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 / Розроблення клієнтських додатків для мобільних платформ”

Виконав:

студент групи ІО-81

ЗК ІО-8118

Лялін Ігор

Київ 2021

**(8118%2)+1 = Варіант 1**

**Скріншоти роботи програм**

Завдання 1

{'IO-81': ['Дудка Максим', 'Иванов Володимир', 'Кобук Назар', 'Рахуба Вероніка'], 'IO-82': ['Востриков Нікіта', 'Лесик Сергій', 'Мартинюк Назар', 'Роман Олександр', 'Тарасенко Юлія', 'Фещенко Кирил', 'Ющенко Андрій'], 'IO-83': ['Аверкова Анастасія', 'Крамар Віктор', 'Соловйов Даніїл'], 'IВ-81': ['Базова Лідія', 'Грабко Михайло', 'Дрозд Світлана', 'Кулініч Віталій', 'Снігурець Олег'], 'IВ-82': ['Иванов Дмитро', 'Лихацька Юлія', 'Скрипченко Володимир'], 'IВ-83': ['Бондаренко Максим', 'Головенець Руслан', 'Дровнін Павло', 'Кочерук Давид', 'Матвійчук Андрій', 'Ткаченко Ярослав'], 'IП-83': ['Жуков Михайло', 'Мінченко Володимир'], 'IП-84': ['Дмитренко Олександр']}

Завдання 2

{'IO-81': {'Дудка Максим': [10, 3, 6, 7, 3, 2, 3, 6], 'Иванов Володимир': [3, 7, 5, 12, 0, 1, 7, 15], 'Кобук Назар': [0, 7, 1, 6, 2, 7, 5, 16], 'Рахуба Вероніка': [3, 5, 6, 1, 0, 4, 1, 0]}, 'IO-82': {'Востриков Нікіта': [8, 6, 4, 7, 7, 6, 2, 8], 'Лесик Сергій': [12, 7, 11, 7, 10, 6, 4, 7], 'Мартинюк Назар': [6, 9, 12, 2, 0, 5, 1, 6], 'Роман Олександр': [6, 5, 6, 3, 6, 11, 8, 7], 'Тарасенко Юлія': [11, 3, 12, 7, 7, 5, 4, 3], 'Фещенко Кирил': [12, 7, 2, 11, 2, 8, 6, 1], 'Ющенко Андрій': [5, 8, 9, 2, 6, 7, 8, 11]}, 'IO-83': {'Аверкова Анастасія': [12, 10, 5, 4, 10, 0, 6, 3], 'Крамар Віктор': [1, 8, 8, 10, 5, 0, 3, 16], 'Соловйов Даніїл': [12, 3, 12, 10, 1, 11, 6, 4]}, 'IВ-81': {'Базова Лідія': [5, 11, 5, 1, 4, 8, 3, 8], 'Грабко Михайло': [12, 1, 6, 1, 7, 8, 6, 16], 'Дрозд Світлана': [3, 5, 10, 10, 2, 0, 11, 0], 'Кулініч Віталій': [3, 8, 1, 3, 8, 12, 11, 14], 'Снігурець Олег': [5, 4, 1, 6, 3, 5, 7, 13]}, 'IВ-82': {'Иванов Дмитро': [5, 7, 8, 12, 11, 2, 1, 12], 'Лихацька Юлія': [3, 9, 0, 6, 8, 3, 8, 12], 'Скрипченко Володимир': [12, 7, 8, 7, 3, 7, 4, 0]}, 'IВ-83': {'Бондаренко Максим': [2, 8, 7, 9, 8, 11, 8, 0], 'Головенець Руслан': [9, 7, 0, 10, 4, 2, 10, 1], 'Дровнін Павло': [1, 0, 1, 0, 7, 10, 1, 9], 'Кочерук Давид': [12, 1, 1, 3, 10, 10, 9, 8], 'Матвійчук Андрій': [10, 12, 10, 1, 12, 4, 0, 11], 'Ткаченко Ярослав': [0, 6, 6, 9, 6, 9, 10, 10]}, 'IП-83': {'Жуков Михайло': [8, 6, 7, 6, 7, 6, 8, 0], 'Мінченко Володимир': [1, 11, 8, 6, 5, 6, 1, 13]}, 'IП-84': {'Дмитренко Олександр': [6, 0, 12, 6, 10, 4, 12, 7]}}

