

DICOM

～ダイコムから情報を読み出す～

DICOMとは

現在、医療画像の形式として利用されている
CT,MR,PETなどの画像も全部この形式

DICOM 単に画像データだけではなく
多くの情報が含まれている

(例として、機械, 検査日時, 病院名や患者の名前や
誕生日, 検査の種類, 検査部位, シークエンスの名前,
Field of View, 患者のポジション, スライスの厚み,
ピクセルデータの間隔, シリーズ数, シリーズナンバー
などが入っていてこのほかにもたくさんある)

基本知識の確認

文字は1文字で1byteを使って表現される。

1byte = 8bit であり、1byteでは2の8乗、すなわち256まで数えることができる。

これから出てくる数字は16進数なので、1byteは二桁の数字であらわされる。
(例として70なら16進数で'46'と表される)

DICOMヘッダー

G(Group)、E(Element)、VR(Value Representation)、Dataの連続で構成

G	E	VR	Data	G	E	VR	data
G	E	VR	data		G	E	VR
Data						
.....			G	E	VR	IMAGE	
→→→							

例として
(G,E)=(7FE0, 0010)
VR=OW+画像長

DICOMヘッダー

G (Group)、E (Element)とは・・・

- 2Byte分それぞれ使われている
- あるDataの名前(タグ)
- タグの番号とそれに対応するDataはDICOM規格により決まっている

Group	情報内容
0008	画像に関する情報
0010	患者に関する情報
0018	収集に関する情報
0020	画像付帯情報
0028	画像表示に関する情報

Group	情報内容
0008	画像に関する情報
0010	患者に関する情報
0018	収集に関する情報
0020	画像付帯情報
0028	画像表示に関する情報

Table 2 DICOM Tag

Tag番号	Tag英名称	日本語名称
(0008, 0020)	Study Date	検査日付
(0008, 0030)	Study Time	検査時刻
(0008, 0050)	Accession Number	受付番号
(0010, 0010)	Patient's Name	患者の名前
(0010, 0020)	Patient ID	患者ID
(0010, 0030)	Patient's Birth Date	患者の誕生日
(0010, 0040)	Patient's Sex	患者の性別
(0010, 1010)	Patient's Age	患者の年齢
(0018, 0015)	Body Part Examined	検査部位
(0020, 0011)	Series Number	シリーズ番号
(0028, 1050)	Window Center	ウィンドウ中心
(0028, 1051)	Window Width	ウィンドウ幅

DICOMヘッダー

VR(Value Representation)とは・・・

- 4Byte分使われている
- 中身は最初の2ByteがDataの形式で、残りの2ByteがData長となっている

Dataの形式とは・・・

- DS、USなどといった文字が入っている
- 例)DS(Decimal String)はDataが10進数
を示す文字列であることを示す

DICOMヘッダー

VR(Value Representation)とは・・・

例外1) Dataの形式はDICOMによっては入っていないことがある

⇒4Byte分すべてを使ってData長が入っている

例外2) Data形式が画像Dataの際に使われるOW(Other Word String)のときは、Data長が画像Data長となるので、残り2Byteには入りきらない

⇒この2Byteには何も入らず、次の4ByteにData長が入る

DICOMヘッダー

Dataとは・・・

- VRに書いてあるData長分のByte数で書かれている
- Dataから、画像情報・患者情報・撮影についてなどの情報を得ることができる
- Data得るにはタグを見つける必要がある

DICOM画像

DICOMヘッダー後に入っており、画像Dataに対応するタグは(7FE0,0010)である

G	E	VR	Data	G	E	VR	data	
G	E	VR	data			G	E	VR
Data							
.....			G	E	VR	IMAGE		
→→→								

例として
(G,E)=(7FE0, 0010)
VR=OW+画像長

プログラム

DICOMの読み方(ヘッダー)...

1. 2Byteずつ読み、ほしいgroupを探す
[br.ReadInt16()で2byteずつ読める]
2. ほしいgroupが見つかったら、そのすぐ
次の2Byte(element)を読む
[br.ReadInt16()で2byteずつ読める]
3. 目的のタグであったらVRからData長を
読み、そのByte分Dataを読む

