

Лабораторна робота № 22

Тема. Двійкові файли та потоки.
Мета. Формування вмінь і навиків створення та використання двійкових файлів і потоків. Закріплення вмінь і навиків використання функцій вводу-виводу та обробки рядків. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання.

1. Коли та з якою метою доцільно використовувати двійкові файли?
2. У чому переваги та недоліки послідовних відносно типізованих двійкових файлів?
3. Для чого в класах обробки потоків використовується властивість *BaseStream*?
4. Як створити двійковий файл?
5. Як здійснюється зчитування/запис даних з двійкового файла?
6. Як в двійкових файлах зберігаються рядки? Від чого залежить їх розмір?
7. Як здійснюється перевірка кінця двійкового файла і коли вона застосовується?
8. Як закрити файл? Що відбувається на рівні операційної системи при закритті файла?

Завдання.

1. **Скласти дві програми: першу – для створення двійкового файла послідовного доступу, другу - для опрацювання даних, записаних у цей файл.**

Вимоги до програми:

- значення даних першої програми ввести з клавіатури з відповідною перевіркою коректності типів та даних;
- перед відкриттям файла в другій програмі для обробки передбачити перевірку існування файла. При відсутності файла видати відповідне повідомлення та завершити виконання програми;
- результати виконання другої програми вивести у вікні консольного додатку чи в одному діалоговому вікні.

Варіанти:

1. Протокол лижних гонок записати у файл *Ski.dat*. Для кожного учасника вводиться прізвище, час старту (години, хвилини, секунди) та час фінішу. Використовуючи сформований файл, вивести прізвища учасників, які виконали норматив;
2. Сформувати Файл *Book.dat* з прізвищ улюблених письменників учнів класу (учні по черзі вводять по три прізвища письменників). Використовуючи сформований файл, надрукувати прізвища п'яти найпопулярніших письменників;
3. Сформувати файл *Exam.dat* за результатами здачі сесії групою студентів. При цьому для кожного студента вводиться прізвище та три відповідні оцінки. Використовуючи сформований файл, вивести результати сесії у вигляді таблиці. У заголовку таблиці вивести назви предметів;
4. Задана розріджена матриця (розмірності не більш 10x10), яка складається з нулів та одиниць. Сформувати файл *Mlist.dat*, в якому запам'ятати матрицю в такому вигляді: кількість рядків, кількість стовпців, а далі – номери рядків та стовпців, на перетині яких знаходяться ненульові елементи. Використовуючи сформований файл, вивести матрицю на екран у звичному вигляді;
5. Задана розріджена матриця (розмірності не більш 10x10). Сформувати файл *Mlist.dat*, в якому запам'ятати матрицю в такому вигляді: кількість рядків, кількість стовпців, а далі – номери рядків, стовпців та значень ненульових елементів. Використовуючи сформований файл, вивести матрицю на екран у звичному вигляді;
6. Сформувати файл *Friend.dat* з прізвищ та дат (дня, місяця і року) народження Ваших друзів. Використовуючи сформований файл, вивести прізвища тих, хто народився влітку;
7. Сформувати файл *Name.dat*, в якому зберегти список імен. Використовуючи сформований файл, вивести імена, що починаються з букви *A*;
8. Сформувати файл *Stud.dat* з наступною структурою: прізвище студента, стать (одна буква – *Ч* або *Ж*), рік народження. Вивести список студентів чоловічої статі, вказавши їх вік. Після списку вивести середній вік відібраних студентів;
9. Сформувати файл *Car.dat*, що має містити інформацію про автолюбителів у форматі: прізвище водія, марка автомобіля, колір. Використовуючи сформований файл, вивести на екран дисплея відомості про автолюбителів, що мають автомобіль *Жигулі* зеленого кольору;
10. Сформувати файл *Stud.dat* з даними студентів одного курсу (прізвище, стать, рік та місяць народження). Використовуючи сформований файл, вивести прізвища студентів чоловічої статі, що народились влітку;
11. Сформувати файл *Rozklad.dat*, що містить інформацію про поїзди, які відправляються з залізничного вокзалу м. Києва (номер поїзда, станція призначення, час відправлення, час в дорозі). Використовуючи сформований файл, вивести інформацію про поїзди, які відправляються у ковельському напрямі від 20 до 24 годин;
12. Сформувати файл *Play.dat*, що має містити інформацію про побутові магнітофони у форматі: марка магнітофона та його вартість. Використовуючи сформований файл, вивести інформацію про магнітофони вартістю менше 500 гривень;

