Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 2 з дисципліни

«Основи програмування 2»

# «Бінарні файли»

Варіант №8

Виконав студент ІП-14 Громов Владислав Сергійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

(прізвище, ім'я, по батькові)

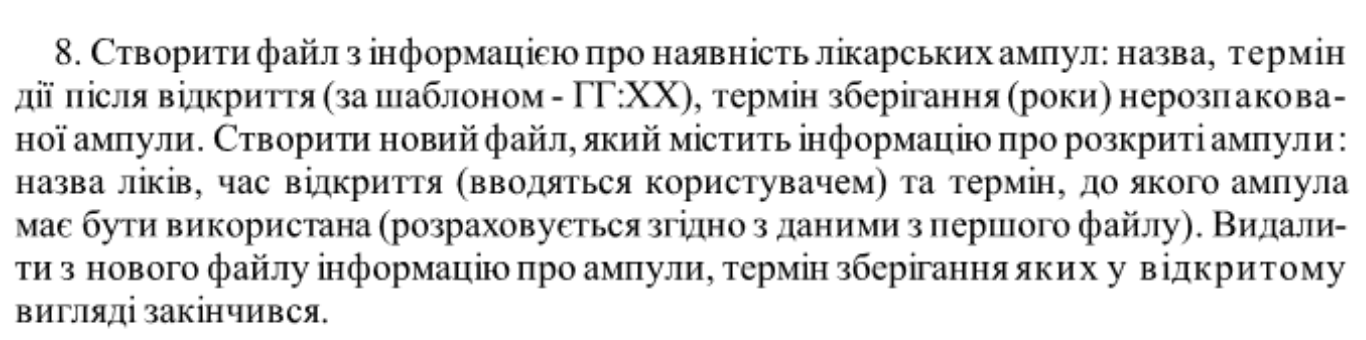
Київ 2021

**Лабораторна робота №2**

**Тема:** Бінарні файли

**Мета**: вивчити особливості створення і обробки бінарних файлів

**Хід роботи**

**Задача.** 

*Розв’язання*

**Математична модель**

|  |  |  |
| --- | --- | --- |
| **Змінна** | **Тип змінної** | **Призначення** |
| lines | Масив строк | Введене значення |
| path | Строка | Введене значення |
| drugsList | Масив строк | Вхідне значення |
| names | Масив строк | Введене значення |
| enter | Строка | Введене значення |
| x | Строка | Введене значення |