プログラム

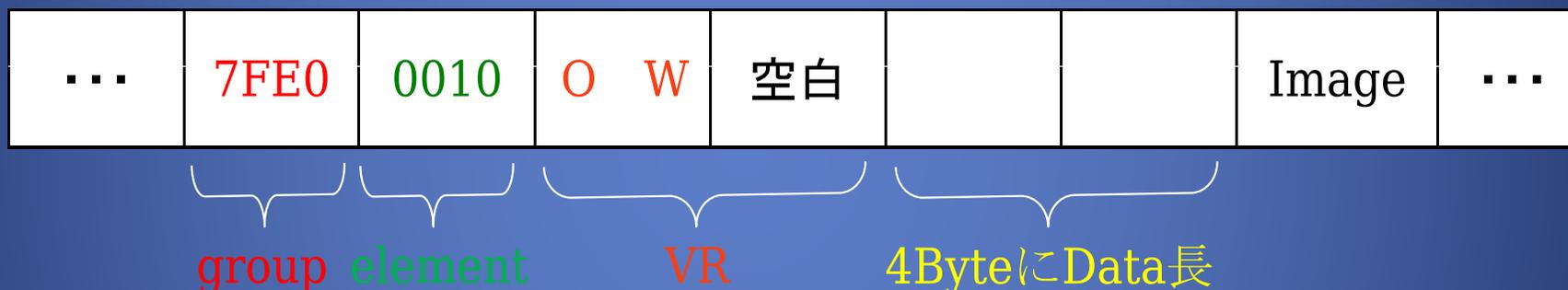
DICOMの読み方(画像)...

1. タグ(7FE0,0010)を見つける
2. VRにOWがあるかどうかを調べる
3. OWがなければ、VRの次から画像Dataが入っており、OWがあれば、次の4Byteに画像Data長があるので、その次から画像Dataが入っている

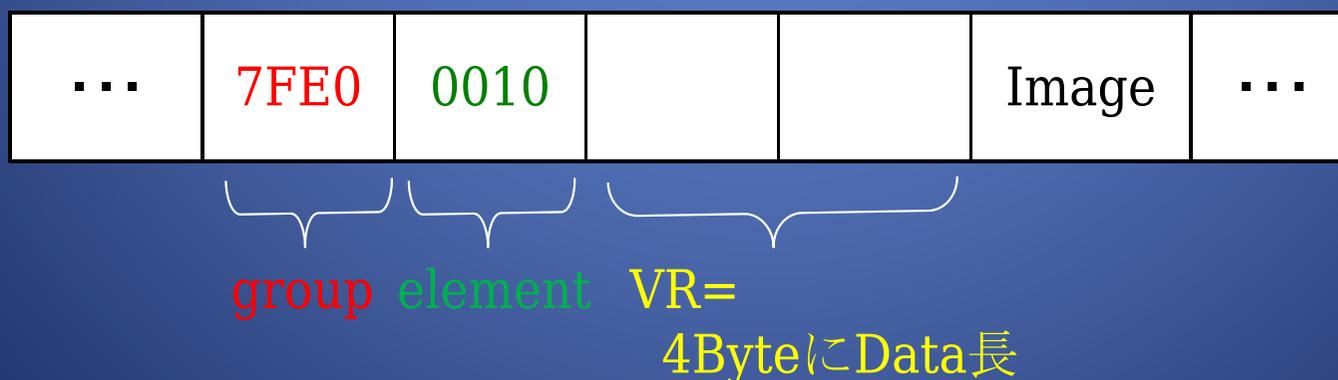
プログラム

DICOMの読み方(画像)...

OWがあるとき(2Byte表示)



OWがないとき(2Byte表示)



