

C#でいきなりDICOM

C++との主な違い 一応まとめ

変数はどこでも宣言可能。

便利。

インテリセンスが良い。

予測機能がある。

C++との主な違い 一応まとめ

配列の書きかたはxx[i][j]ではなくxx[i,j]

上記で動くが、C#では多次元配列は避けた方が良さらしい。
避ける場合は無理やり1次元配列にする。

基本的にポインタは使えないらしい。

ポインタを使いこなせていない人には関係ない話。

C++との主な違い 一応まとめ

よくわからない関数がたくさんある。

なんとかなる。

よくわからないコードがたくさんある。

なんとかなる。

コードで見る違い

とりあえずコードを紹介

フォーム作成は適当にやればできる



← このコードをみていく。

このフォームはツールボックスからアイテムをドラッグドロップすればすぐにできる。

上から ProgressBar
 Button
 PictureBox
右に vScrollBar

Buttonのfontやtextは右下のプロパティで可変
PictureBoxはBackColorやSizeで調整

全コード

```
Form1.cs Form1.cs [デザイン]
DICOMviewer.Form1 num
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;
using System.Drawing.Imaging;

namespace DICOMviewer
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            int num, width, height, slice, slicen=0, min=0,max=0;
            int tdose, ht, weight, sh, sm, ss;
            double thick;
            static int[, ] xx;

            private void button1_Click(object sender, EventArgs e)//read file...
            private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial...
        }
    }

namespace BitmapPlus...
```

コード上部

コード体部

コード下部

コード上部

```
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;  
using System.Drawing.Imaging;
```

最初からある。
気にしない。

とりあえず付け足す。
気にしない。

C++でいうところの #include

コード体部

```
namespace DICOMviewer
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        int num, width, height, slice, slicen=0, min=0,max=0;
        int tdose, ht, weight, sh, sm, ss;
        double thick;
        static int[, ] xx;

        private void button1_Click(object sender, EventArgs e)//read file...
        private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial...
    }
}
```

勝手に作られる部分
気にしない

コード体部

```
namespace DICOMviewer
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

ここにコードをかく

```
int num, width, height, slice, slicen=0, min=0,max=0;
int tdose, ht, weight, sh, sm, ss;
double thick;
static int[, ,] xx;

private void button1_Click(object sender, EventArgs e)//read file...
private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial...
```

コード体部

```
int num, width, height, slice, slicen=0, min=0,max=0;  
① int tdose, ht, weight, sh, sm, ss;  
double thick;  
static int[, ,] xx;  
  
② private void button1_Click(object sender, EventArgs e)//read file...  
  
③ private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial...
```

折りたたんでいるから
コンパクト

コードを省略している

流れとしては

- ①変数宣言
- ②ボタンがクリックされたときに実行するコード
- ③スクロールバーがスクロールされたときに実行するコード

“private...” は、フォームからボタンなどをダブルクリックすると自動的に作成される。

コード体部

```
int num, width, height, slice, slicen=0, min=0,max=0;  
① int tdose, ht, weight, sh, sm, ss;  
double thick;  
static int[, ,] xx;  
  
② private void button1_Click(object sender, EventArgs e)//read file...  
  
③ private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial...
```

始め(①)に宣言している変数は共有される。
つまり、②で `width = 64;` と入力されれば、③でもその情報が保持される。

とりあえずここで特筆すべきは `static int[, ,] xx;`
`static` は二次元以上の配列宣言にはとりあえずつけておく。
カンマの数が何次元配列かを表す。
つまり、「int型の3次元配列を持つ変数」の意。

複数宣言したい場合は `static int[, ,] xx, x0, yy;` などと続けていく。

二次元配列なら『`int[,]`』、一次元配列なら『`int[]`』
型をかえたいならintをdoubleなどに書き変える。

コード体部

引き続き配列宣言について。

違和感があるのは、配列の大きさを指定していないから。

C++では `static int xx[64][64], yy[32][32];` のように宣言していた。

C#ではいつでもどこでも(②とか③の中で)配列の大きさを決められるし変えられる。

配列を使用する前には `xx = new int [64,64];` のように記述する必要がある。

また、大きさの指定は変数でも可能で、例えば

```
static int[,] xx;  
int width =64, height = 64;  
xx = new int [width, height];
```

ともかける。3文目は馴染みないだろうが、3日で慣れる。

計算のコードについては、この配列さえおさえれば

あとはC++と同じコーディングだと個人的には思ってます。

コード体部

```
int num, width, height, slice, slicen=0, min=0,max=0;  
① int tdose, ht, weight, sh, sm, ss;  
double thick;  
static int[, ,] xx;  
  
② private void button1_Click(object sender, EventArgs e)//read file...  
  