**Випробування коду на С#:**

*Код:*

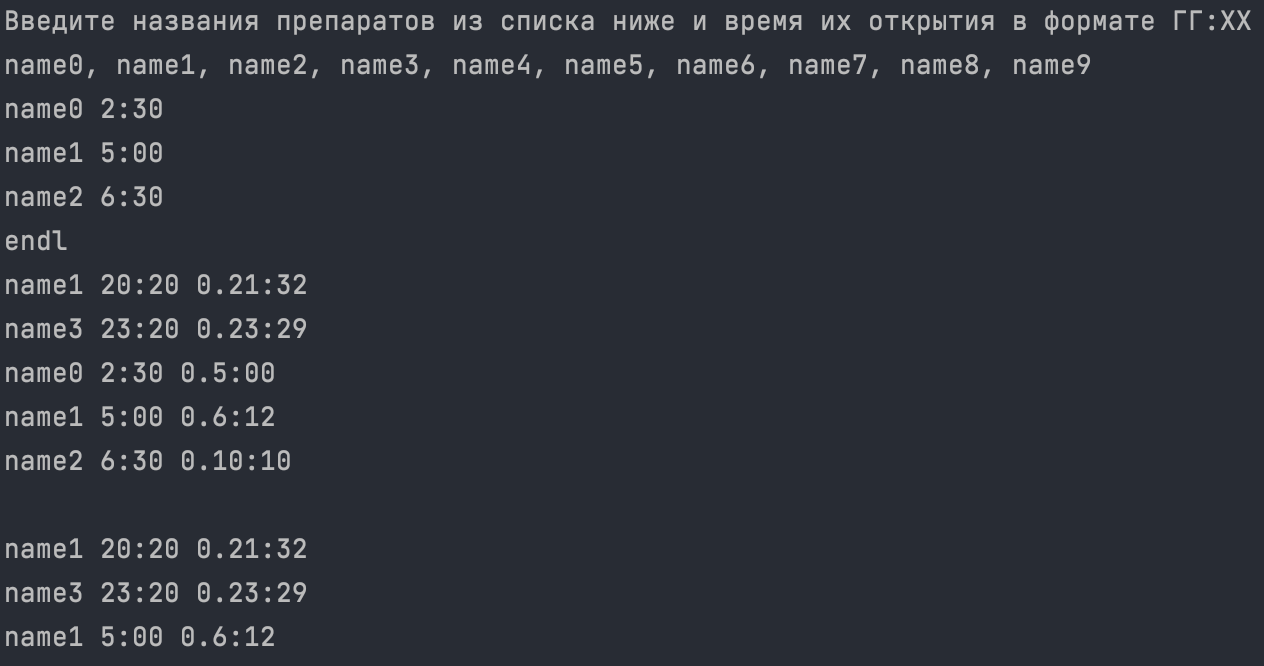
*Program.cs*

*using* System;  
*using* System.IO;  
*using static* Lab2.Funcs;  
  
*namespace* Lab2  
{  
 *class* Program  
 {  
 *static void* Main(*string*[] args)  
 {  
 *string* path = "";  
 Drug[] lines = {*new* Drug("name0", "2:30"), *new* Drug("name1", "1:10"), *new* Drug("name2", "0:31"), *new* Drug("name3", "10:10")};  
 FileWrite(lines, $"{path}drugs.txt", FileMode.OpenOrCreate);  
 Console.WriteLine("Введите названия препаратов из списка ниже и время их открытия в формате ГГ:ХХ");  
 Drug[] drugsList = FileRead($"{path}drugs.txt");  
 *foreach* (Drug drug *in* drugsList)  
 Console.Write($"{drug.DrugName} ");  
 Console.WriteLine();  
 *string* enter = "";  
 Drug[] withEnd = *new* Drug[0];  
 *while* (enter != "endl")  
 {  
 enter = Console.ReadLine();  
 *if* (enter == "endl")  
 *continue*;  
 *foreach* (*var* drug *in* drugsList)  
 {  
 *if*(drug.DrugName.Equals(enter.Split(" ")[0]))  
 {  
 *string* x = FindEndTime(DateTime.Parse(enter.Split(" ")[1]), DateTime.Parse(drug.Time));  
 withEnd = Add<Drug>(withEnd, *new* Drug(drug.DrugName, x));  
 }  
 }  
 }  
 FileWrite(withEnd, $"{path}openDrugs.txt", FileMode.OpenOrCreate);  
   
 TxtOut(FileRead($"{path}openDrugs.txt"));  
 RemoveOverdue($"{path}openDrugs.txt");  
 TxtOut(FileRead($"{path}openDrugs.txt"));  
 }  
   
 }  
}

*Funcs.cs:*

*using* System;  
*using* System.IO;  
*using* System.Runtime.Serialization.Formatters.Binary;  
*using* System.Text;  
*using* System.Threading.Tasks;  
  
*namespace* Lab2  
{  
 *public class* Funcs  
 {  
 *public static void* FileWrite(Drug[] drugs, *string* path, FileMode fm)   
 {  
   
 FileStream stream = File.Open(path, fm);  
 BinaryFormatter bf = *new* BinaryFormatter();  
 *foreach* (*var* i *in* drugs)  
 bf.Serialize(stream, i);   
 stream.Close();  
   
 }  
   
   
  
 *public static* T[] Add<T>(T[] ss, T z)  
 {  
 T[] x = *new* T[ss.Length+1];  
 *for* (*int* i = 0; i < ss.Length; i++)  
 {  
 x[i] = ss[i];  
 }  
 x[ss.Length] = z;  
 *return* x;  
 }  
   
 *public static* Drug[] FileRead(*string* path)  
 {  
 Drug[] txt = *new* Drug[0];  
 FileStream stream = File.Open(path, FileMode.Open);  
 BinaryFormatter bf = *new* BinaryFormatter();  
 *while* (stream.Position < stream.Length)  
 {  
 Drug drug = (Drug) bf.Deserialize(stream);  
 txt = Add(txt, drug);  
 }  
  
 stream.Close();  
 *return* txt;  
 }  
  
 *public static void* TxtOut(Drug[] ss)  
 {  
 *foreach* (*var* el *in* ss)  
 {  
 Console.WriteLine($"{el.DrugName}\t{el.Time}");  
 }  
  
 Console.WriteLine();  
 }  
  
 *public static* T[] RemoveElement<T>(T[] elements, *int* j)  
 {  
 T[] newLines = *new* T[elements.Length - 1];  
 *int* k = 0;  
 *for* (*int* i = 0; i < elements.Length; i++)  
 {  
 *if* (i != j)  
 {  
 newLines[k] = elements[i];  
 k++;  
 }  
   