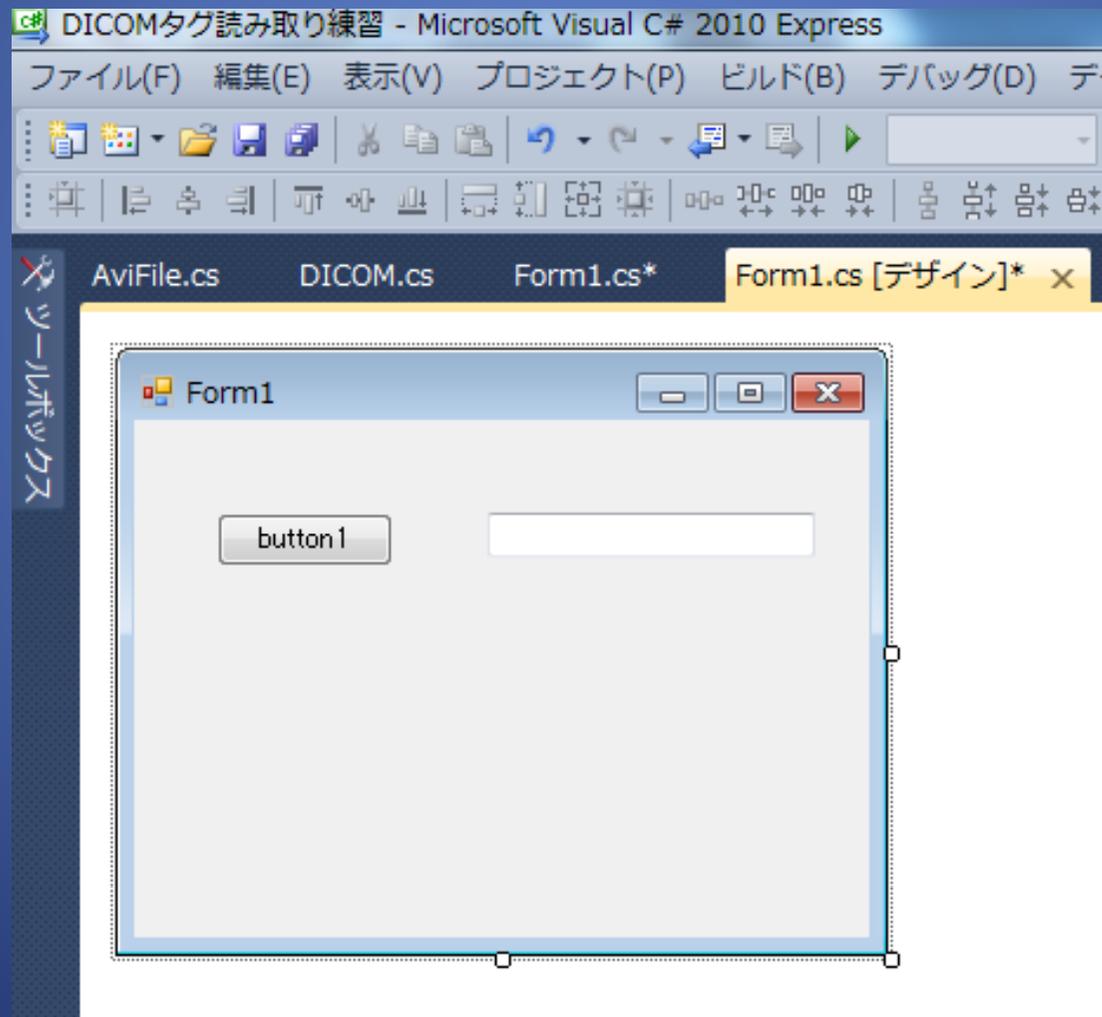
プログラム

プログラムの書き方(今回は画像の縦幅を読んでいる)

```
BinaryReader br = new BinaryReader(ofd.OpenFile());
    //group 0028 element 0010
    int data, height;
    char[] VR = new char[10];
    while (true)
    { if (br.ReadInt16() == 0x0028)
        {if (br.ReadInt16() == 0x0010)
            {for (int i = 0; i < 4; i++) {VR[i] = br.ReadChar();}
                if (VR[0] == 'U' && VR[1] == 'S')
                    {data = VR[2] + VR[3] * 256; }
                else
                    {data = VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256; }
                }
                height = br.ReadInt16();
                break;
            }
        }
    }
br.Close();
textBox1.Text =height.ToString();
```


実演 (DICOMから名前を読む)