③ private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial...
```

②を展開してみる

コード体部

```
private void button1_Click(object sender, EventArgs e) //read file
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    if (fbd.ShowDialog() == DialogResult.OK)
    {
        DirectoryInfo d = new DirectoryInfo(fbd.SelectedPath);
        FileSystemInfo[] f = d.GetFilesSystemInfos();
        num = f.Length;

        int LV, flg=0;
        short group, element;
        string str;
        char[] VR = new char[10];
        char[] data = new char[100];

        long pos = 0;

        progressBar1.Maximum = num;
        progressBar1.Minimum = 0;

        for (int n = 0; n < num; n++)
        {
            try
            {
                str = "";
                BinaryReader b = new BinaryReader(new FileStream(f[n].FullName, FileMode.Open));
                b.BaseStream.Seek(0, SeekOrigin.Begin);

                while (true)
                {
                    GROUP:
                    group = b.ReadInt16();
                    pos = b.BaseStream.Position;
                    switch (group)
                    {

                        case 0x0028:
                            element = b.ReadInt16();
                            switch (element)
                            {
                                case 0x0010://height
                                    for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
                                    if (VR[0] != 'U' && VR[1] != 'S')
                                    {
                                        LV = (int)(VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256);
                                        if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; };
                                    }
                                    else { LV = VR[2] + VR[3] * 256; if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; }; }
                                    height = b.ReadInt16();
                                    break;

                                case 0x0011://width
                                    for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
                                    if (VR[0] != 'U' && VR[1] != 'S')
                                    {
                                        LV = (int)(VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256);
                                        if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; };
                                    }
                                    else { LV = VR[2] + VR[3] * 256; if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; }; }
                                    width = b.ReadInt16();
                                    break;

                                default: b.BaseStream.Position = pos; goto GROUP;
                            }
                            //element
                            break;//group:0x0028 break

                        case 0x0010:
                            element = b.ReadInt16();
                            switch (element)
                            {
                                case 0x1030//積読士者〇の体重d
                                    for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
                                    if (VR[0] != 'D' && VR[1] != 'S')
                                    {
                                        LV = (int)(VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256);
                                        if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; };
                                    }
                                    else { LV = VR[2] + VR[3] * 256; if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; }; }
                                    for (int c = 0; c < 2; c++) //LVだ?とAEGラー←[
                                    {
                                        data[c] = b.ReadChar();
                                        str += data[c].ToString();
                                    }
                                    weight = int.Parse(str);
                                    str = "";
                                    break;
                                default: b.BaseStream.Position = pos; goto GROUP;
                            }
                        }
                    }
                }
            }
            break;
        }
    }
}
```

```
case 0x0018:
element = b.ReadInt16();
switch (element)
{
    case 0x0050//thickness
    {
        for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
        if (VR[0] != 'D' && VR[1] != 'S')
        {
            LV = (int)(VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256);
            if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; };
        }
        else { LV = VR[2] + VR[3] * 256; if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; }; }
        for (int c = 0; c < 4; c++)
        {
            data[c] = b.ReadChar();
            str += data[c].ToString();
        }
        thick = double.Parse(str);
        str = "";
    }
    break;

case 0x1072//投?号*時?間〇
{
    for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
    if (VR[0] != 'T' && VR[1] != 'W')
    {
        LV = (int)(VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256);
        if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; };
    }
    else { LV = VR[2] + VR[3] * 256; if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; }; }

    for (int c = 0; c < 6; c++)
    {
        data[c] = b.ReadChar();
    }

    for (int c = 0; c < 2; c++) str += data[c].ToString();
    sh = int.Parse(str);
    str = "";
    for (int c = 2; c < 4; c++) str += data[c].ToString();
    sm = int.Parse(str);
    str = "";
    for (int c = 4; c < 6; c++) str += data[c].ToString();
    ss = int.Parse(str);
    str = "";
}
break;

case 0x1074://Total dose
{
    for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
    if (VR[0] != 'D' && VR[1] != 'S')
    {
        LV = (int)(VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256);
        if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; };
    }
    else
    {
        LV = VR[2] + VR[3] * 256;
        if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; };
    }

    for (int c = 0; c < LV; c++)
    {
        data[c] = b.ReadChar();
        str += data[c].ToString();
    }
    tdose = int.Parse(str);
    str = "";
}
break;

