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.

Методології програмування»

«Бінарні файли»

Варіант 3

Виконав студент ІП-11 Веремчук Ігор Ігорович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2022

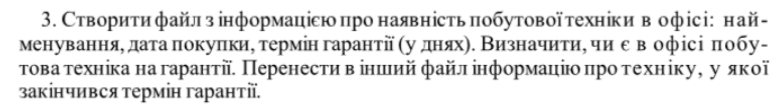
**Лабораторна робота №2**

**Бінарні файли**

**Мета**

****

**Завдання(варіант 3):**



**Постановка задачі**

Для виконання поставленої задачі створимо допоміжні файли (та структуру на С#) з методами/функціями для роботи з файлами, текстом та консольним вводом.

**Програма на С#**

Файл 1:

using System;  
using System.Collections.Generic;  
using System.Text;  
using System.IO;  
  
namespace Lab2Sharp  
{  
 internal static class Program  
 {  
 static void Main(string[] args)  
 {  
 FileManager fileManager = new FileManager();  
 GadgetManager gadgetManager = new GadgetManager();  
 const string **pathToFile1** = @"D:\BasicsOfProgramming(OOP)\Lab2\Lab2Sharp\Lab2Sharp\lab\_files\gadgets";  
 const string **pathToFile2** =  
 @"D:\BasicsOfProgramming(OOP)\Lab2\Lab2Sharp\Lab2Sharp\lab\_files\not\_warranty\_gadgets";  
  
 fileManager.FillFileFromConsole(**pathToFile1**);  
  
 var gadgets = gadgetManager.GetGadgetsFromFile(**pathToFile1**);  
 var warrantyGadgets = gadgetManager.GetNotWarrantyGadgets(gadgets);  
  
 fileManager.CreateFile(**pathToFile2**, GadgetManager.ConvertGadgetsToText(warrantyGadgets));  
  
 fileManager.PrintAllFile(**pathToFile1**);  
 fileManager.PrintAllFile(**pathToFile2**);  
 Console.ReadLine();  
 }  
 }  
}

Файл 2:

using System;  
using System.Collections.Generic;  
using System.Text;  
using System.IO;  
  
namespace Lab2Sharp  
{  
 internal class FileManager  
 {  
 public void CreateFile(string pathToFile) *//створення файлу* {  
 FileStream fileStream = new FileStream(pathToFile, FileMode.**Create**);  
 fileStream.Close();  
 }  
  
 public void CreateFile(string pathToFile, string text) *//створення файлу* {  
 FileStream fileStream = new FileStream(pathToFile, FileMode.**Create**);  
 byte[] array = Encoding.Default.GetBytes(text);  
 fileStream.Write(array, 0, array.Length);  
 fileStream.Close();  
 }  
  
 public void AppendToFile(string pathToFile, string text) *//доповнення файлу* {  
 FileStream fileStream = new FileStream(pathToFile, FileMode.**Append**);  
 byte[] array = Encoding.Default.GetBytes(text + "\n");  
 fileStream.Write(array, 0, array.Length);  
 fileStream.Close();  
 }  
  
 public static string ReadAllFromFile(string pathToFile) *//зчитування з файлу* {  
 FileStream fileStream = new FileStream(pathToFile, FileMode.**Open**);  
 byte[] array = new byte[fileStream.Length];  
 fileStream.Read(array, 0, array.Length);  
 string textFromFile = Encoding.Default.GetString(array);  
 fileStream.Close();  
 return textFromFile;  
 }  
  
 public void PrintAllFile(string pathToFile) *//вивід з файлу* {  
 Console.WriteLine($"\nFile name is: {pathToFile.Substring(pathToFile.LastIndexOf('\\') + 1)}");  
 string[] text = ReadAllFromFile(pathToFile).Split('\n', StringSplitOptions.**RemoveEmptyEntries**);  
 for (int i = 0; i < text.Length; i++)  
 {  
 if ((i + 1) % 3 == 0)  
 {  
 Console.WriteLine("Gadget");  
 Console.WriteLine($"Name: {text[i - 2]}");  
 Console.WriteLine($"Purchase date: {text[i - 1]}");  
 Console.WriteLine($"Warranty time: {text[i]}");  
 Console.WriteLine();  
 }  
 }  
 }  
  
