

File Operations

...

Binary Operations

- For a stream, binary operations refer to writing/reading sequence of bytes to/from stream
- In text operation, it requires to convert data to characters (sequence of bytes) to write to stream, and convert characters read to data used in the application
- In binary operation, raw data will be written to array of bytes, and array of bytes read are put to memory location holding an variable/object.
- An instance of the class **BinaryWriter** writes data to an associating stream.

- Creating a BinaryWriter associating with a given writeable stream

+ Prototypes:

BinaryWriter(stream:Stream)

+ Example:

```
FileStream fileStream = new FileStream("example.data", FileMode.Create);
BinaryWriter binaryWriter = new BinaryWriter(fileStream);
```

- Writing data to associating stream by overloading instance methods: Write()

Example : Writing data: integer, double, string values to file

```
using (FileStream fs = new FileStream("binaryfile.dat", FileMode.Create))
{
    using (BinaryWriter writer = new BinaryWriter(fs))
    {
        int intValue = 42;
        double doubleValue = 3.14159;
        string stringValue = "Hello, world!";

        writer.Write(intValue);
        writer.Write(doubleValue);
        writer.Write(stringValue);
    }
}
```

- An instance of the class **BinaryReader** read data from an associating stream.

- Creating a BinaryReader associating with a given readable stream

+ Prototypes:

BinaryReader(stream:Stream)

+ Example:

```
FileStream fileStream = new FileStream("example.txt", FileMode.Open);
StreamReader streamReader = new StreamReader(fileStream);
```

- Reading a data from associating stream by instance methods: Read...()

Example : Reading integer, double, and string values from a file

```
using (FileStream fs = new FileStream("binaryfile.dat", FileMode.Open))
{
    using (BinaryReader reader = new BinaryReader(fs))
    {
        int intValue = reader.ReadInt32();
        double doubleValue = reader.ReadDouble();
        string stringValue = reader.ReadString();

        Console.WriteLine(
            $"int: {intValue}, double: {doubleValue}, string: {stringValue}");
    }
}
```

Note: When reading, an exception of the class *EndOfStreamException* might be thrown

Demo-People Storage(BinaryData)

- 1- Saving (in binary data)all people input to a file
- 2- Loading (from binary data) all people from file and viewing