case 0x1075://半?減、期u
{
    for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
    if (VR[0] != 'D' && VR[1] != 'S')
    {
        LV = (int)(VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256);
        if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; };
    }
    else
    {
        LV = VR[2] + VR[3] * 256;
        if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; };
    }

    for (int c = 0; c < LV; c++)
    {
        data[c] = b.ReadChar();
        str += data[c].ToString();
    }
    ht = int.Parse(str);
    str = "";
}
break;

default: b.BaseStream.Position = pos; goto GROUP;
}
//ele
break;//gro:0018
```

```
case 0x0020:
element = b.ReadInt16();
switch (element)
{
    case 0x0013//slice番号?
        for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
        if (VR[0] != 'Y' && VR[1] != 'S')
        {
            LV = (int)(VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256);
            if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; };
        }
        else { LV = VR[2] + VR[3] * 256; if (100 < LV) { b.BaseStream.Position = pos; goto GROUP; }; }
        for (int c = 0; c < LV; c++)
        {
            data[c] = b.ReadChar();
            str += data[c].ToString();
        }
        slice = int.Parse(str);
        str = "";
        break;
        default: b.BaseStream.Position = pos; goto GROUP;
    }
    break;//gr:0020
}
//switch

if (group == 0x7f0e)
{
    element = b.ReadInt16();
    if (element == 0x0010)
    {
        for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
        if (VR[0] != 'O' && VR[1] != 'W') LV = b.ReadInt32();
        else LV = VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256;

        if (flg == 0) { sx = new int[width, height, num]; flg++;
        for (int j = 0; j < height; j++)
        {
            for (int i = 0; i < width; i++)
            {
                short count = b.ReadInt16();
                xs[i, j, slice] = count;
                if (count > max) max = count;
                if (count < min) min = count-0;
                if (slice > slicen) slicen = slice;
            }
        }
        break;
    }
}
//group
//row:COMread無3限ループ→v
//try
catch {};
progressBar1.Value = n;
Application.DoEvents();
//file:無〇減?0?forループ→v
progressBar1.Value = 0;
vScrollBar1.Minimum = 1;
vScrollBar1.Maximum = num;
}
```

長い

コード体部

コードのほとんどはDICOMの中に入っている、
画像の大きさ・厚さ、ピクセルサイズ、検査日時など
もりだくさんの情報を読み取るためのもの。

前のスライドでも必要最低限のものしか読んでいないが、
あれくらいのサイズにはなる。

②ではDICOMデータから画像データのみ(boneみたいなもの)を
取り出す作業をしている。ごちゃごちゃしてるので読み飛ばしてかまわない。

従って、ここでは細かい情報を読み取るコードは割愛し、
簡単にまとめたコードを紹介する。

コードが間違っていたりするので、さっと眺めるだけにしてください。
模範プログラムはそのうちもらえらると思います。

コード体部

読み飛ばしok

```
private void button1_Click(object sender, EventArgs e)//read file
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    if (fbd.ShowDialog() == DialogResult.OK)
    {
        DirectoryInfo d = new DirectoryInfo(fbd.SelectedPath);
        FileSystemInfo[] f = d.GetFileSystemInfos();
        num = f.Length;

        int LV, flg=0;
        short group, element;
        string str;
        char[] VR = new char[10];
        char[] data = new char[100];

        long pos = 0;

        progressBar1.Maximum = num;
        progressBar1.Minimum = 0;

        for (int n = 0; n < num; n++)
        {
            try
            {
                str = "";
                BinaryReader b = new BinaryReader(new FileStream(f[n].FullName, FileMode.Open));
                b.BaseStream.Seek(0, SeekOrigin.Begin);

                while (true)
                {
                    GROUP:
                    group = b.ReadInt16();
                    pos = b.BaseStream.Position;

                    switch (group)
                    {
                    }//switch

                    if (group == 0x7fe0)
                    {
                        element = b.ReadInt16();
                        if (element == 0x0010)
                        {
                            for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
                            if (VR[0] != '0' && VR[1] != '0') LV = b.ReadInt32();
                            else LV = VR[0] + VR[1] * 256 + VR[2] * 256 + VR[3] * 256 * 256 * 256;

