

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 №1 з дисципліни
«Основи програмування»
«Бінарні файли»
Варіант 33

Виконав студент ІП-1134 Шамков Іван Дмитрович
(прізвище, ім'я, по батькові)

Перевірив викладач Вітковська Ірина Іванівна
(прізвище, ім'я, по батькові)

Київ 2022
Лабораторна робота №1
Бінарні файли

Лабораторна робота 1

Бінарні файли

Мета – вивчити особливості створення і обробки бінарних файлів.

Варіант: 33

Умова задачі:

33. Створити файл із списком творів автора: назва, дата написання, рік видання (якщо твір не видано, то - 0). Визначити кількість зимових творів автора. Перенести в окремий файл інформацію про твори, написані та видані в минулому столітті.

Постановка задачі:

Отже, математичне формулювання нашої задачі полягає в тому, щоб отримати від користувача дані про книгу(назва, дата написання, рік видання), а потім одразу її записати в бінарний файл(запис проводимо за допомогою функції `write((char*)&name, sizeof(type_of_name))`). Після цього ми працюємо з новоствореним файлом наступним шляхом: спочатку беремо дату написання записаних у файл книг, перевіряємо чи є місяць зимовим(12, 01, 02) та рахуємо їхню кількість; потім знову читаємо файл, перевіряємо тепер, чи починається рік видання книги на "19". Якщо так, то переносимо його до новоствореного файлу.

Текст файла проекту:

Python:

```
main.py × Book.py × createtxt.py × readtxt.py × countwinter.py × lastcentury.py ×
1 from createtxt import create_txt
2     from readtxt import read_txt
3     from countwinter import count_winter
4     from lastcentury import last_century
5
6 if __name__ == '__main__':
7     name1 = "File.txt"
8     name2 = "Last century.txt"
9     create_txt(name1)
10    read_txt(name1)
11    print("Кількість творів виданих узимку: " + str(count_winter(name1)))
12    last_century(name1, name2)
13    read_txt(name2)
14
```

```
main.py × Book.py × createtxt.py × readtxt.py × countwinter.py × lastcentury.py ×
1 class Book:
2     """Дані про книгу, її ім'я, дата написання та рік видання"""
3     def __init__(self, name="", date="", year=""):
4         self.name = name
5         self.date = date
6         self.year = year
7
8     def write(self):
9         print("Назва книги: " + self.name +
10              "\nДата написання книги: " + self.date +
11              "\nРік видання книги: " + self.year + "\n")
12
```

```

main.py × Book.py × createtxt.py × readtxt.py × countwinter.py × lastcentury.py ×
1 from Book import Book
2 from readtxt import get_previous_info
3 import pickle
4
5
6 def create_txt(name: str):
7     # Оформлення перезаписування та доповнення файлу
8     # Доповнюємо його через повний перезапис, що зроблено через особливість модулю pickle
9     answer = str(input("Очистити файл?(так/ні): "))
10    if answer == "так":
11        array = []
12    else:
13        array = get_previous_info(name)
14
15    file_write = open(name, "wb")
16
17    if file_write.closed:
18        print("\n_____Файл " + str(name) + " не вдалося відкрити для запису_____\\n")
19    else:
20        print("\n_____Файл " + str(name) + " відкрито для запису_____\\n")
21
22    num = int(input("Вкажіть кількість книг для запису: "))
23    for i in range(num):
24        # book = Book() створює кожний раз у циклі порожній об'єкт класу Book
25        # Якщо написати цей рядок до циклу, то в наш масив ми додаватимемо один і той самий елемент
26        book = Book()
27        book_name = str(input("\nВкажіть ім'я книги: "))
28        book.name = book_name
29
30        book_date = str(input("Вкажіть дату написання книги(00.00): "))