まず、C#を開き、ツールボックスから
ボタンとテキストボックスを配置してみよう



実演

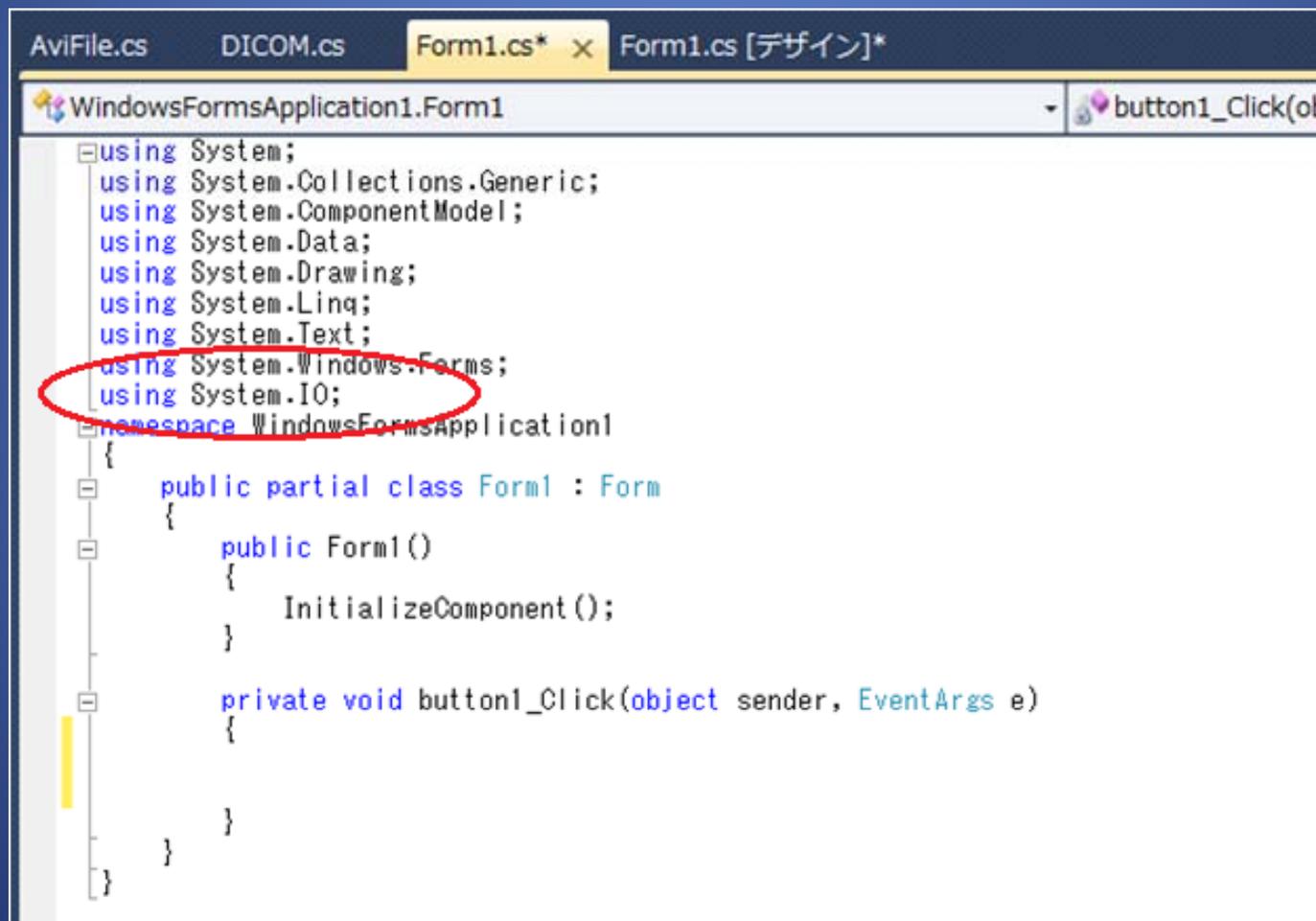
作ったボタンをダブルクリックして
ボタンのイベントハンドラーを作ろう。

```
AviFile.cs  DICOM.cs  Form1.cs* x  Form1.cs [デザイン]*
WindowsFormsApplication1.Form1  button1_Click(obj)
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            この{}の中にコードを書き込んでいく
        }
    }
}
```

実演

コード画面上部のusing～にSystem.IOを書き込み



```
WindowsFormsApplication1.Form1 button1_Click(ob
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

このコードをButtonの{ }内に
書き込む

```
OpenFileDialog ofd = new OpenFileDialog();
ofd.ShowDialog();
BinaryReader br = new BinaryReader(ofd.OpenFile());
int data;
char[] VR = new char[10];
char[] NAME = new char[100];
string str="";
while (true)
{
    if (br.ReadInt16() == 0x0010)//0xが先頭につくと16進数の意味になる
    {
        if (br.ReadInt16() == 0x0010)
        {
            for (int i = 0; i < 4; i++)
            {
                VR[i] = br.ReadChar();
            }

            if (VR[0] == 'P' && VR[1] == 'N')
            {
                data = VR[2] + VR[3] * 256;
            }
            else
            {
                data = VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256;
            }

            for (int i = 0; i < data; i++)
            {
                NAME[i] = br.ReadChar();
                str += NAME[i];
            }
            break;
        }
    }
}
br.Close();
textBox1.Text =str.ToString();
```

実演

書き終わったら動かしてみよう

The screenshot shows a Windows desktop environment. In the foreground, a Visual Studio IDE window is open, displaying a C# form named 'Form1'. The form contains a single button labeled 'button1' and a text box containing the name 'Kato Chietsugu^^'. The background shows a Windows Explorer window displaying a directory listing of files. The files are listed in a table with columns for '名前' (Name), '更新日時' (Last Modified), and '種類' (Type). The files are all DCM files with names starting with 'm001-s0001-IM-0001-0279-0001....' and ending with 'DCM ファイル'. The update time for all files is '2012/07/24 18:03'. The Explorer window also shows a sidebar with 'ローカルディスク (C:)' and 'ネットワーク' (Network) options.

名前	更新日時	種類
m001-s0001-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0002-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0003-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0004-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0005-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0006-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0007-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0008-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0009-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0010-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0011-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0012-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0013-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0014-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0015-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0016-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0017-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル
m001-s0018-IM-0001-0279-0001....	2012/07/24 18:03	DCM ファイル