                            if (flg == 0) { xx = new int[width, height, num]; flg++;}
                            for (int j = 0; j < height; j++)
                            {
                                for (int i = 0; i < width; i++)
                                {
                                    short count = b.ReadInt16();
                                    xx[i, j, slice] = count;
                                    if (count > max) max = count;
                                    if (count < min) min = count=0;
                                    if (slice > slicen) slicen = slice;
                                }
                            }
                            break;
                        }
                    }
                }
            }
            catch { }
            progressBar1.Value = n;
            Application.DoEvents();
        }
        progressBar1.Value = 0;
        vScrollBar1.Minimum = 1;
        vScrollBar1.Maximum = num;
    }
}
```

DICOMデータに入っている
画像の大きさやスライスの厚さなどの
情報を読み取るコードを切った。

この部分。

コード体部 構造

読み飛ばしok

```
private void button1_Click(object sender, EventArgs e) //read file
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    if (fbd.ShowDialog() == DialogResult.OK)
    {
        DirectoryInfo d = new DirectoryInfo(fbd.SelectedPath);
        FileSystemInfo[] f = d.GetFileSystemInfos();
        num = f.Length;

        int LV, flg=0;
        short group, element;
        string str;
        char[] VR = new char[10];
        char[] data = new char[100];

        long pos = 0;

        progressBar1.Maximum = num;
        progressBar1.Minimum = 0;
    }
}
```

データがあるフォルダ指定

前のスライドで割愛したところで使う変数の宣言
string は文字列変数

```
for (int n = 0; n < num; n++)
{
    try
    {
        str = "";
        BinaryReader b = new BinaryReader(new FileStream(f[n].FullName, FileMode.Open));
        b.BaseStream.Seek(0, SeekOrigin.Begin);

        while (true)
        {
            GROUP:
            {
                group = b.ReadInt16();
                pos = b.BaseStream.Position;
                switch (group)
                {
                }
            }
            //switch

            if (group == 0x7fe0)
            {
                element = b.ReadInt16();
                if (element == 0x0010)
                {
                    for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
                    if (VR[0] != '0' && VR[1] != '0') LV = b.ReadInt32();
                    else LV = VR[0] * 256 + VR[1] * 256 + VR[2] * 256 + VR[3] * 256 * 256 * 256;

                    if (flg == 0) { xx = new int[width, height, num]; flg++; }
                    for (int j = 0; j < height; j++)
                    {
                        for (int i = 0; i < width; i++)
                        {
                            short count = b.ReadInt16();
                            xx[i, j, slice] = count;
                            if (count > max) max = count;
                            if (count < min) min = count=0;
                            if (slice > slicen) slicen = slice;
                        }
                    }
                    break;
                }
            }
            //group
        }
        //rowDICOMread無限ループ
    }
    catch { };
    progressBar1.Value = n;
    Application.DoEvents();
}
progressBar1.Value = 0;
vScrollBar1.Minimum = 1;
vScrollBar1.Maximum = num;
}
```

DICOMデータ読み込み

コード体部

読み飛ばしok

```
private void button1_Click(object sender, EventArgs e)//read file
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    if (fbd.ShowDialog() == DialogResult.OK)
    {
        DirectoryInfo d = new DirectoryInfo(fbd.SelectedPath);
        FileSystemInfo[] f = d.GetFileSystemInfos();
        num = f.Length;

        int LV, Flg=0;
        short group, element;
        string str;
        char[] R = new char[10];
        char[] data = new char[100];

        long l;
        progressBar1.Maximum = num;
    }
}
```

データがあるフォルダ指定

呪文として使うべきコード
これによりフォルダを選択できる

```
FolderBrowserDialog fbd = new FolderBrowserDialog();
if (fbd.ShowDialog() == DialogResult.OK)
{
    DirectoryInfo d = new DirectoryInfo(fbd.SelectedPath);
    FileSystemInfo[] f = d.GetFileSystemInfos();
    num = f.Length;
}
```

ダイアログが表示され、そこからフォルダが選択されれば
if文の中に入る。
後ろにつける()は気にしない。
書かないとエラー。

```
        pos = progressBar1.Position;
        switch (group)
        {
        }
    }
}

//switch
if (group == 0x7fe0)
{
    element = b.ReadInt16();
    if (element == 0x0010)
    {
        for (int c = 0; c < 4; c++) VR[c] = 0;
        if (VR[0] != '0' && VR[1] != '0')
        else LV = VR[0] + VR[1] * 256 + VR[2] * 256 + VR[3] * 256;

        if (flg == 0) { xx = new int[width]; }
        for (int j = 0; j < height; j++)
        {
            for (int i = 0; i < width; i++)
            {
                short count = b.ReadInt16();
                xx[i, j, slice] = count;
                if (count > max) max = count;
                if (count < min) min = count=0;
                if (slice > slicen) slicen = slice;
            }
        }
        break;
    }
}
}

