

Лабораторна робота № 21

- Тема.** Текстові файли та потоки.
Мета. Формування вмінь і навиків створення та використання текстових 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 файли в програмуванні?
2. Які типи файлів і потоків використовуються в C#? У чому переваги та недоліки кожного типу стосовно форматів записів та швидкості обробки?
3. Що відбувається на рівні операційної системи при відкритті файла?
4. Як створити текстовий файл?
5. Як здійснюється зчитування даних з текстового файла?
6. Як здійснюється перевірка кінця текстового файла і коли вона застосовується?
7. Як закрити файл? Що відбувається на рівні операційної системи при закритті файла?

Завдання.

1. **Скласти дві програми: першу – для створення 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.txt*. 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.txt* 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.txt* 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.txt*, 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.txt*, 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.txt* з прізвищ та дат (дня, місяця і року) народження Ваших друзів. Використовуючи сформований файл, вивести прізвища тих, хто народився влітку;
7. Сформувати файл *Name.txt*, в якому зберегти список імен. Використовуючи сформований файл, вивести імена, що починаються з букви *A*;
8. Сформувати файл *Stud.txt* 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.txt*, 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.txt* 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.txt*, 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.txt*, 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.txt*, 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.txt*, 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:

```

struct Book
{
    public string name;
    public string author;
    public int year;

    public Book(bool readData):this()
    {
        if (readData)
        {
            Console.WriteLine("Введіть назву: ");
            this.name = Console.ReadLine();
            Console.WriteLine("Введіть автора: ");
            this.author = Console.ReadLine();
            Console.WriteLine("Введіть рік виходу: ");
            this.year = Convert.ToInt32(Console.ReadLine());
        }
    }

    public Book(string name, string author, int year)
    {this.name =name;
    this.author=author;
    this.year=year;
    }

    public void Info()
    {
        Console.WriteLine("Книга '{0}' (автор(и) {1}) була видана в {2} році", name, author, year);
    }
}
...
Book[] masBook = new Book[] { new Book { name = "Изучаем С#", author = "Єндрю Стіллен, Дженифер Грин" },
    new Book("Кобзар", "Тарас Григорович Шевченко", 1840),
    new Book(true), new Book(false), new Book()};
StreamWriter f = new StreamWriter("I:\\LR18.txt",);
int i;
for (i = 0; i < masBook.Count(); i++)
{
    f.WriteLine(masBook[i].author);
    f.WriteLine(masBook[i].name);
    f.WriteLine(masBook[i].year.ToString());
}
f.Close();

```

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

```

StreamReader f1 = new StreamReader("I:\\LR18.txt");
int countLine=0;
while (!f1.EndOfStream)
{f1.ReadLine();
    countLine++;
}
if (countLine==0)
{
    Console.WriteLine("Дані книг відсутні");
    Console.ReadLine();
}

```

```

f1.BaseStream.Seek(0,SeekOrigin.Begin); //позиціонуємо на початок в базовому потоці
f1.DiscardBufferedData();//очищаємо проміжний буфер для коректного зчитування з початку
masBook = new Book[countLine / 3];
i = 0;
while (!f1.EndOfStream)
{masBook[i].author=f1.ReadLine();
masBook[i].name = f1.ReadLine();
masBook[i].year = Int32.Parse(f1.ReadLine());
i++;
}
f1.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 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).