 }  
  
 *return* newLines;  
 }  
  
 *public static string* FindEndTime(DateTime openTime, DateTime endTime)  
 {  
 openTime = openTime.AddHours(endTime.Hour);  
 openTime = openTime.AddMinutes(endTime.Minute);  
   
 *return* openTime.ToString("g");  
  
 }  
  
   
  
 *public static void* RemoveOverdue(*string* path)  
 {  
 Drug[] lines = FileRead(path);  
 *for* (*int* i = 0; i < lines.Length; i++)  
 {  
 DateTime endTime = DateTime.Parse(lines[i].Time);  
 *if* (endTime < DateTime.Now)  
 {  
 lines = RemoveElement(lines, i);  
 i--;  
 }  
 }  
 FileWrite(lines, path, FileMode.Create);  
   
 }  
   
 [Serializable]  
 *public struct* Drug  
 {  
 *public string* DrugName{get; set;}  
 *public string* Time{get; set;}  
  
   
 *public* Drug(*string* drugName, *string* time)  
 {  
 DrugName = drugName;  
 Time = time;  
 }  
  
 }  
   
 }  
}

*Результат:*

****

**Випробування коду на Python:**

*Код:*

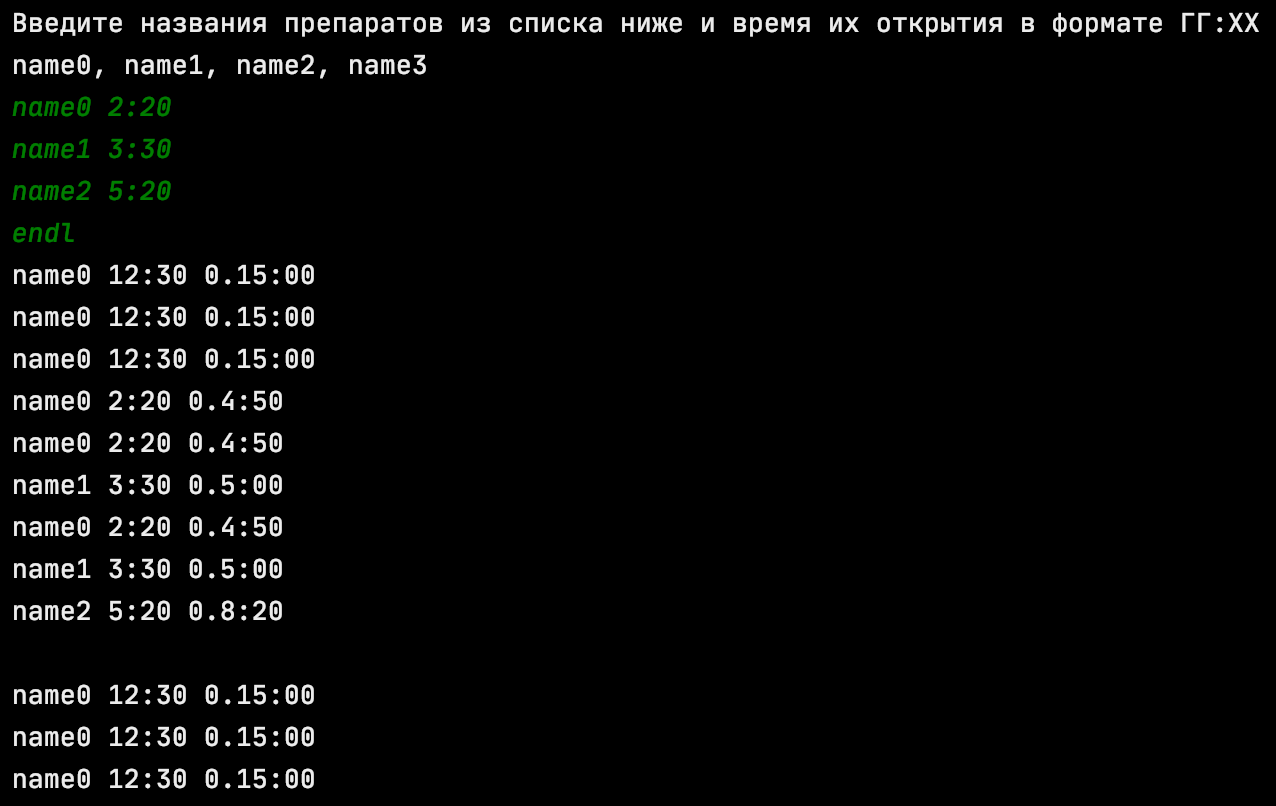
*Main.py:*

import funcs  
  
path = "";  
lines = ["name0 2:30", "name1 1:30", "name2 3:00", "name3 0:10"]  
funcs.FileWrite(lines, "{0}drugs.txt".format(path), "wb")  
print("Введите названия препаратов из списка ниже и время их открытия в формате ГГ:ХХ")  
drugsList = funcs.ListOfDrugs("{0}drugs.txt".format(path))  
print(", ".join(drugsList))  
ss = ""  
z = " ".join(funcs.FileRead("{0}drugs.txt".format(path))).split(" ")  
c = []  
while (ss != "endl"):  
 ss = input()  
 if (ss == "endl"):  
 continue  
 x = funcs.FindEndTime(ss.split(" ")[1], z[z.index(ss.split()[0]) + 1])  
 c.append("{0} {1} {2}".format(ss.split(" ")[0], ss.split(" ")[1], x))  
 funcs.FileWrite(c, "{0}openDrugs.txt".format(path), "ab")  
funcs.TxtOut(funcs.FileRead("{0}openDrugs.txt".format(path)))  
funcs.RemoveOverdue("{0}openDrugs.txt".format(path))  
funcs.TxtOut(funcs.FileRead("{0}openDrugs.txt".format(path)))

*funcs.py:*

from datetime import datetime, date, time  
import pickle  
  
  
def FileWrite(text, path, fb):  
 file = open(path, fb)  
 for i in text:  
 file.write((i + "\n").encode())  
  
  
def FileRead(path):  