```

```

main.py × Book.py × createtxt.py × readtxt.py × countwinter.py × lastcentury.py ×
17 if file_write.closed:
18     print("\n_____Файл " + str(name) + " не вдалося відкрити для запису_____\\n")
19 else:
20     print("\n_____Файл " + str(name) + " відкрито для запису_____\\n")
21
22     num = int(input("Вкажіть кількість книг для запису: "))
23     for i in range(num):
24         # book = Book() створює кожний раз у циклі порожній об'єкт класу Book
25         # Якщо написати цей рядок до циклу, то в наш масив ми додаватимемо один і той самий елемент
26         book = Book()
27         book_name = str(input("\nВкажіть ім'я книги: "))
28         book.name = book_name
29
30         book_date = str(input("Вкажіть дату написання книги(00.00): "))
31         book.date = book_date
32
33         book_year = str(input("Вкажіть рік видання книги(1111): "))
34         book.year = book_year
35         array.append(book)
36
37         pickle.dump(array, file_write)
38
39     print("\n_____Файл " + str(name) + " закрито_____\\n")
40     file_write.close()
41

```

```

1 import Book
2 import pickle
3
4
5 def get_previous_info(name: str):
6     """
7     Функція для отримання попередньої інформації з файлу
8     та його перезапису нею одним блоком, що викликано особливістю
9     роботи з модулем pickle
10    """
11    with open(name, "rb") as file_read:
12        array = pickle.load(file_read)
13    return array
14
15
16 def read_txt(name: str):
17     file_read = open(name, "rb")
18     if file_read.closed:
19         print("\n_____Файл " + str(name) + " не вдалося відкрити для читання_____\\n")
20     else:
21         print("\n_____Файл " + str(name) + " відкрито для читання_____\\n")
22
23         array = pickle.load(file_read)
24         for book in array:
25             book.write()
26
27         print("\n_____Файл " + str(name) + " закрито_____\\n")
28     file_read.close()

```

```

1 import pickle
2 import Book
3
4
5 def count_winter(name: str):
6     with open(name, "rb") as file_read:
7         print("\n_____Файл " + str(name) + " відкрито для підрахунку творів, написаних узимку_____\\n")
8         array = pickle.load(file_read)
9         count = 0
10        for book in array:
11            date = book.date[3:]
12            if date == "12" or date == "01" or date == "02":
13                count += 1
14        print("\n_____Файл " + str(name) + " закрито для підрахунку творів, написаних узимку_____\\n")
15    return count
16

```

```
main.py × Bookpy × createtxt.py × readtxt.py × countwinter.py × lastcentury.py ×
1 import pickle
2 import Book
3
4
5 def last_century(name_read, name_write):
6     file_read = open(name_read, "rb")
7     file_write = open(name_write, "wb")
8
9     if file_read.closed and file_write.closed:
10         print("\n_____Файли " + str(name_read) + " та " + str(name_write) + " не вдалося відкрити_____\\n")
11     else:
12         print("\n_____Файли " + str(name_read) + " та " + str(name_write) +
13             " відкрито для пошуку творів минулого сторіччя_____\\n")
14
15         array_read = pickle.load(file_read)
16         array_write = []
17         # Пошук у масиві книг, що були видані минулого сторіччя
18         for book in array_read:
19             if book.year[:2] == "19":
20                 array_write.append(book)
21         pickle.dump(array_write, file_write)
22
23         print("\n_____Файли " + str(name_read) + " та " + str(name_write) + " закрито_____\\n")
24     file_read.close()
25     file_write.close()
26
```

C++:

```
1  #include <Windows.h>
2  #include <iostream>
3  #include "createtxt.h"
4  #include "readtxt.h"
5  #include "countwinter.h"
6  #include "lastcentury.h"
7
8  int main() {
9      SetConsoleCP(1251);
10     SetConsoleOutputCP(1251);
11     create_txt("File.txt");
12     read_txt("File.txt");
13     cout<< "\nКількість творів, написаних узимку: " << count_winter("File.txt") << "\n";
14     last_century("File.txt", "Last century.txt");
15     read_txt("Last century.txt");
16     return 0;
17 }
```

Bookh x lastcentury.h countwinter.cpp readtxt.cpp

Lab 2 (Global Scope)

```
1  #pragma once
2  #include <string>
3  using std::string;
4
5
6  struct Book {
7      char name[255];
8      char date[6];
9      char year[5];
10 };
11
12
```

