2019 年 4 月 18 日（木）実施

C# プログラムの基礎

基本構造

1) クラス

Visual C#のプログラムの基本単位を**クラス**と呼ぶ。Windows フォームアプリケーションを作成する際，プロジェクトを作成すると生成されるファイルのうち，Form1.cs を例にとれば，そのクラス名は Form1 である。クラスは class **キーワード**を用いて宣言する。

【Form1.cs】

```
using System;
 (中略)
using System.Windows.Forms;

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

        private void Form1_Load(object sender, EventArgs e)
        {
            // Form1 が読み込まれた際の処理
        }
    }
}
```

2) ブロック

中括弧『{』から『}』に囲まれたものを**ブロック**と呼ぶ。ブロックには複数の要素をひとまとめにする機能がある。

3) メンバ

クラスの中にある要素を**メンバ**と呼ぶ。メンバにはデータを扱うフィールドと呼ばれる要素及び処理を扱う**メソッド**（上の例では Form1_Load）や**コンストラクタ**（上の例では Form1）等がある。

4) コメント

プログラムの動作には影響を与えずに，注釈を書き込めるものを**コメント**と呼ぶ。『//』が書かれると，そこから改行の手前までがコメントとなる（複数行に渡るコメントを書くには，『/*』と『*/』とを用いて，それらの間に書く）。

＊ プログラム記述上の注意　コメントや文字列を表示する為の記述以外は，キーボードにある英字，数字及び記号の範囲で，半角文字で記述する。また，文末にはセミコロン『;』を書く。

名前空間

Visual C#で様々なプログラムを作成していく上で，他のプログラムと区別し，複数のクラスを

まとめて管理する為に**名前空間**が用いられる。上の Form1.cs では，プロジェクト名として命名した Second が namespace **キーワード**によって名前空間に設定されている。

また，.NET Framework の構成要素のクラスライブラリの名前空間には **System 名前空間**が含まれている。System 名前空間の下には，Windows 名前空間，更にその下には Forms 名前空間があるという様に階層的な構造がある。この構造を System.Windows.Forms の様にドットで繋いで表す。フォームにラベルの様なコントロール（部品）を配置する際に，**完全修飾名**では System.Windows.Forms.Label と書くことになるが，using ディレクティブを記述しておけばそこに書かれた名前空間の記述を省略することが出来る（Form1.cs の例では using System.Windows.Forms;）。

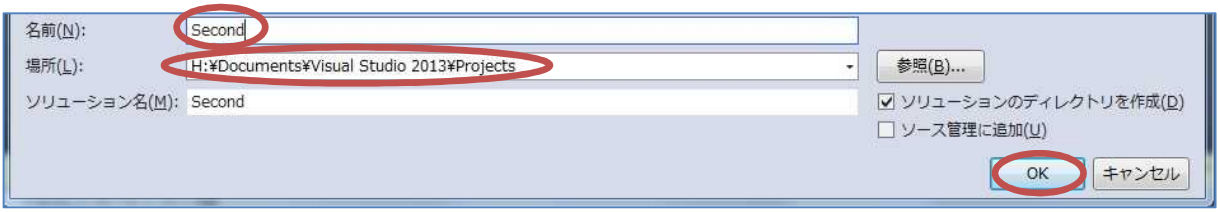
本日の課題

フォームにラベルを 2 個配置して，一方には単なる文字列，他方には計算結果を含む文字列を表示する。

手順

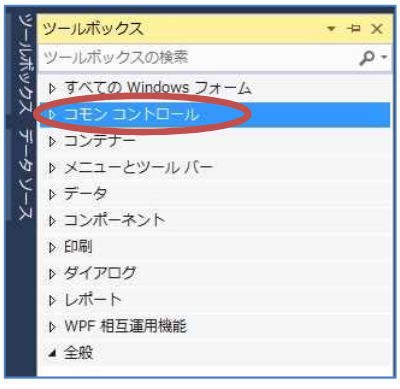
1）プロジェクトの作成

Visual Studio 2013 を起動したら，［ファイル］→［新規作成］→［プロジェクト］と辿って，プロジェクトを作成する。『新しいプロジェクト』ダイアログボックスでは，プログラミング言語を『Visual C#』，プロジェクトテンプレートとしては，『Windows フォームアプリケーション』を選択し，『名前』を「Second」に書き換え，『場所』が「H:¥Documents¥Visual Studio 2013¥Projects」となっていることを確認してから『OK』を押す（詳細は第 1 回の教材を参照）。