//group
//rowDICOMread無限ループ
}
}

//file編纂のforループ
progressBar1.Value = 0;
vScrollBar1.Minimum = 1;
vScrollBar1.Maximum = num;
}
```

if ここまで。
フォルダが選択されなかった場合は何もせずに終わる。

コード体部

読み飛ばしok

```
private void button1_Click(object sender, EventArgs e)//read file
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    if (fbd.ShowDialog() == DialogResult.OK)
    {
        DirectoryInfo d = new DirectoryInfo(fbd.SelectedPath);
        FileSystemInfo[] f = d.GetFileSystemInfos();
        num = f.Length;

        int LV, Flg=0;
        short group, element;
        string str;
        char[] R = new char[10];
        char[] data = new char[100];

        long count = 0;
        progressBar1.Maximum = num;
    }
}
```

データがあるフォルダ指定

呪文として使うべきコード
これによりフォルダを選択できる

```
FolderBrowserDialog fbd = new FolderBrowserDialog();
if (fbd.ShowDialog() == DialogResult.OK)
{
    DirectoryInfo d = new DirectoryInfo(fbd.SelectedPath);
    FileSystemInfo[] f = d.GetFileSystemInfos();
    num = f.Length;
}
```

dにはフォルダのパス(場所)が入る
dの情報がfに入り、fから
選んだフォルダの中にファイルが何個あるか
つまり、DICOMデータにおいては
画像が何枚あるか
の情報を num に入れる。

```
        pos = progressBar1.Position;
        switch (group)
        {
            //switch
        }
        if (group == 0x7fe0)
        {
            element = b.ReadInt16();
            if (element == 0x0010)
            {
                for (int c = 0; c < 4; c++) VR[c] = b.Re
                if (VR[0] != '0' && VR[1] != '0') LV = 0
                else LV = VR[0] + VR[1] * 256 + VR[2] *
                if (flg == 0) { xx = new int[width, heig
                for (int j = 0; j < height; j++)
                {
                    for (int i = 0; i < width; i++)
                    {
                        short count = b.ReadInt16();
                        xx[i, j, slice] = count;
                        if (count > max) max = count;
                        if (count < min) min = count=0;
                        if (slice > slicen) slicen = slice;
                    }
                }
                break;
            }
        }
        //group
        //rowDICOMread無限ループ
        //try
        catch { };
        progressBar1.Value = n;
        Application.DoEvents();
        //file網羅のforループ
        progressBar1.Value = 0;
        vScrollBar1.Minimum = 1;
        vScrollBar1.Maximum = num;
    }
}
```

if ここまで。
フォルダが選択されなかった場合は何もせずに終わる。

コード体部

読み飛ばしok

```
private void button1_Click(object sender, EventArgs e)//read file
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    if (fbd.ShowDialog() == DialogResult.OK)
    {
        DirectoryInfo d = new DirectoryInfo(fbd.SelectedPath);
        FileSystemInfo[] f = d.GetFileSystemInfos();
        num = f.Length;

        int LV, flg=0;
        short group, element;
        string str;
        char[] VR = new char[10];
        char[] data = new char[100];

        long pos = 0;

        progressBar1.Maximum = num;
        progressBar1.Minimum = 0;

        for (int n = 0; n < num; n++)
        {
            try
            {
                s = "";
                BinaryReader b = new BinaryReader(new FileStream(f[n].FullName, FileMode.Open, FileAccess.Read));
                b.BaseStream.Seek(0, SeekOrigin.Begin);
                while (true)
                {
                    for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
                    if (VR[0] != '0' && VR[1] != 'W') LV = b.ReadInt32();
                    else LV = VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256;
                    if (flg == 0) { xx = new int[width, height, num]; flg = 1;
                    for (int j = 0; j < height; j++)
                    {
                        for (int i = 0; i < width; i++)
                        {
                            progressBar1.Value = n;
                            Application.DoEvents();;
                        }
                    }
                    break;
                }
                //group
                rowDICOMread無限ループ
            }
            catch { }
        }
        progressBar1.Value = n;
        Application.DoEvents();;
    }
    progressBar1.Value = 0;
    vScrollBar1.Minimum = 1;
    vScrollBar1.Maximum = num;
}
```

現在の進行状況をユーザーに知らせるためのもの。
barのmaxをファイル数のnumとして
ファイルを読む毎に、つまりforループが回る毎に
そのときのnの値をbarで表示させる。

インテリセンス(予測機能)により、ドットを打てば
Maximumを含むいろんなmethodが表示される。

わかりにくいですが解説も付いているので
色々見ていると出来ることが増えるかも。

コード体部

読み飛ばしok