Bookh lastcentury.h countwinter.cpp x readtxt.cpp lastcentury.cpp countwinter.h readtxt.h createtxt.cpp createtxt.h Main.cpp

Lab 2 (Global Scope) count_winter(string name)

```
1  #include "countwinter.h"
2
3  int count_winter(string name) {
4      int count = 0;
5      ifstream file_read;
6      file_read.open(name, ios::binary);
7
8      if (!file_read.is_open()) {
9          cout << "\nФайл " << name << " не вдалося відкрити для підрахунку творів, написаних узимку\n";
10     }
11     else {
12         cout << "\nФайл " << name << " відкрито для підрахунку творів, написаних узимку\n";
13         Book temp;
14         while (file_read.read((char*)&temp, sizeof(Book))) {
15             string date = temp.date;
16             date = date.substr(3, 2);
17             if (date == "12" || date == "01" || date == "02") {
18                 count++;
19             }
20         }
21
22         cout << "\nФайл " << name << " закрито\n";
23     }
24
25     file_read.close();
26     return count;
27 }
28
```

```
Bookh lastcentury.h countwinter.cpp readtxt.cpp lastcentury.cpp countwinter.h readtxt.h createtxt.cpp createtxt.h Main.cpp
Lab 2 (Global Scope) read_txt(string name)
1 #include "readtxt.h"
2
3 void read_txt(string name) {
4     ifstream file_read;
5     file_read.open(name, ios::binary);
6
7     if (!file_read.is_open()) {
8         cout << "\n_____Файл " << name << " не вдалося відкрити для читання_____\n";
9     }
10    else {
11        cout << "\n_____Файл " << name << " відкрито для читання_____\n";
12        Book temp;
13        while (file_read.read((char*)&temp, sizeof(Book))) {
14            cout << "\nНазва книги: "<< temp.name << "\nДата написання книги: "<< temp.date << "\nPік видання книги: "<< temp.year << "\n";
15        }
16        cout << "\n_____Файл " << name << " закрито_____\n";
17    }
18
19    file_read.close();
20 }
```

```
Bookh lastcentury.h countwinter.cpp readtxt.cpp lastcentury.cpp countwinter.h readtxt.h createtxt.cpp createtxt.h Main.cpp
Lab 2 (Global Scope) count_winter(string name)
1 #include "countwinter.h"
2
3 int count_winter(string name) {
4     int count = 0;
5     ifstream file_read;
6     file_read.open(name, ios::binary);
7
8     if (!file_read.is_open()) {
9         cout << "\n_____Файл " << name << " не вдалося відкрити для підрахунку творів, написаних узимку_____\n";
10    }
11    else {
12        cout << "\n_____Файл " << name << " відкрито для підрахунку творів, написаних узимку_____\n";
13        Book temp;
14        while (file_read.read((char*)&temp, sizeof(Book))) {
15            string date = temp.date;
16            date = date.substr(3, 2);
17            if (date == "12" || date == "01" || date == "02") {
18                count++;
19            }
20        }
21
22        cout << "\n_____Файл " << name << " закрито_____\n";
23    }
24
25    file_read.close();
26    return count;
27 }
28 }
```

```
Bookh lastcentury.h countwinter.cpp readtxt.cpp lastcentury.cpp countwinter.h readtxt.h createtxt.cpp createtxt.h Main.cpp
Lab 2 (Global Scope) last_century(string name_read, string name_write)
1 #include "lastcentury.h"
2
3 void last_century(string name_read, string name_write) {
4     ifstream file_read;
5     ofstream file_write;
6     file_read.open(name_read, ios::binary);
7     file_write.open(name_write, ios::binary);
8
9     if (!file_read.is_open() && !file_write.is_open()) {
10        cout << "\n_____Файли " << name_read << " та " << name_write << " не вдалося відкрити для підрахунку творів, виданих минулого століття_____\n";
11    }
12    else {
13        cout << "\n_____Файли " << name_read << " та " << name_write << " відкрито для підрахунку творів, виданих минулого століття_____\n";
14        Book temp;
15
16        while (file_read.read((char*)&temp, sizeof(Book))) {
17            string year = temp.year;
18            year = year.substr(0, 2);
19            if (year == "19") {
20                file_write.write((char*)&temp, sizeof(Book));
21            }
22        }
23
24        cout << "\n_____Файли " << name_read << " та " << name_write << " закрито_____\n";
25    }
26
27    file_read.close();
28    file_write.close();
29    return;
30 }
31
32 }
```


