

Основы программирования

Средства ввода-вывода

Двоичные потоки данных

Двоичные потоки данных

Байтовые (бинарные, двоичные) потоки позволяют читать и записывать данные в двоичном формате.

Эти байтовые потоки данных относятся к потокам-оберткам, то есть надстраиваются над любым потоком типа `Stream`.

Их основная функциональность – запись или чтение значений простых типов данных в двоичном виде в байтовый поток, над которым надстроен поток-обертка.

.

Запись в двоичный поток

В байтовых потоках запись и чтение производится побайтно.

Поскольку значение простого типа данных может занимать больше одного байта, то приходится выполнять преобразование значения к набору байтов и записывать их в байтовый поток.

При чтении реализуется обратное – чтение набора байт в необходимом количестве и конфигурирование из них многобайтного значения.

Эти алгоритмы могут реализовываться программистом. Для универсальности и корректности прямых и обратных преобразований используются спец. классы `BinaryWriter` и `BinaryReader`.

Класс BinaryWriter

СВОЙСТВО:

`BaseStream` – базовый поток, с которым работает `BinaryWriter`.

Методы:

`Close()` – закрывает поток и освобождает ресурсы.

`Flush()` – очищает буфер

`Seek()` – устанавливает курсор в потоке.

`Write()` – записывает данные в поток.

Класс BinaryReader

СВОЙСТВО:

`BaseStream` – базовый поток, с которым работает `BinaryWriter`.

Методы:

`Close()` – закрывает поток и освобождает ресурсы.

`Read()` – считывает байты из потока и перемещает курсор вперед.

`ReadByte()`, `ReadChar()`, `ReadDouble()` ...

Класс `BinaryReader`. Методы `Read...`

`ReadByte()` – считывает один байт и перемещает указатель на один байт.

`ReadChar()` – считывает значение `char`, то есть один символ, и перемещает указатель на столько байтов, сколько занимает символ в текущей кодировке.

`Read...()` – считывает значение типа `...` и перемещает указатель на соответствующее типу число байт.

`ReadString()` – считывает значение `string`. Каждая строка предваряется значением длины строки, которое представляет 7-битное целое число.

Пример

```
FileStream fs = new FileStream("my.dat",  
    FileMode.OpenOrCreate, FileAccess.ReadWrite,  
    FileShare.None);  
BinaryWriter bw = new BinaryWriter(fs);  
bw.WriteString("Строка в двоичном формате");
```

```
int a = 10;  
float b = 2.5;  
bool c = false;  
char[] chars = {'a', 'b', 'c'};
```

```
bw.Write(a);  
bw.Write(b);  
bw.Write(c);  
bw.Write(chars);  
bw.BaseStream.Position = 0;
```

Пример

```
BinaryReader br = new BinaryReader(fs);
int t = 0;
while (br.PeekChar() != -1) {
    Console.Write(br.ReadByte());
    t++;
    if (t == 5) {
        t = 0;
        Console.WriteLine(); }
//каждые 5 байт переводим курсор на новую строку
}

bw.Close();
br.Close();
fs.Close();
```

Пример

```
State[] states = new State[2];
states[0] = new State("Чехия", "Прага", 37168);
states[1] = new State("Франция", "Париж", 640679);
string path= @"C:\SomeDir\states.dat";
// создаем объект BinaryWriter
BinaryWriter bw = new
BinaryWriter(File.Open(path, FileMode.OpenOrCreate))
// записываем в файл значение
foreach (State s in states)
{
bw.Write(s.name);
bw.Write(s.capital);
bw.Write(s.area);
}
```

Пример

```
// создаем объект BinaryReader
BinaryReader br = new BinaryReader(File.Open(path,
FileMode.Open));
while (br.PeekChar() != -1)      {
string name = br.ReadString();
string capital = br.ReadString();
int area = br.ReadInt32();
Console.WriteLine("Страна: {0} столица:
{1} площадь {2} кв.км", name, capital, area);
}
bw.Close();
br.Close();
```

Пример. Комментарий

С помощью потока-обертки из байтового файла можно считать необходимое количество байт и автоматически сформировать значение искомого типа.

Не следует забывать, что чтение все равно производится побайтно, то есть считать можно совсем не ту информацию, которая была записана, если при чтении не знать порядок записи.

ССЫЛКИ, ИСТОЧНИКИ...

1. System.IO Namespace (Режим доступа <https://docs.microsoft.com/ru-ru/dotnet/api/system.io?view=netcore-2.1>).
2. C# и .NET | Бинарные файлы. BinaryWriter и BinaryReader (metanit.com) <https://metanit.com/sharp/tutorial/5.6.php>