 public void FillFileFromConsole(string pathToFile) *//заповнення файлу з консолі* {  
 Console.Write("Enter number of gadgets: ");  
 int gadgetCount = Convert.ToInt32(Console.ReadLine());  
 var gadgets = new List<Gadget>();  
 Gadget gadget = new Gadget();  
  
 for (int i = 0; i < gadgetCount; i++)  
 {  
 Console.Write("Enter the name of gadget: ");  
 string? inputName = Console.ReadLine();  
 Console.Write("Enter gadget's date of purchase (dd.mm.yyyy): ");  
 bool correctDate = DateTime.TryParse(Console.ReadLine(), out gadget.PurchaseDate);  
 Console.Write("Enter gadget's warranty time (days): ");  
 bool correctWarranty = uint.TryParse(Console.ReadLine(), out gadget.WarrantyTime);  
  
 if (inputName != null && correctDate && correctWarranty)  
 {  
 gadget.Name = inputName;  
 gadgets.Add(gadget);  
 }  
  
 Console.WriteLine();  
 }  
  
 CreateFile(pathToFile, GadgetManager.ConvertGadgetsToText(gadgets.ToArray()));  
 }  
 }  
}

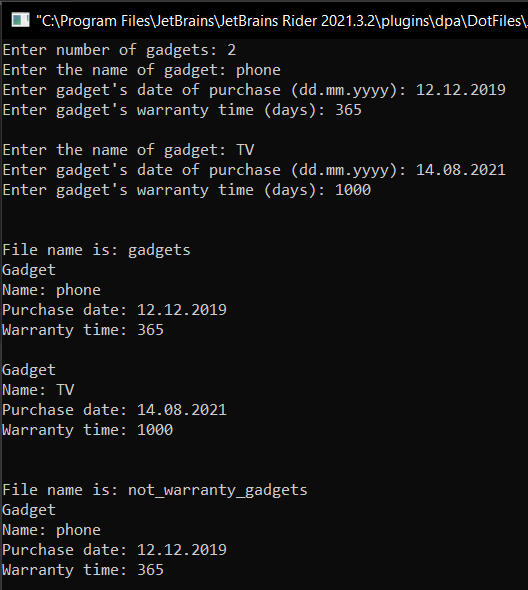
Файл 3:

using System;  
using System.Collections.Generic;  
using System.Text;  
using System.IO;  
  
namespace Lab2Sharp  
{  
 internal class GadgetManager  
 {  
 public Gadget[] GetGadgetsFromFile(string pathToFile) *//зчитування інформації про гаджети з файлу* {  
 string[] textFromFile =  
 FileManager.ReadAllFromFile(pathToFile).Split('\n', StringSplitOptions.**RemoveEmptyEntries**);  
 var gadgets = new List<Gadget>();  
 Gadget gadget = new Gadget();  
  
 for (int i = 0; i < textFromFile.Length; i++)  
 {  
 if ((i + 1) % 3 == 0)  
 {  
 gadget.Name = textFromFile[i - 2];  
 gadget.PurchaseDate = DateTime.Parse(textFromFile[i - 1]);  
 gadget.WarrantyTime = uint.Parse(textFromFile[i]);  
 gadgets.Add(gadget);  
 }  
 }  
  
 return gadgets.ToArray();  
 }  
  
 public Gadget[] GetNotWarrantyGadgets(Gadget[] gadgets) *//отримати гаджети не на гарантії* {  
 DateTime currentTime = DateTime.Today;  
 return gadgets.Where(gadget => gadget.PurchaseDate.AddDays(gadget.WarrantyTime) < currentTime).ToArray();  
 }  
  
 public static string ConvertGadgetsToText(Gadget[] gadgets) *//перетворити інформацію про гаджети в текст* {  
 string text = string.Empty;  
 foreach (Gadget gadget in gadgets)  
 {  
 text += $"{gadget.Name}\n{gadget.PurchaseDate.ToShortDateString()}\n{gadget.WarrantyTime}\n";  
 }  
  
 return text;  
 }  
 }  
}

Файл 4:

using System;  
  
namespace Lab2Sharp  
{  
 internal struct Gadget  
 {  
 public string Name;  
 public DateTime PurchaseDate;  
 public uint WarrantyTime;  
 }  
}