Копії екранних форм:

C++:

```
Очистити файл Main.txt?(так/ні) так
```

```
_____Файл File.txt відкрито для запису_____
```

```
Вкажіть кількість книг для запису: 3
```

```
Вкажіть ім'я книги: Сойчине крило
```

```
Вкажіть дату написання книги(00.00): 12.02
```

```
Вкажіть рік видання книги(1111): 2003
```

```
Вкажіть ім'я книги: Захар Беркут
```

```
Вкажіть дату написання книги(00.00): 14.08
```

```
Вкажіть рік видання книги(1111): 1919
```

```
Вкажіть ім'я книги: Фарбований Лис
```

```
Вкажіть дату написання книги(00.00): 27.12
```

```
Вкажіть рік видання книги(1111): 1999
```

```
_____Файл File.txt закрито_____
```

```
_____Файл File.txt відкрито для читання_____
```

```
Назва книги: Сойчине крило
```

```
Дата написання книги: 12.02
```

```
Рік видання книги: 2003
```

```
Назва книги: Захар Беркут
```

```
Дата написання книги: 14.08
```

```
Рік видання книги: 1919
```

```
Назва книги: Фарбований Лис
```

```
Дата написання книги: 27.12
```

```
Рік видання книги: 1999
```

```
_____Файл File.txt закрито_____
```