Завдання 3

{'IO-81': {'Дудка Максим': 40, 'Иванов Володимир': 50, 'Кобук Назар': 44, 'Рахуба Вероніка': 20}, 'IO-82': {'Востриков Нікіта': 48, 'Лесик Сергій': 64, 'Мартинюк Назар': 41, 'Роман Олександр': 52, 'Тарасенко Юлія': 52, 'Фещенко Кирил': 49, 'Ющенко Андрій': 56}, 'IO-83': {'Аверкова Анастасія': 50, 'Крамар Віктор': 51, 'Соловйов Даніїл': 59}, 'IВ-81': {'Базова Лідія': 45, 'Грабко Михайло': 57, 'Дрозд Світлана': 41, 'Кулініч Віталій': 60, 'Снігурець Олег': 44}, 'IВ-82': {'Иванов Дмитро': 58, 'Лихацька Юлія': 49, 'Скрипченко Володимир': 48}, 'IВ-83': {'Бондаренко Максим': 53, 'Головенець Руслан': 43, 'Дровнін Павло': 29, 'Кочерук Давид': 54, 'Матвійчук Андрій': 60, 'Ткаченко Ярослав': 56}, 'IП-83': {'Жуков Михайло': 48, 'Мінченко Володимир': 51}, 'IП-84': {'Дмитренко Олександр': 57}}

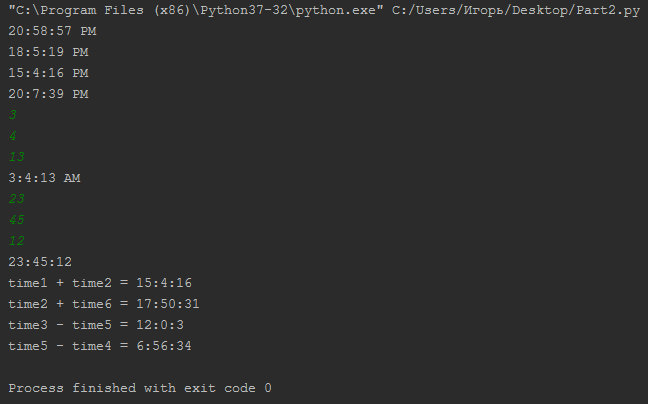
Завдання 4

{'IO-81': 38.5, 'IO-82': 51.714285714285715, 'IO-83': 53.333333333333336, 'IВ-81': 49.4, 'IВ-82': 51.666666666666664, 'IВ-83': 49.166666666666664, 'IП-83': 49.5, 'IП-84': 57.0}

Завдання 5

{'IO-81': [], 'IO-82': ['Лесик Сергій'], 'IO-83': [], 'IВ-81': ['Кулініч Віталій'], 'IВ-82': [], 'IВ-83': ['Матвійчук Андрій'], 'IП-83': [], 'IП-84': []}

Process finished with exit code 0



**Лістинг коду**

Part1.py

# Частина 1  
import random  
  
# Дано рядок у форматі "Student1 - Group1; Student2 - Group2; ..."  
  
studentsStr = "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; " \  
 "Иванов 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; Иванов Дмитро - ІВ-82"  
  
# Завдання 1  
# Заповніть словник, де:  
# - ключ – назва групи  
# - значення – відсортований масив студентів, які відносяться до відповідної групи  
  