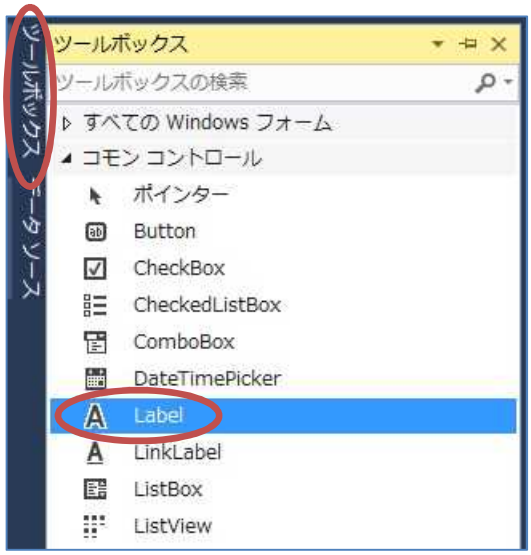


2）コントロールの配置

［表示］→［ツールボックス］と辿って，ツールボックスを表示する。



『コモンコントロール』を選択して展開する。



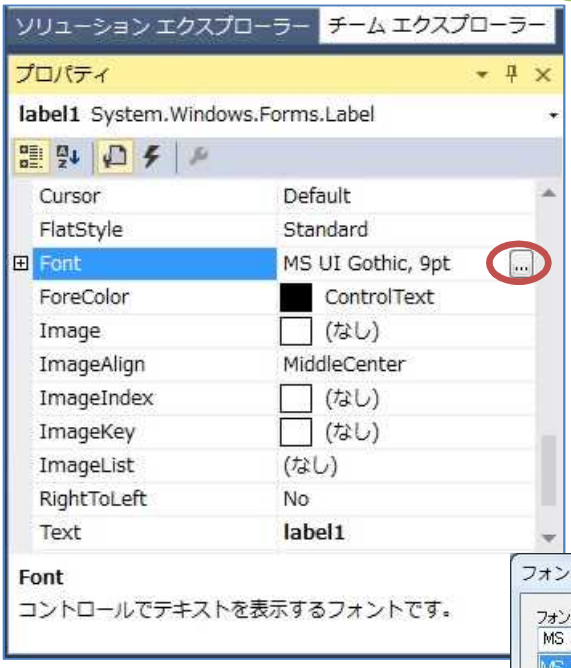
『Label』を選択してから，左にある縦の『ツールボックス』タブをクリックして，メニューを引っ込める。

フォームをクリックしてラベルを貼る（『label1』の名前が付く）。

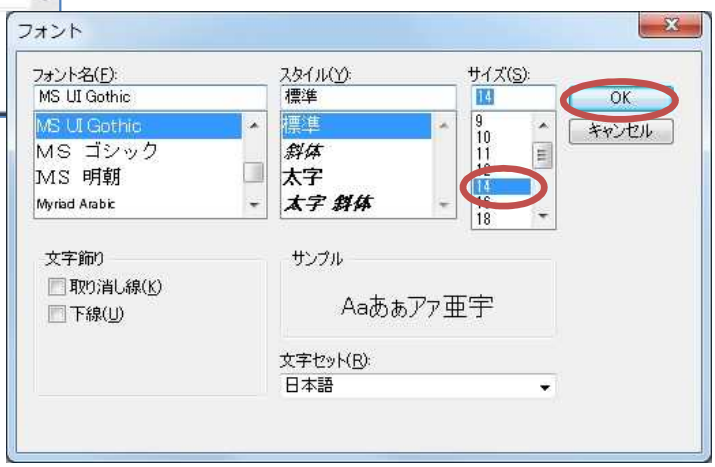
同じ操作を繰り返して，もう 1 個のラベルをフォームに貼る（『label2』の名前が付く）。

