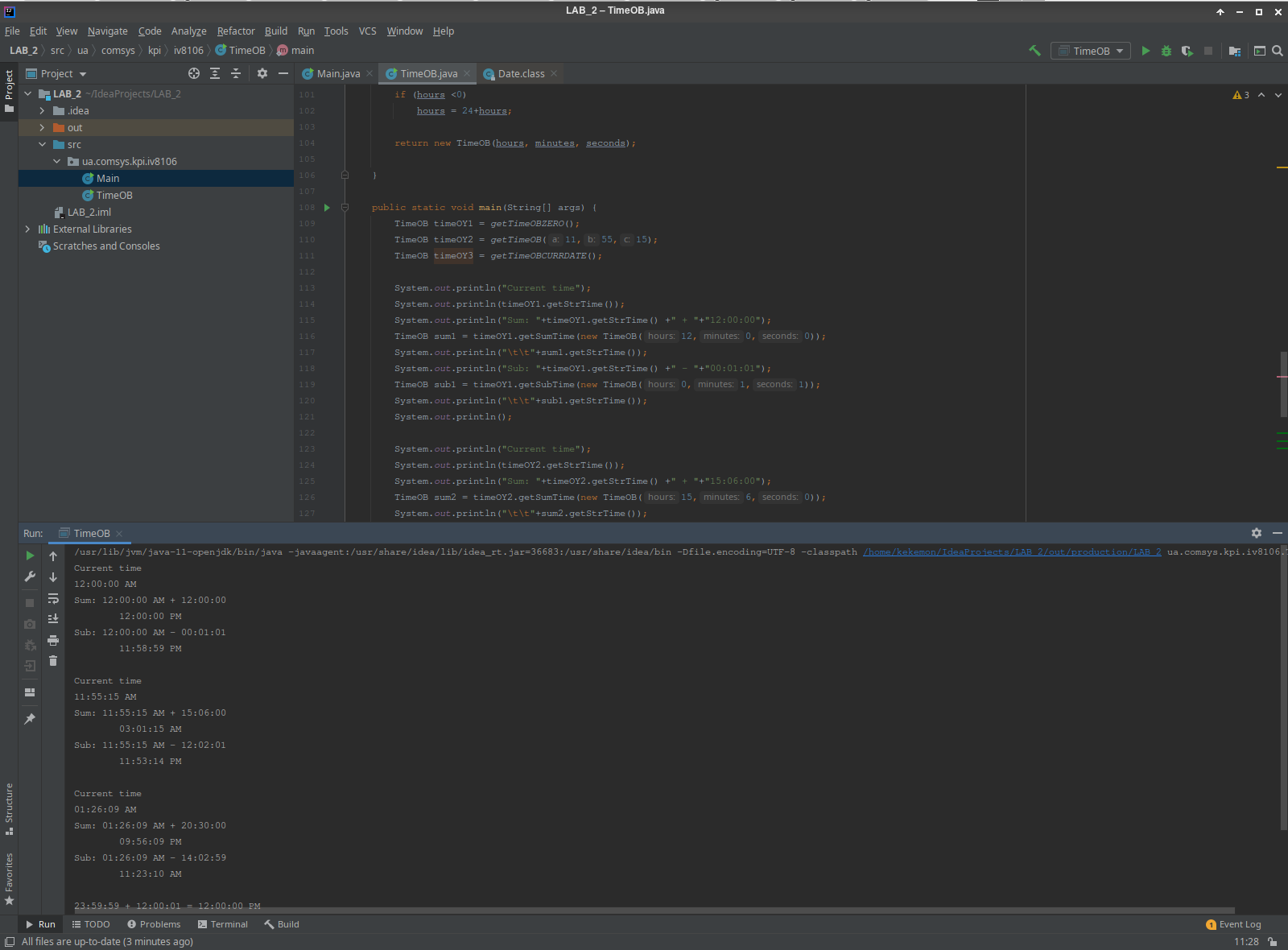
**Виконання:**

Скріншоти:

Перша частина:



Друга частина:



**Лістинг коду:**

**Main.java**

package ua.comsys.kpi.iv8106;

import java.util.\*;

public class Main {

public static void main(String[] args) {

String stringOfStudents = "Дмитренко Олександр - ІП-84; Матвійчук Андрій - ІВ-83; Лесик Сергій - ІО-82; Ткаченко Ярослав - ІВ-83; Аверкова Анастасія - ІО-83; Соловйов Даніїл - ІО-83; Рахуба Вероніка - ІО-81; Кочерук Давид - ІВ-83; Лихацька Юлія - ІВ-82; Головенець Руслан - ІВ-83; Ющенко Андрій - ІО-82; Мінченко Володимир - ІП-83; Мартинюк Назар - ІО-82; Базова Лідія - ІВ-81; Снігурець Олег - ІВ-81; Роман Олександр - ІО-82; Дудка Максим - ІО-81; Кулініч Віталій - ІВ-81; Жуков Михайло - ІП-83; Грабко Михайло - ІВ-81; Іванов Володимир - ІО-81; Востриков Нікіта - ІО-82; Бондаренко Максим - ІВ-83; Скрипченко Володимир - ІВ-82; Кобук Назар - ІО-81; Дровнін Павло - ІВ-83; Тарасенко Юлія - ІО-82; Дрозд Світлана - ІВ-81; Фещенко Кирил - ІО-82; Крамар Віктор - ІО-83; Іванов Дмитро - ІВ-82";

// Завдання 1

HashMap<String, ArrayList<String>> studentsGroups = new HashMap<>();

String[] students = stringOfStudents.split("[( - );]");

for (int i=0; i<students.length; i+=5){

if (studentsGroups.containsKey(students[i+3]))

studentsGroups.get(students[i+3]).add(students[i]+" "+students[i+1]);

else {

ArrayList<String> stud = new ArrayList<String>();

stud.add(students[i]+" "+students[i+1]);

studentsGroups.put(students[i+3], stud);

}

}

for (String group : studentsGroups.keySet())

Collections.sort(studentsGroups.get(group));

System.out.println("Лабораторна робота 1.2");

System.out.println("Частина перша: ");

System.out.println();

System.out.println("Завдання 1");

System.out.println(studentsGroups);

System.out.println();

// Завдання 2

int[] points = {12, 12, 12, 12, 12, 12, 12, 16};

HashMap<String, HashMap<String, ArrayList<Integer>>> studentPoints = new HashMap<>();

for (String group: studentsGroups.keySet()){

if (!studentPoints.containsKey(group)){

HashMap<String,ArrayList<Integer>> temp = new HashMap<>();

studentPoints.put(group, temp);

}

for (String student : studentsGroups.get(group)) {

ArrayList<Integer> temp = new ArrayList<>();

studentPoints.get(group).put(student, temp);

ArrayList<Integer> s = studentPoints.get(group).get(student);

for (int point : points)

s.add(randomValue(point));

}

}

System.out.println("Завдання 2");

System.out.println(studentPoints);

System.out.println();

// Завдання 3

HashMap<String, HashMap<String, Integer>> sumPoints = new HashMap<>();

for (String group: studentPoints.keySet()){

if (!sumPoints.containsKey(group)){

HashMap<String, Integer> temp = new HashMap<>();

sumPoints.put(group, temp);

}

for (Map.Entry<String, ArrayList<Integer>> student : studentPoints.get(group).entrySet()) {

int sum = 0;

for (int point : studentPoints.get(group).get(student.getKey()))

sum += point;

sumPoints.get(group).put(student.getKey(), sum);

}

}

System.out.println("Завдання 3");

System.out.println(sumPoints);

System.out.println();

// Завдання 4

HashMap<String, Float> groupAvg = new HashMap<>();

for (String group: sumPoints.keySet()){

Set<String> students2 = sumPoints.get(group).keySet();

int sum = 0, n = students2.size();

for (String student : students2) {

sum += sumPoints.get(group).get(student);

}

groupAvg.put(group, (float)sum/n);

}

System.out.println("Завдання 4");

System.out.println(groupAvg);

System.out.println();

// Завдання 5

HashMap<String, ArrayList<String>> passedPerGroup = new HashMap<>();

for (String group : studentsGroups.keySet()){

if (!passedPerGroup.containsKey(group)){

ArrayList<String> temp = new ArrayList<>();

passedPerGroup.put(group, temp);

}

for (String student : studentsGroups.get(group)) {

if (sumPoints.get(group).get(student) >= 60){

passedPerGroup.get(group).add(student);

}

}

}

System.out.println("Завдання 5");

System.out.println(passedPerGroup);

System.out.println();

}

private static int randomValue(int maxValue){

Random rand = new Random();

switch(rand.nextInt(6)) {

case 1:

return (int) (maxValue \* 0.7);

case 2:

return (int) (maxValue \* 0.9);

case 3:

case 4:

case 5:

return maxValue;

default:

return 0;

}

}

}

**TimeOB.java**

package ua.comsys.kpi.iv8106;

import java.util.Date;

public class TimeOB {

public static TimeOB getTimeOB(int a, int b, int c) {

return new TimeOB(a, b, c);

}

public static TimeOB getTimeOBZERO() {

return new TimeOB();

}

public static TimeOB getTimeOBCURRDATE() {

return new TimeOB(new Date());

}

Integer hours, minutes, seconds;