 txt = []  
 file = open(path, "rb")  
 for text in file:  
 str = ""  
 str += text.decode()  
 txt.append(str[:-1])  
 return txt  
  
def TxtOut(ss):  
 for el in ss:  
 print(el)  
 print()  
  
  
def RemoveElement(elements, j):  
 newLines = []  
 k = 0  
 i = 0  
 while i < elements.Length:  
 if i != j:  
 newLines[k] = elements[i]  
 k += 1  
 i += 1  
 return newLines  
  
def FindEndTime(openTime, endTime):  
 openH = int(openTime.split(":")[0])  
 openM = int(openTime.split(":")[1])  
 endH = int(endTime.split(":")[0])  
 endM = int(endTime.split(":")[1])  
 days = 0  
 hours = (openH + endH)  
 minutes = (endM + openM)  
 if minutes >= 60:  
 minutes -= 60  
 hours += 1  
 if hours >= 24:  
 hours -= 24  
 days += 1  
 return "{}.{}:{}".format(days, hours, ("0" + str(minutes)) if len(str(minutes)) == 1 else minutes)  
  
  
  
def ListOfDrugs(path):  
 txt = FileRead(path)  
 list = []  
 i = 0  
 while i < len(txt):  
 list.append(txt[i].split(" ")[0])  
 i += 1  
 return list  
  
def ChangeDate(days, date):  
 now = date.strftime("%d/%m/%Y").split("/")  
 changedDay = (int(now[0]) + days)  
 now[0] = str(changedDay)  
 dt = datetime.strptime("/".join(now) + " {}:{}".format(date.hour, date.minute), "%d/%m/%Y %H:%M")  
 return dt  
  
def RemoveOverdue(path):  
 lines = FileRead(path)  
 i = 0  
 while i < len(lines):  
 s = lines[i].split(" ")[2]  
 ss = s.split(".")[1]  
 h = ss.split(":")[0]  
 x = lines[i].split(" ")[2]  
 xx = x.split(".")[1]  
 m = xx.split(":")[1]  
  
 dt = datetime.today()  
  
  
 curDate = datetime.combine(date(dt.year, dt.month, dt.day), time(int(h), int(m)))  
 endTime = ChangeDate(int(lines[i].split(" ")[2].split(".")[0]), curDate)  
 if endTime < dt:  
 del lines[i]  
 i -= 1  
 i += 1  
 FileWrite(lines, path, "wb")

*Результат:*

**

**Висновок:** під час виконання даної роботи ми вивчили особливості створення і обробки текстових файлів даних. Так само виконали за допомогою них завдання: Створили файл із назвами всіх ліків. дізналися коли закінчується термін деяких із них і видалили ті, що прострочилися”