**Лабораторная работа № 14**

**Разработка класса по работе с бинарными файлами  
 объектов данных**

***Задание 1.***

Разработать класс для работы с бинарным файлом объектов данных. Данные описать как поля класса. Методы класса должны обеспечивать:

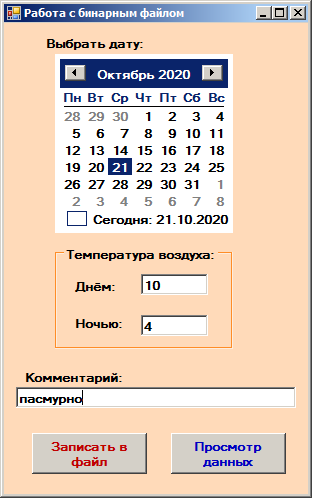
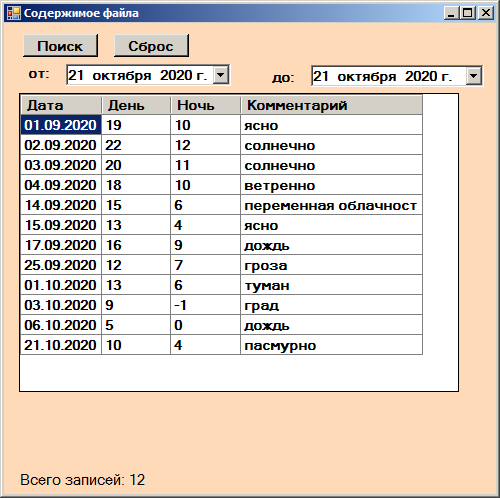
1. Добавление объекта в бинарный файл
2. Вывод всех данных из файла в таблицу
3. Поиск в бинарном файле записей, удовлетворяющих заданному критерию
4. Корректировку отдельной записи файла, начинающейся с заданной позиции (байта)

***Задание 2.***

Разработать многомодульный проект (класс формы 1 и класс формы 2), обеспечивающие следующие функции:

1. Программа должна записывать в бинарный файл данные, вводимые пользователем, формируя, таким образом, простую базу данных. Для ввода некоторых полей данных примените компонент «комбинированный список», для других – поле ввода или поле ввода по маске, для числовых данных небольшого диапазона можно использовать переключатель значений, для выбора даты используйте компонент календарь.
2. Программа должна предоставлять возможность откорректировать отдельную запись в бинарном файле, найденную по значению уникального поля.
3. Вывод данных из файла осуществлять с помощью второй формы, на которой содержимое файла должно выводиться в таблицу. Вторая форма открывается в немодальном режиме. Таблица на второй форме должна сразу же обновляться при добавлении или корректировке данных, без закрытия окна.
4. Кроме вывода содержимого всего файла, в ту же таблицу на второй форме выводить результаты запроса (выборочные записи) в соответствие с вашим заданием.

*Примерный вид Form1 и Form2:*

namespace Работа\_с\_бинарным\_файлом

{

[Serializable]

class Note: IComparable

{

DateTime date;

int t\_Day;

int t\_Naigh;

byte[] coment;

[NonSerialized]

const int MAX\_LENGTH = 20;

static Encoding encoding = System.Text.Encoding.GetEncoding(1251);

// потребуется для преобразования строки в byte[]

public Note(DateTime date, int t\_Day, int t\_Naigh, string coment) // конструктор

{

this.date = date;

this.t\_Day = t\_Day;

this.t\_Naigh = t\_Naigh;

this.coment = StringToByte(coment);

}

/// <summary>

/// преобразование из массива байт в строку

/// </summary>

/// <param name="ar">массив байт</param>

/// <returns>строка</returns>

private static string ByteArrayToString(byte[] b)

{

char[] buff = encoding.GetChars(b);

// раскодирование массива байт

string result = "";

for (int i = 0; i < buff.Length; i++)

{

result += (char)buff[i];

// добавление каждого символа в строку

}

return result;

}

/// <summary>

/// преобразование из строки в массив байт

/// </summary>

/// <param name="s">строка переменной длины</param>

/// <returns>массив байт фиксированной длины</returns>

private static byte[] StringToByte(string s)

{

char[] ar = new char[MAX\_LENGTH];

for (int i = 0; i < MAX\_LENGTH; i++)

{

if (i < s.Length)

ar[i] = s[i];

// преобразование в символьный массив фиксированного размера

else

ar[i] = ' ';

// дополнение строки пробелами до нужного размера

}

byte[] buff = encoding.GetBytes(ar); // кодирование в массив байт

return buff;

}

*Пример метода добавления данных в конец файла:*

public void Save\_Result(string filename)

{

FileStream fa = null;

try

{

fa = new FileStream(filename, FileMode.Append);

BinaryFormatter bw = new BinaryFormatter();

bw.Serialize(fa, this);

}

catch { }

finally { fa.Close(); }

}

*Пример метода корректировки отдельной записи (следует с байта* num*):*

public bool Correct\_Result(string filename, long num)

{

bool result = false;

if (num >= 0)

{

FileStream fw = null;

try

{

fw = new FileStream(filename, FileMode.Open);

BinaryFormatter bf = new BinaryFormatter();

fw.Seek(num, SeekOrigin.Begin);

// установим указатель файла на запись с позицией num

bf.Serialize(fw, this); // записываем в файл изменения

result = true;

}

catch

{

}

finally

{

fw.Close(); //закрытие потока записи

}

}

return result;

}

# ВАРИАНТЫ:

**ВАРИАНТ 4**

Создать структуру данных (класс) для библиотеки, имеющую следующие поля: инвентарный номер, автор, название книги, издательство, год издания, количество страниц.

Разработать следующие функции:

* добавление в файл объектов;
* вывод на экран содержимого всего файла;
* корректировки отдельной записи, найденной по содержимому главного поля;
* отбор записей по содержимому одного из полей и вывод на экран найденных записей;

Интерфейс программы оформить в виде меню.