**Виконання на C#**

****

**Програма на python**

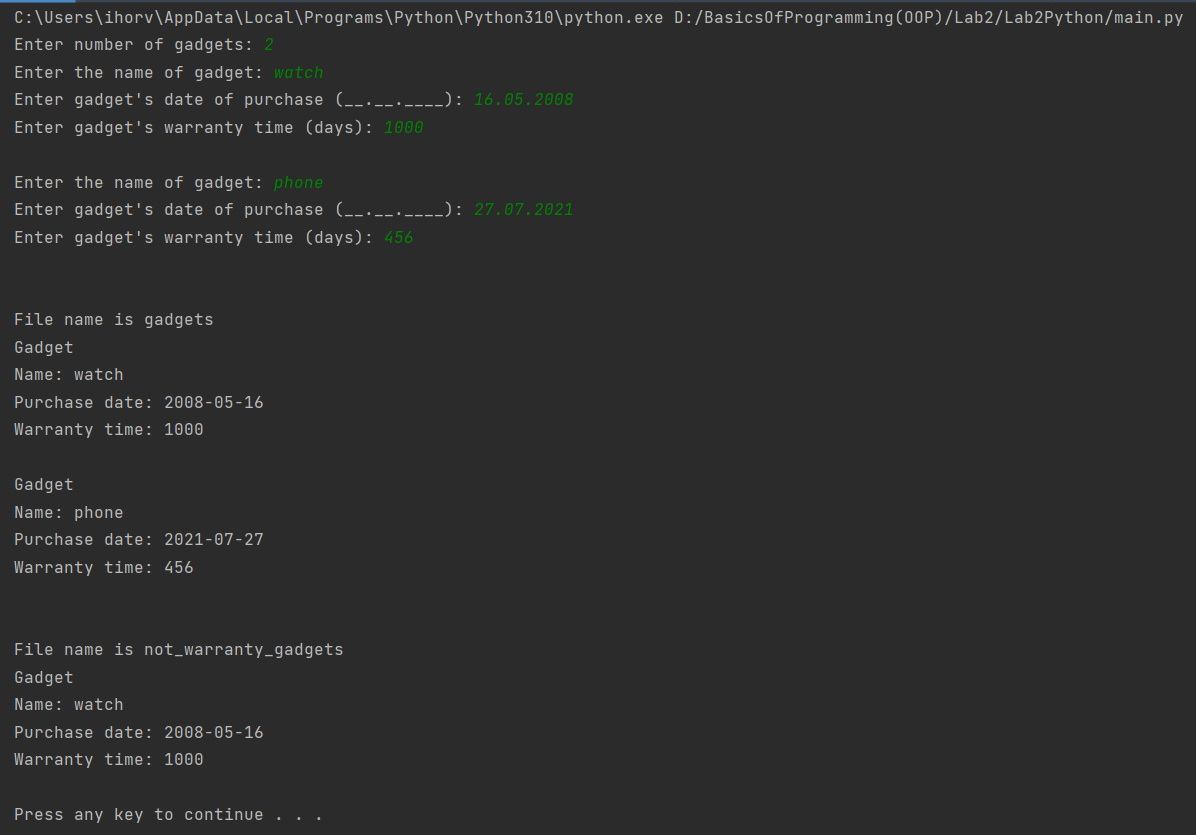
Файл 1:

from os import system  
import file\_manager  
  
  
def main():  
 path1 = "D:\\BasicsOfProgramming(OOP)\\Lab2\\Lab2Python\\lab\_files\\gadgets"  
 path2 = "D:\\BasicsOfProgramming(OOP)\\Lab2\\Lab2Python\\lab\_files\\not\_warranty\_gadgets"  
 file\_manager.fill\_file\_from\_console(path1)  
  
 gadgets = file\_manager.get\_gadgets\_from\_file(path1)  
 warranty\_gadgets = file\_manager.get\_not\_warranty\_gadgets(gadgets)  
  
 file\_manager.create\_file(path2, file\_manager.convert\_gadgets\_to\_text(warranty\_gadgets))  
  
 file\_manager.print\_all\_file(path1)  
 file\_manager.print\_all\_file(path2)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()  
 system('pause')

Файл 2:

import datetime  
from datetime import timedelta  
  
  
def create\_file(path, text: str): # створення файлу  
 with open(path, "wb") as file:  
 byte\_text = str.encode(text)  
 file.write(byte\_text)  
  
  
def append\_to\_file(path, text: str): # доповнення файлу  
 with open(path, "ab") as file:  
 byte\_text = str.encode(text)  
 file.write(byte\_text)  
  
  
def read\_all\_from\_file(path) -> str: # зчитування з файлу  
 byte\_array = []  
 with open(path, "rb") as file:  
 byte\_array = file.read()  
 return byte\_array.decode()  
  
  
def print\_all\_file(path: str): # вивід з файлу  
 file\_name = path[path.rindex("\\") + 1:]  
 print(f"\nFile name is {file\_name}")  
 text = read\_all\_from\_file(path).split('\n')  
 for i in range(len(text)):  
 if (i + 1) % 3 == 0:  
 print("Gadget")  
 print(f"Name: {text[i - 2]}")  
 print(f"Purchase date: {text[i - 1]}")  
 print(f"Warranty time: {text[i]}")  
 print()  
  
  
def fill\_file\_from\_console(path): # заповнення файлу з консолі  
 gadget\_count = int(input("Enter number of gadgets: "))  
 gadgets = []  
 for i in range(gadget\_count):  
 gadget\_name = input("Enter the name of gadget: ")  
 input\_date = input("Enter gadget's date of purchase (\_\_.\_\_.\_\_\_\_): ")  
 purchase\_date = datetime.datetime.strptime(input\_date, "%d.%m.%Y").date()  
 warranty\_time = int(input("Enter gadget's warranty time (days): "))  
 gadgets.append((gadget\_name, purchase\_date, warranty\_time))  
 print()  
  
 create\_file(path, convert\_gadgets\_to\_text(gadgets))  
  
  
def get\_gadgets\_from\_file(path) -> list[tuple]: # зчитування інформації про гаджети з файлу  
 text\_from\_file = read\_all\_from\_file(path).split('\n')  
 gadgets = []  
  
 for i in range(len(text\_from\_file)):  
 if (i + 1) % 3 == 0:  
 gadget\_name = text\_from\_file[i - 2]  
 purchase\_date = datetime.datetime.strptime(text\_from\_file[i - 1], "%Y-%m-%d").date()  
 warranty\_time = int(text\_from\_file[i])  
 gadgets.append((gadget\_name, purchase\_date, warranty\_time))  
  
 return gadgets  
  
  
def get\_not\_warranty\_gadgets(gadgets: list[tuple]) -> list[tuple]: # отримати гаджети не на гарантії  
 return [gadget for gadget in gadgets if gadget[1] + timedelta(gadget[2]) < datetime.datetime.today().date()]  
  
  
def convert\_gadgets\_to\_text(gadgets: list[tuple]) -> str: # перетворити інформацію про гаджети в текст  
 text = ''  
 for gadget in gadgets:  
 text += f"{gadget[0]}\n{gadget[1]}\n{gadget[2]}\n"  
 return text

**Виконання на python**



**Висновок:**

На лабораторній роботі я опанував технологію роботи з бінарними файлами. Програмного їх створення, запису та зчитування тексту з них, поставлене завдання було виконано на мовах C# та python.