```
private void button1_Click(object sender, EventArgs e)//read file
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    if (fbd.ShowDialog() == DialogResult.OK)
    {
        DirectoryInfo d = new DirectoryInfo(fbd.SelectedPath);
        FileSystemInfo[] f = d.GetFileSystemInfos();
        num = f.Length;

        int LV, flg=0;
        short group, element;
        string str;
        char[] VR = new char[10];
        char[] data = new char[100];

        long pos = 0;

        progressBar1.Maximum = num;
        progressBar1.Minimum = 0;

        for (int n = 0; n < num; n++)
```

```
    try
    {
        str = "";
        BinaryReader b = new BinaryReader(new FileStream(f[n].FullName, FileMode.Open));
        b.BaseStream.Seek(0, SeekOrigin.Begin);

        while (true)
        {
            GROUP:
            group = b.ReadInt16();
            pos = b.BaseStream.Position;
            switch (group)
            {

            }

            }//switch
```

```
        if (group == 0x7fe0)
        {
            element = b.ReadInt16();
            if (element == 0x0010)
            {
                for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
                if (VR[0] != '0' || VR[1] != '0') LV = b.ReadInt32();
            }
        }
    }
}
```

```
    try
    {
        str = "";
        BinaryReader b = new BinaryReader(new FileStream(f[n].FullName, FileMode.Open));
        b.BaseStream.Seek(0, SeekOrigin.Begin);

        while (true)
        {
            GROUP:
            group = b.ReadInt16();
            pos = b.BaseStream.Position;
            switch (group)
            {

            }

            }//switch
```

呪文

データを読む準備をしている部分。

データの読み書きは大事だが、
今後簡単にまとめる予定。

```
if (group == 0x7fe0)
{
    element = b.ReadInt16();
    if (element == 0x0010)
    {
        for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
        if (VR[0] != '0' && VR[1] != 'W') LV = b.ReadInt32();
        else LV = VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256;

        if (flg == 0) { xx = new int[width, height, num]; flg++;}
        for (int j = 0; j < height; j++)
        {
            for (int i = 0; i < width; i++)
            {
                short count = b.ReadInt16();
                xx[i, j, slice] = count;
                if (count > max) max = count;
                if (count < min) min = count=0;
                if (slice > slicen) slicen = slice;
            }
        }
        break;
    }
}
} //group
```

```
switch (group)
{
    //switch
```

```
if (group == 0x7fe0)
{
    element = b.ReadInt16();
    if (element == 0x0010)
    {
        for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
        if (VR[0] != '0' && VR[1] != 'W') LV = b.ReadInt32();
        else LV = VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256;

        if (flg == 0) { xx = new int[width, height, num]; flg++;}
        for (int j = 0; j < height; j++)
        {
            for (int i = 0; i < width; i++)
            {
                short count = b.ReadInt16();
                xx[i, j, slice] = count;
                if (count > max) max = count;
                if (count < min) min = count=0;
                if (slice > slicen) slicen = slice;
            }
        }
        break;
    }
}
} //group
```

```
try
{
    //row100read無限ループ
}
catch { };
progressBar1.Value = n;
Application.DoEvents();
} //files網羅のforループ
progressBar1.Value = 0;
vScrollBar1.Minimum = 1;
vScrollBar1.Maximum = num;
```

画像データ読み込み。
半分は呪文。
次のスライドでforループを見る

```
if (group == 0x7fe0)
{
    element = b.ReadInt16();
    if (element == 0x0010)
    {
        for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
        if (VR[0] != 'O' && VR[1] != 'W') LV = b.ReadInt32();
        else LV = VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256;

        if (flg == 0) { xx = new int[width, height, num]; flg++;}
        for (int j = 0; j < height; j++)
        {
            for (int i = 0; i < width; i++)
            {
                short count = b.ReadInt16();
                xx[i, j, slice] = count;
                if (count > max) max = count;
                if (count < min) min = count=0;
                if (slice > slicen) slicen = slice;
            }
        }
        break;
    }
}
} //group
```

ここで xx の配列を new しているが、new するたびに初期化されてしまうので、一度だけ new するために flg を使った。今考えれば if (n == 0) の方が良い。

前に説明した通り、文字で配列の大きさを指定可能。
width などは DICOM データから読んだ値を入れている。