13. Сформувати файл *Book.txt*, який має містити дані про книги з програмування Вашою особистою бібліотеки (прізвище автора і його ініціали, назва книги, рік видання). Використовуючи сформований файл, вивести на екран дані останньої виданої книги;
14. Сформувати файл *Magazin.dat*, що має містити відомості про магазини Вашого району у форматі: тип товарів магазину (*Універмаг, Продукти, Тканини, ...*), його номер та адресу. Використовуючи сформований файл, вивести інформацію про всі магазини що займаються книгами;
15. Сформувати файл *Kub.dat*, що має містити відомості про кубики у форматі: розмір кожного кубика (довжина ребра в сантиметрах), його колір (червоний, жовтий, зелений чи синій) та матеріал (дерево, метал, картон). Знайти кількість дерев'яних кубиків з ребром 3 см і кількість металевих кубиків з ребром, більшим за 5 см.

Наприклад, програма для формування файлу завдання варіанту № 13 може мати вигляд:

```
Book[] masBook = new Book[] { new Book { name = "Изучаем С#", author = "Ендрю Стиллмен, Джениффер Грин" },
                              new Book("Кобзар", "Тарас Григорович Шевченко", 1840),
                              new Book(true), new Book(false), new Book(), book,book1,book2};
FileStream f = new FileStream("I:\\LR19.dat", FileMode.Create);
BinaryWriter w = new BinaryWriter(f,Encoding.UTF8);
int i;
for (i = 0; i < masBook.Count(); i++)
{
    w.Write(masBook[i].author+""");
    Console.WriteLine(w.BaseStream.Position);
    w.Write(masBook[i].name+""");
    Console.WriteLine(w.BaseStream.Position);
    w.Write(masBook[i].year);
    Console.WriteLine(w.BaseStream.Position);
}
w.Close();
```

А програма для розв'язання цього завдання може бути такою:

```
f = new FileStream("I:\\LR19.dat", FileMode.OpenOrCreate);
BinaryReader r = new BinaryReader(f, Encoding.UTF8);
int countZap=0;
while (r.BaseStream.Position<r.BaseStream.Length)
{r.ReadString();
 r.ReadString();
 r.ReadInt32();
 countZap++;
}
if (countZap == 0)
{Console.WriteLine("Дані книг відсутні");
 Console.ReadLine();
}
r.BaseStream.Seek(0,SeekOrigin.Begin); //позиціонуємо на початок в базовому потоці
masBook = new Book[countZap];
i = 0;
while (i < countZap)
{masBook[i].author=r.ReadString();
 masBook[i].name = r.ReadString();
 masBook[i].year = r.ReadInt32();
 i++;
}
r.Close();
Console.WriteLine("Перелік книг:");
foreach (Book cBook in masBook)
    cBook.Info();
int index = 0, lastYear = masBook[0].year;
for (i = 1; i < masBook.Count(); i++)
    if (masBook[i].year > lastYear)
    {
        lastYear = masBook[i].year;
        index = i;
    }
Console.Write("Серед них найсвіжіша: ");
masBook[index].Info();
Console.ReadLine();
```

2. Скласти програму для дозапису даних у двійковий файл послідовного доступу, створений в результаті виконання першої програми попереднього завдання. Після дозапису даних у файл знову виконати програму обробки його даних (тобто другу програму попереднього завдання). Порівняйте розміри файлів, створених програмами попередньої та цієї лабораторної роботи.