studentsGroups = {}  
  
# Ваш код починається тут  
pairs = []  
IO81, IO82, IO83, IV81, IV82, IV83, IP83, IP84 = [], [], [], [], [], [], [], []  
studentgroup = studentsStr.split("; ")  
for i in studentgroup:  
 i = i.split(" - ")  
 pairs.append(i)  
for i in range(0, 31):  
 if "ІО-81" in pairs[i]:  
 IO81.append(pairs[i][0])  
 IO81.sort()  
for i in range(0, 31):  
 if "ІО-82" in pairs[i]:  
 IO82.append(pairs[i][0])  
 IO82.sort()  
for i in range(0, 31):  
 if "ІО-83" in pairs[i]:  
 IO83.append(pairs[i][0])  
 IO83.sort()  
for i in range(0, 31):  
 if "ІВ-81" in pairs[i]:  
 IV81.append(pairs[i][0])  
 IV81.sort()  
for i in range(0, 31):  
 if "ІВ-82" in pairs[i]:  
 IV82.append(pairs[i][0])  
 IV82.sort()  
for i in range(0, 31):  
 if "ІВ-83" in pairs[i]:  
 IV83.append(pairs[i][0])  
 IV83.sort()  
for i in range(0, 31):  
 if "ІП-83" in pairs[i]:  
 IP83.append(pairs[i][0])  
 IP83.sort()  
for i in range(0, 31):  
 if "ІП-84" in pairs[i]:  
 IP84.append(pairs[i][0])  
 IP84.sort()  
studentsGroups.setdefault('IO-81', IO81)  
studentsGroups.setdefault('IO-82', IO82)  
studentsGroups.setdefault('IO-83', IO83)  
studentsGroups.setdefault('IВ-81', IV81)  
studentsGroups.setdefault('IВ-82', IV82)  
studentsGroups.setdefault('IВ-83', IV83)  
studentsGroups.setdefault('IП-83', IP83)  
studentsGroups.setdefault('IП-84', IP84)  
# Ваш код закінчується тут  
  
print("Завдання 1")  
print(studentsGroups)  
print()  
  
# Дано масив з максимально можливими оцінками  
  
points = [12, 12, 12, 12, 12, 12, 12, 16]  
  
# Завдання 2  
# Заповніть словник, де:  
# - ключ – назва групи  
# - значення – словник, де:  
# - ключ – студент, який відносяться до відповідної групи  
# - значення – масив з оцінками студента (заповніть масив випадковими значеннями)  
  
studentPoints = {}  
  
  
# Ваш код починається тут  
def random\_numbers():  
 random\_list = [random.randint(0, 12) for j in range(7)]  
 random\_list.append(random.randint(0, 16))  
 return random\_list  
  
  
dict1, dict2, dict3, dict4, dict5, dict6, dict7, dict8 = {}, {}, {}, {}, {}, {}, {}, {}  
for i in IO81:  
 dict1.setdefault(i, random\_numbers())  
for i in IO82:  
 dict2.setdefault(i, random\_numbers())  
for i in IO83:  
 dict3.setdefault(i, random\_numbers())  
for i in IV81:  
 dict4.setdefault(i, random\_numbers())  
for i in IV82:  
 dict5.setdefault(i, random\_numbers())  
for i in IV83:  
 dict6.setdefault(i, random\_numbers())  
for i in IP83:  
 dict7.setdefault(i, random\_numbers())  
for i in IP84:  
 dict8.setdefault(i, random\_numbers())  
studentPoints.update({'IO-81': dict1})  
studentPoints.update({'IO-82': dict2})  
studentPoints.update({'IO-83': dict3})  
studentPoints.update({'IВ-81': dict4})  
studentPoints.update({'IВ-82': dict5})  
studentPoints.update({'IВ-83': dict6})  
studentPoints.update({'IП-83': dict7})  
studentPoints.update({'IП-84': dict8})  
# Ваш код закінчується тут  
  
print("Завдання 2")  
print(studentPoints)  
print()  
  
# # Завдання 3  
# # Заповніть словник, де:  
# # - ключ – назва групи  
# # - значення – словник, де:  
# # - ключ – студент, який відносяться до відповідної групи  
# # - значення – сума оцінок студента  
  
sumPoints = {}  
  
# # Ваш код починається тут  
sumPoints = studentPoints.copy()  
for keys, values in sumPoints.items():  
 for key, value in values.items():  
 b = sum(value)  
 values.update({key: b})  
 sumPoints.update({keys: values})  
# # Ваш код закінчується тут  
  
print("Завдання 3")  
print(sumPoints)  
print()  
  
# # Завдання 4  
# # Заповніть словник, де:  
# # - ключ – назва групи  
# # - значення – середня оцінка всіх студентів групи  
  
groupAvg = {}  
  
# # Ваш код починається тут  
c = []  
d = []  
e = 0  
groupAvg = studentPoints.copy()  
for keys, values in groupAvg.items():  
 for key, value in values.items():  
 c.append(value)  
 e = sum(c) / len(c)  
 d.append(e)  
 e = 0  
 c.clear()  
groups = groupAvg.keys()  
groupAvg = dict(zip(groups, d))  
  