```
if (group == 0x7fe0)
{
    element = b.ReadInt16();
    if (element == 0x0010)
    {
        for (int c = 0; c < 4; c++) VR[c] = b.ReadChar();
        if (VR[0] != 'O' && VR[1] != 'W') LV = b.ReadInt32();
        else LV = VR[0] + VR[1] * 256 + VR[2] * 256 * 256 + VR[3] * 256 * 256 * 256;

        if (flg == 0) { xx = new int[width, height, num]; flg++;}
        for (int j = 0; j < height; j++)
        {
            for (int i = 0; i < width; i++)
            {
                short count = b.ReadInt16();
                xx[i, j, slice] = count;
                if (count > max) max = count;
                if (count < min) min = count=0;
                if (slice > slicen) slicen = slice;
            }
        }
        break;
    }
}
} //group
```

してみると、変なところで int, short など変数宣言されている。
C#では、このようなことが可能。

階層的に変数の有効範囲が決まり、変なことをするとエラーが起きるが
宣言してエラーが起きたら直すということで通用する。

b.ReadInt16(); は呪文。
これによりデータを読んで、配列xxに入れていくことになる。

max 以外の if 文は必要ない。ただの消し忘れ。

コード体部

読み飛ばしok

```
private void button1_Click(object sender, EventArgs e)//read file
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    if (fbd.ShowDialog() == DialogResult.OK)
    {
        DirectoryInfo d = new DirectoryInfo(fbd.SelectedPath);
        FileSystemInfo[] f = d.GetFileSystemInfos();
        num = f.Length;

        int LV, flg=0;
        short group, element;
        string str;
        char[] VR = new char[10];
        char[] data = new char[100];

        long pos = 0;

        progressBar1.Maximum = num;
        progressBar1.Minimum = 0;

        for (int n = 0; n < num; n++)
        {
            try
            {
                str = "";
                BinaryReader b = new BinaryReader(new FileStream(f[n].FullName,
                b.BaseStream.Seek(0, SeekOrigin.Begin);

                while (true)
                {
                    GROUP:
                    group = b.ReadInt16();
                    pos = b.BaseStream.Position;
                    switch (group)
                    {

                    }

                    //switch

                    if (group == 0x7fe0)
                    {
                        element = b.ReadInt16();
                        if (element == 0x0010)
                        {
                            //readInt32(
                            // * 256 +
                            num]; f
                        }
                    }
                }
            }
            catch { }
        }

        for (int i = 0; i < width; i++)
        {
            short count = b.ReadInt16();
            xx[i, j, slice] = count;
            if (count > max) max = count;
            if (count < min) min = count=0;
            if (slice > slicen) slicen = slice;
        }
        break;
    }
}

//rowDICOMread無限ループ
//
cat { };
progressBar1.Value = n;
Application.DoEvents();
}
//files 羅のforループ
progressBar1.Value = 0;
vScrollBar1.Minimum = 1;
vScrollBar1.Maximum = num;
}
```

スクロールバーの最大値と最小値を決める。

ちなみになぜか

`vScrollBar1.Maximum = 100;`

と定義すると、max は91までしかいかない。

設定した数値から9だけ小さくなるので
避けるためには+9する必要がある。

自分の勘違いだったらごめんなさい。

以上、ざっと説明した。

コード体部

```
int num, width, height, slice, slicen=0, min=0,max=0;  
① int tdose, ht, weight, sh, sm, ss;  
double thick;  
static int[, ,] xx;  
  
② private void button1_Click(object sender, EventArgs e)//read file...  
③ private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial...
```

次は③

画像を表示させる。

コードを展開する。

コード体部

```
private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial
{
    try{
        int k = vScrollBar1.Value;
        pictureBox1.Width = width;
        pictureBox1.Height = height;

        Bitmap bmps = new Bitmap(width, height);
        BitmapPlus.BitmapPlus bmpsp = new BitmapPlus.BitmapPlus(bmps);
        bmpsp.BeginAccess();
        int count;
        for (int j = 0; j < height; j++)
        {
            for (int i = 0; i < width; i++)
            {
                count = (xx[i,j,k]-min) * 255 / (max-min);
                if (count < 0) count = 0;
                if (count > 255) count = 255;
                Color col = Color.FromArgb(count, count, count);
                bmpsp.SetPixel(i, j, col);
            }
        }
        bmpsp.EndAccess();
        pictureBox1.Image = bmps;
        pictureBox1.Refresh();
    }
    catch { }
}
```

try{}
catch{}
は、tryしてみてダメだったら
catchにいくというもの。

正しいコーディングをしていれば不要だが、
スクロールバーではエラーが起こりやすいので
重宝する。

作っている最中では、外してエラーチェックしてみる
のもよいかも。

コード体部