Файл File.txt відкрито для підрахунку творів, написаних узимку

Файл File.txt закрито

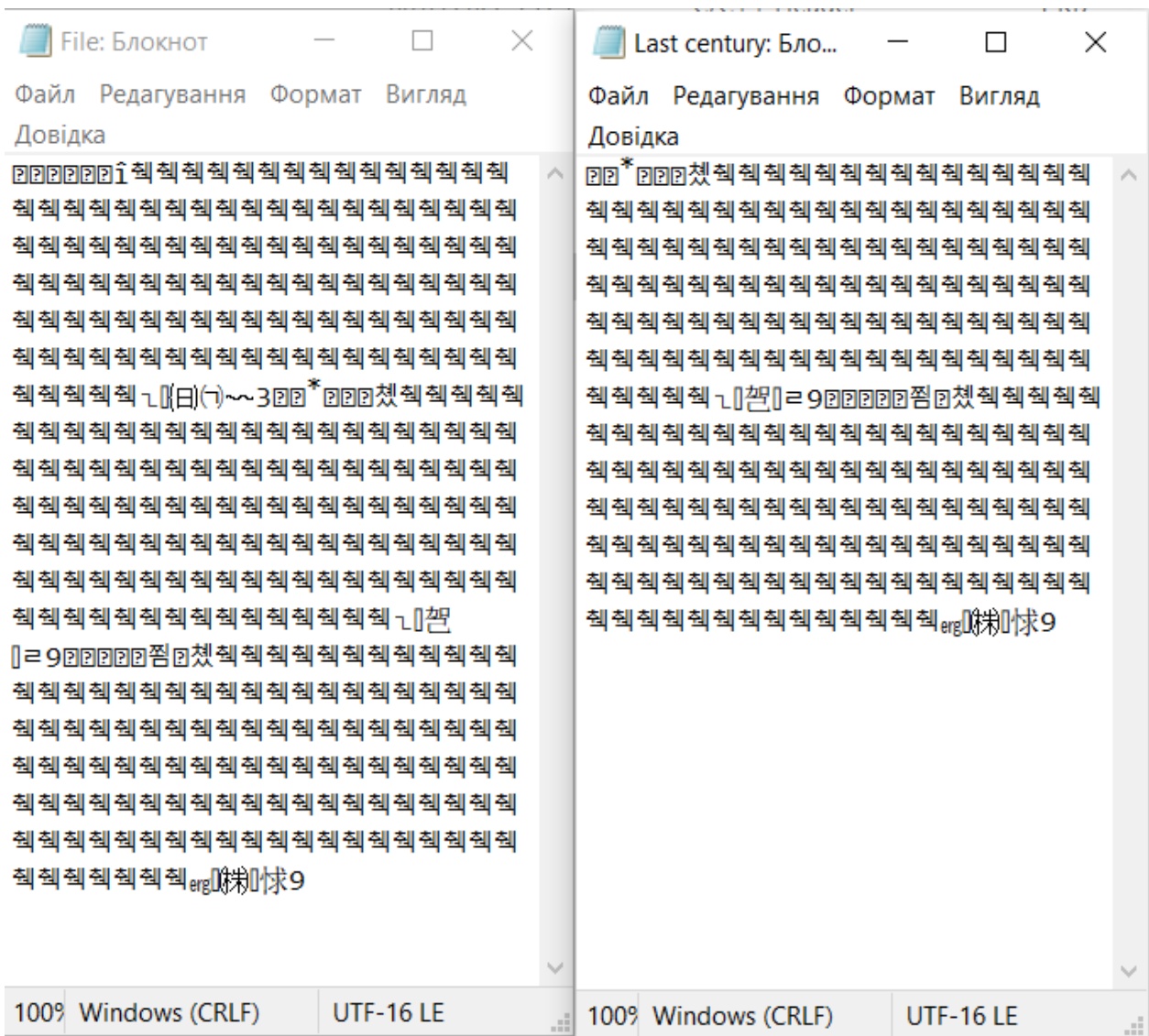
Кількість творів, написаних узимку: 2

Файл Last century.txt відкрито для читання

Назва книги: Захар Беркут
Дата написання книги: 14.08
Рік видання книги: 1919

Назва книги: Фарбований Лис
Дата написання книги: 27.12
Рік видання книги: 1999

Файл Last century.txt закрито



Python:

```
-----Файл File.txt відкрито для запису-----

Вкажіть кількість книг для запису: 3

Вкажіть ім'я книги: Сойчине крило
Вкажіть дату написання книги(00.00): 12.12
Вкажіть рік видання книги(1111): 1850

Вкажіть ім'я книги: Захар Беркут
Вкажіть дату написання книги(00.00): 08.11
Вкажіть рік видання книги(1111): 1999

Вкажіть ім'я книги: Фарбований Лис
Вкажіть дату написання книги(00.00): 15.02
Вкажіть рік видання книги(1111): 1900

-----Файл File.txt закрито-----
```

```
-----Файл File.txt відкрито для читання-----

Назва книги: Сойчине крило
Дата написання книги: 12.12
Рік видання книги: 1850

Назва книги: Захар Беркут
Дата написання книги: 08.11
Рік видання книги: 1999

Назва книги: Фарбований Лис
Дата написання книги: 15.02
Рік видання книги: 1900

-----Файл File.txt закрито-----
```

```
-----Файл File.txt відкрито для підрахунку творів, написаних узимку-----

-----Файл File.txt закрито для підрахунку творів, написаних узимку-----

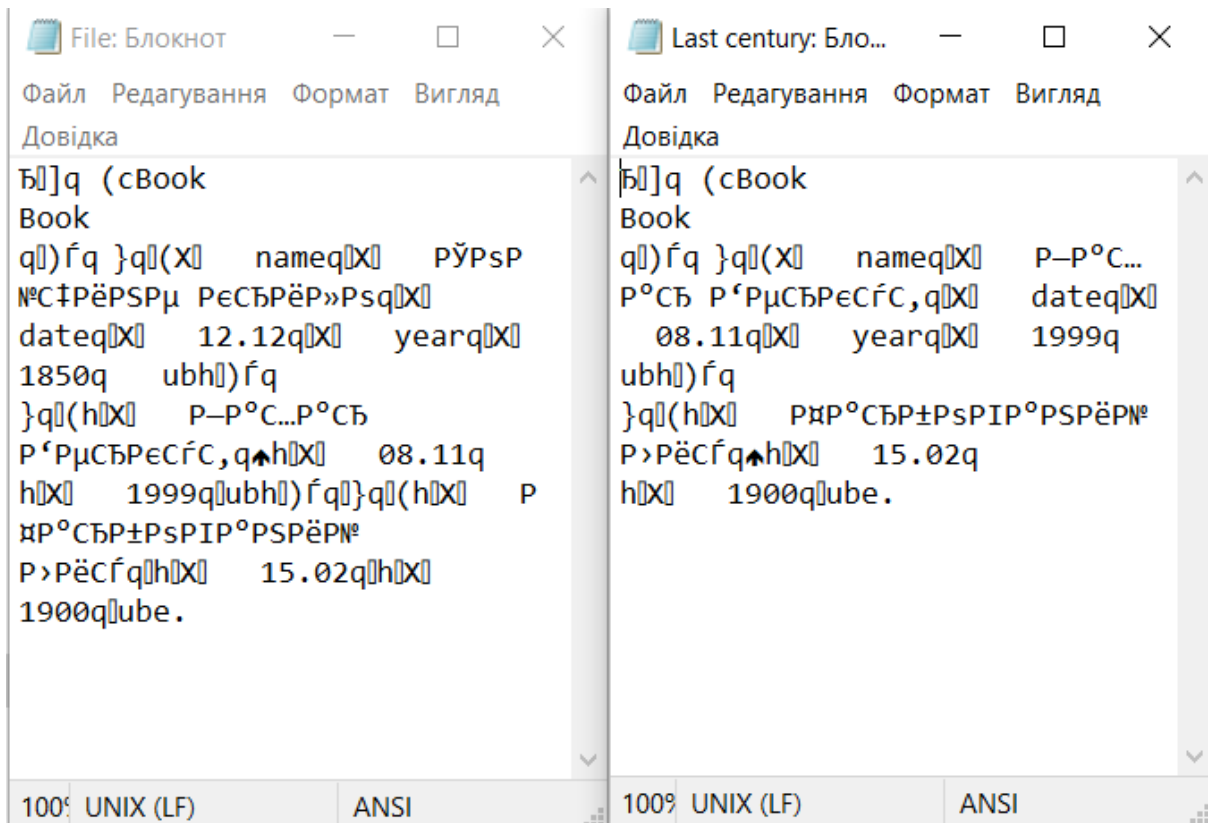
Кількість творів виданих узимку: 2