# # Ваш код закінчується тут  
  
print("Завдання 4")  
print(groupAvg)  
print()  
  
# # Завдання 5  
# # Заповніть словник, де:  
# # - ключ – назва групи  
# # - значення – масив студентів, які мають >= 60 балів  
  
passedPerGroup = {}  
  
# # Ваш код починається тут  
dictionary = sumPoints.copy()  
for keys, values in dictionary.items():  
 student\_list = []  
 for key, value in values.items():  
 if value >= 60:  
 student\_list.append(key)  
 passedPerGroup.update({keys: student\_list})  
# # Ваш код закінчується тут  
  
print("Завдання 5")  
print(passedPerGroup)

Part2.py

from datetime import time  
  
  
class TimeIL:  
 def \_\_init\_\_(self, hours=0, minutes=0, seconds=0):  
 self.hours = hours  
 self.minutes = minutes  
 self.seconds = seconds  
  
 def \_\_str\_\_(self):  
 return '{}:{}:{}'.format(self.hours, self.minutes, self.seconds)  
  
 def input\_time(self):  
 self.hours = int(input())  
 self.minutes = int(input())  
 self.seconds = int(input())  
  
 def datetime\_time(self):  
 self.hours = int(input())  
 self.minutes = int(input())  
 self.seconds = int(input())  
 t = time(self.hours, self.minutes, self.seconds)  
 return print(t)  
  
 def display\_time(self):  
 if self.hours > 23 or self.hours < 0 or self.minutes > 59 or self.minutes < 0 or self.seconds > 59 or \  
 self.seconds < 0:  
 print('Error: Entered time is not correct')  
 else:  
 if self.hours <= 12:  
 print(f'{self.hours}:{self.minutes}:{self.seconds} AM')  
 else:  
 print(f'{self.hours}:{self.minutes}:{self.seconds} PM')  
  
 def add\_time(self, time\_1):  
 self.hours += time\_1.hours  
 self.minutes += time\_1.minutes  
 self.seconds += time\_1.seconds  
 if self.hours < 0 or self.minutes < 0 or self.seconds < 0:  
 return 'Error: Countered time is not correct'  
 if self.seconds > 59:  
 self.seconds = self.seconds - 60  
 self.minutes = self.minutes + 1  
 if self.minutes > 59:  
 self.minutes = self.minutes - 60  
 self.hours = self.hours + 1  
 if self.hours > 23:  
 self.hours = self.hours - 24  
 return TimeIL(self.hours, self.minutes, self.seconds)  
  
 def sub\_time(self, time\_1):  
 self.hours -= time\_1.hours  
 self.minutes -= time\_1.minutes  
 self.seconds -= time\_1.seconds  
 if self.seconds < 0:  
 self.seconds = self.seconds + 60  
 self.minutes = self.minutes - 1  
 if self.minutes < 0:  
 self.minutes = self.minutes + 60  
 self.hours = self.hours - 1  
 if self.hours < 0:  
 self.hours = self.hours + 24  
 return TimeIL(self.hours, self.minutes, self.seconds)  
  
  
time1 = TimeIL(20, 58, 57)  
time1.display\_time()  
time2 = TimeIL(18, 5, 19)  
time2.display\_time()  
time3 = TimeIL(15, 4, 16)  
time3.display\_time()  
time4 = TimeIL(20, 7, 39)  
time4.display\_time()  
time5 = TimeIL()  
time5.input\_time()  
time5.display\_time()  
time6 = TimeIL()  
time6.datetime\_time()  
print(f"time1 + time2 = {time1.add\_time(time2)}")  
print(f"time2 + time6 = {time2.add\_time(time6)}")  
print(f"time3 - time5 = {time3.sub\_time(time5)}")  
print(f"time5 - time4 = {time5.sub\_time(time4)}")

**Висновок**

Під час виконання роботи було створено 2 програми мовою Python: одна для роботи зі словниками, інша - для роботи з класом TimeIL та його методами. Роботу програм було протестовано для всіх можливих випадків. Мету роботи досягнуто.