```
private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial
{
    try{
        int k = vScrollBar1.Value;
        pictureBox1.Width = width;
        pictureBox1.Height = height;

        Bitmap bmps = new Bitmap(width, height);
        BitmapPlus.BitmapPlus bmpsp = new BitmapPlus.BitmapPlus(bmps);
        bmpsp.BeginAccess();
        int count;
        for (int j = 0; j < height; j++)
        {
            for (int i = 0; i < width; i++)
            {
                count = (xx[i,j,k]-min) * 255 / (max-min);
                if (count < 0) count = 0;
                if (count > 255) count = 255;
                Color col = Color.FromArgb(count, count, count);
                bmpsp.SetPixel(i, j, col);
            }
        }
        bmpsp.EndAccess();
        pictureBox1.Image = bmps;
        pictureBox1.Refresh();
    }
    catch {}
}
```

スクロールバーの値をkにいれる。
スクロールバーのとり得る範囲は指定しなければならない。
25p をみてください。(3p前)

pictureBoxの大きさを画像に合わせる。
width, height は DICOM から読んだ画像の大きさ。

見覚えのあるforループ。
わからないところは呪文。
(count,count,count)は順にred, green, blue
画像を表示させたいときはこれを唱える。

コード体部

```
private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial
{
    try{
        int k = vScrollBar1.Value;
        pictureBox1.Width = width;
        pictureBox1.Height = height;

        Bitmap bmps = new Bitmap(width, height);
        BitmapPlus.BitmapPlus bmpsp = new BitmapPlus.BitmapPlus(bmps);
        bmpsp.BeginAccess();
        int count;
        for (int j = 0; j < height; j++)
        {
            for (int i = 0; i < width; i++)
            {
                count = (xx[i,j,k]-min) * 255 / (max-min);
                if (count < 0) count = 0;
                if (count > 255) count = 255;
                Color col = Color.FromArgb(count, count, count);
                bmpsp.SetPixel(i, j, col);
            }
        }
        bmpsp.EndAccess();
        pictureBox1.Image = bmps;
        pictureBox1.Refresh();
    }
    catch {}
}
```

C#では逃れられない呪文。

C#にもSetPixel関数は用意されているが、動作はおそろしく遅いらしい。

ネットを探せば色んな対策コードが公開されている。ここでは富山君から受け継いだものを用いている。

このコードを使うためには unsafe コードを有効にするという作業が必要だが、きっと富山君が紹介するのでここでは割愛。

コード下部

```
Form1.cs Form1.cs [デザイン]
DICOMviewer.Form1 num
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;
using System.Drawing.Imaging;

namespace DICOMviewer
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        int num, width, height, slice, slicen=0, min=0,max=0;
        int tdose, ht, weight, sh, sm, ss;
        double thick;
        static int[, ] xx;

        private void button1_Click(object sender, EventArgs e)//read file...
        private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)//axial...
    }
}

namespace BitmapPlus...
```

} コード下部

前スライドのコードを使うために必要
中身は私にはわかりません。
貼り付けておけば動く。

以上DICOM読み込みのコードを紹介した。

次回からは「使えるコード」を紹介したい。

なお、ボツったフォーム作成の説明スライドを
せっかくだからのせておく。

ツールボックス

ツールボックスからlabelをドラッグドロップして、
右下のプロパティのText, Font可変

ツールボックスからTextBox

ツールボックスからButton
プロパティでText,Font可変

プロパティ

Form1 System.Windows

Size	300, 300
SizeGripS	Auto
StartPosi	WindowsDe
Tag	
Text	Form1
TopMost	False

Text
コントロールに関連付けら...

プロパティ

picturebox

画像はここに表示させる

大きさはマウスで可変だが、切りの良い数字にしたいときは
プロパティのSizeを使う

BackColorで色可変

The screenshot shows the Visual Studio IDE with a Windows Form named 'Form1' in design view. The form contains a text label 'Y=aX+b', two input fields for 'a' and 'b', and a 'ひょうじ' (Display) button. A large empty rectangular area is reserved for a PictureBox, indicated by a dashed border and a red arrow pointing to it from the text box above. The Properties window on the right shows the 'Form1 System.Windows' properties, with 'BackColor' and 'BackgroundImage' highlighted. The 'BackColor' property is currently set to 'Control'.

エラー一覧

0 エラー 0 警告 0 メッセージ

説明	ファイル	行	列	プロジェクト

おわり