```

```
-----Файл Last century.txt відкрито для читання-----

Назва книги: Захар Беркут
Дата написання книги: 08.11
Рік видання книги: 1999

Назва книги: Фарбований Лис
Дата написання книги: 15.02
Рік видання книги: 1900

-----Файл Last century.txt закрито-----
```



Висновок

Отже, виконавши цю лабораторну роботу, ми навчилися працювати з бінарними файлами. Проекти, на мою думку, розроблені коректно, адже заплановані елементи працюють, а саме: отримання даних від користувача, запис цієї інформації у текстові документи, читання цих текстових документів, а також робота функцій підрахунку кількості творів, написаних узимку, та пошук творів, виданих минулого століття, з подальшим записом цих творів. При кожному відкритті текстового документа перевіряємо, чи відкритий він, щоб не отримати помилку. Для полегшення роботи з інформацією, а також через особливість роботи з бінарними файлами, була створена структура(клас у Python) під назвою Book, яка містить три поля: name, date, year. Так, ми можемо записати об'єкт структури Book у бінарний файл одним цілим, а потім його з легкістю прочитати та звернутися до потрібних нам даних об'єкту. Через особливість роботи з об'єктами типу string використовуємо масив типу char. Як і описували в постановці задачі, просто записуємо об'єкти типу Book в бінарний файл, потім його читаємо та виконуємо потрібні нам дії з пошуком певних творів та їх записом в інший бінарний файл.