TimeOB(){

this(0,0,0);

}

TimeOB(Date date){

this(date.getHours(), date.getMinutes(), date.getSeconds());

}

TimeOB(int hours, int minutes, int seconds) {

boolean check = (hours >=0 && hours<24 && minutes>=0 && minutes<60 &&

seconds>=0 && seconds<60);

if (check) {

this.hours = hours;

this.minutes = minutes;

this.seconds = seconds;

}

else {

System.out.println("Invalid parameter for time");

System.exit(0);

}

}

String getStrTime(){

String zone;

if (this.hours >=0 && this.hours<12)

zone = "AM";

else

zone = "PM";

int temp\_hours = (this.hours == 0 | this.hours == 12) ? 12 : this.hours%12;

return String.format("%02d:%02d:%02d %s", temp\_hours, this.minutes, this.seconds, zone);

}

TimeOB getSumTime(TimeOB timeOY){

// TimeOY now = new TimeOY(new Date());

int hours, minutes, seconds;

hours = timeOY.hours + this.hours;

minutes = timeOY.minutes + this.minutes;

seconds = timeOY.seconds + this.seconds;

if (seconds >59){

seconds= seconds%60;

minutes++;

}

if (minutes >59) {

minutes = minutes % 60;

hours++;

}

hours = hours%24;

return new TimeOB(hours, minutes, seconds);

}

TimeOB getSubTime(TimeOB timeOY){

int hours, minutes, seconds;

hours = this.hours - timeOY.hours;

minutes = this.minutes - timeOY.minutes;

seconds = this.seconds - timeOY.seconds;

if (seconds < 0){

seconds = 60 + seconds;

if ( minutes == 0 ){

minutes=59;

hours --;

}

else

minutes--;

}

if (minutes <0){

minutes = 60+minutes;

if ( hours == 0 )

hours=23;

else

hours--;

}

if (hours <0)

hours = 24+hours;

return new TimeOB(hours, minutes, seconds);

}

public static void main(String[] args) {

TimeOB timeOY1 = getTimeOBZERO();

TimeOB timeOY2 = getTimeOB(11,55,15);

TimeOB timeOY3 = getTimeOBCURRDATE();

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

System.out.println(timeOY1.getStrTime());

System.out.println("Sum: "+timeOY1.getStrTime() +" + "+"12:00:00");

TimeOB sum1 = timeOY1.getSumTime(new TimeOB(12,0,0));

System.out.println("\t\t"+sum1.getStrTime());

System.out.println("Sub: "+timeOY1.getStrTime() +" - "+"00:01:01");

TimeOB sub1 = timeOY1.getSubTime(new TimeOB(0,1,1));

System.out.println("\t\t"+sub1.getStrTime());

System.out.println();

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

System.out.println(timeOY2.getStrTime());

System.out.println("Sum: "+timeOY2.getStrTime() +" + "+"15:06:00");

TimeOB sum2 = timeOY2.getSumTime(new TimeOB(15,6,0));

System.out.println("\t\t"+sum2.getStrTime());

System.out.println("Sub: "+timeOY2.getStrTime() +" - "+"12:02:01");

TimeOB sub2 = timeOY2.getSubTime(new TimeOB(12,2,1));

System.out.println("\t\t"+sub2.getStrTime());

System.out.println();

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

String time33 = timeOY3.getStrTime();

System.out.println(time33);

System.out.println("Sum: "+time33 +" + "+"20:30:00");

TimeOB sum3 = timeOY3.getSumTime(new TimeOB(20,30,0));

System.out.println("\t\t"+sum3.getStrTime());

System.out.println("Sub: "+time33 +" - "+"14:02:59");

TimeOB sub3 = timeOY3.getSubTime(new TimeOB(14,2,59));

System.out.println("\t\t"+sub3.getStrTime());

System.out.println();

TimeOB time1 = new TimeOB(23, 59, 59);

TimeOB time2 = new TimeOB(12, 0, 1);

System.out.println("23:59:59 + 12:00:01 = " + time1.getSumTime(time2).getStrTime());

TimeOB time3 = new TimeOB(0, 0, 0);

TimeOB time4 = new TimeOB(0, 0, 1);

System.out.println("00:00:00 - 00:00:01 = " + time3.getSubTime(time4).getStrTime());

TimeOB time11 = new TimeOB(0, 0, 0);

TimeOB time22 = new TimeOB(12, 0, 0);

System.out.println("00:00:00 = " + time11.getStrTime());

System.out.println("12:00:00 = " + time22.getStrTime());

}

}

**Висновок:**

В середовищі розробки Intelij IDEA, було створено 2 програми для роботи з часом (**Main.java**) та словниками (**TimeOB.java**). Обидві програми працюють коректно.