『label1』を選択して，プロパティのフォントサイズを変更する。

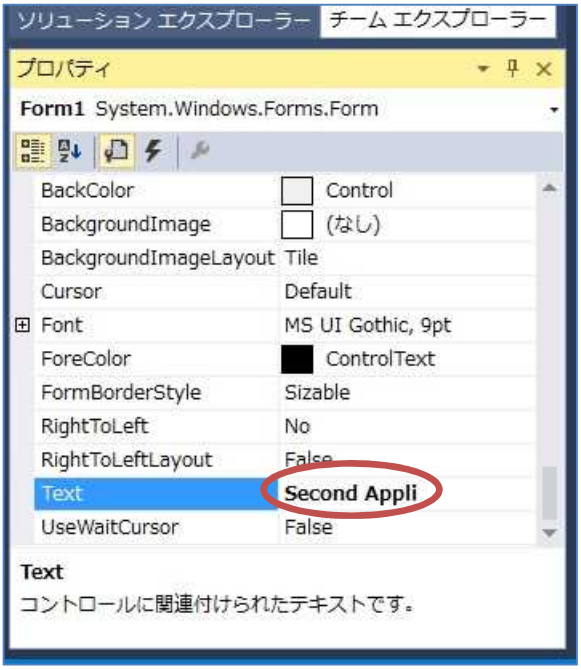
先ず，『Font』の右側にある『…』ボタンを押す。



出てきた『フォント』ダイアログボックスの『サイズ』で「14」を選択して『OK』を押す。



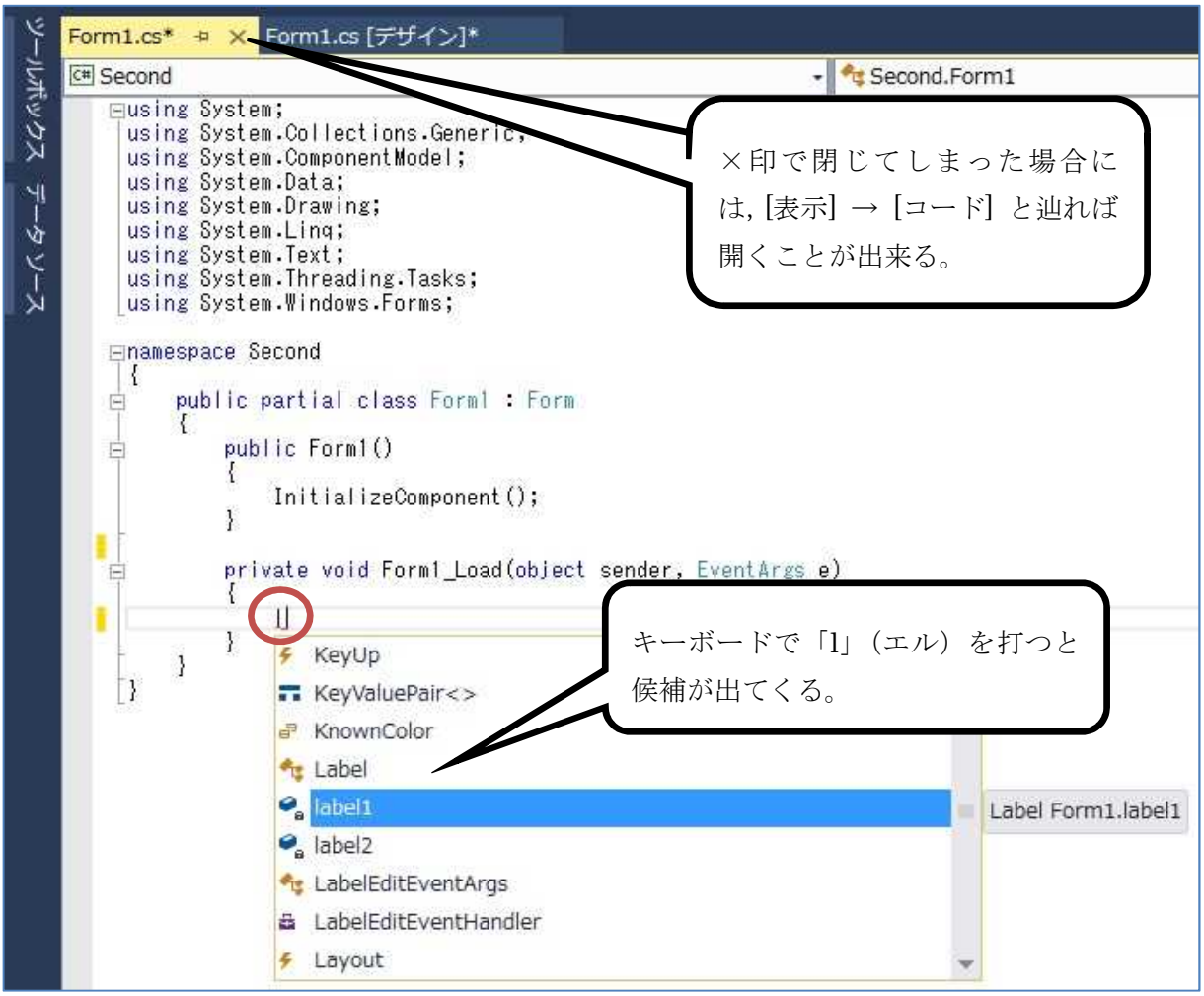
『label2』を選択して，『label1』と同様に，プロパティのフォントサイズを 14 ポイントに変更する。

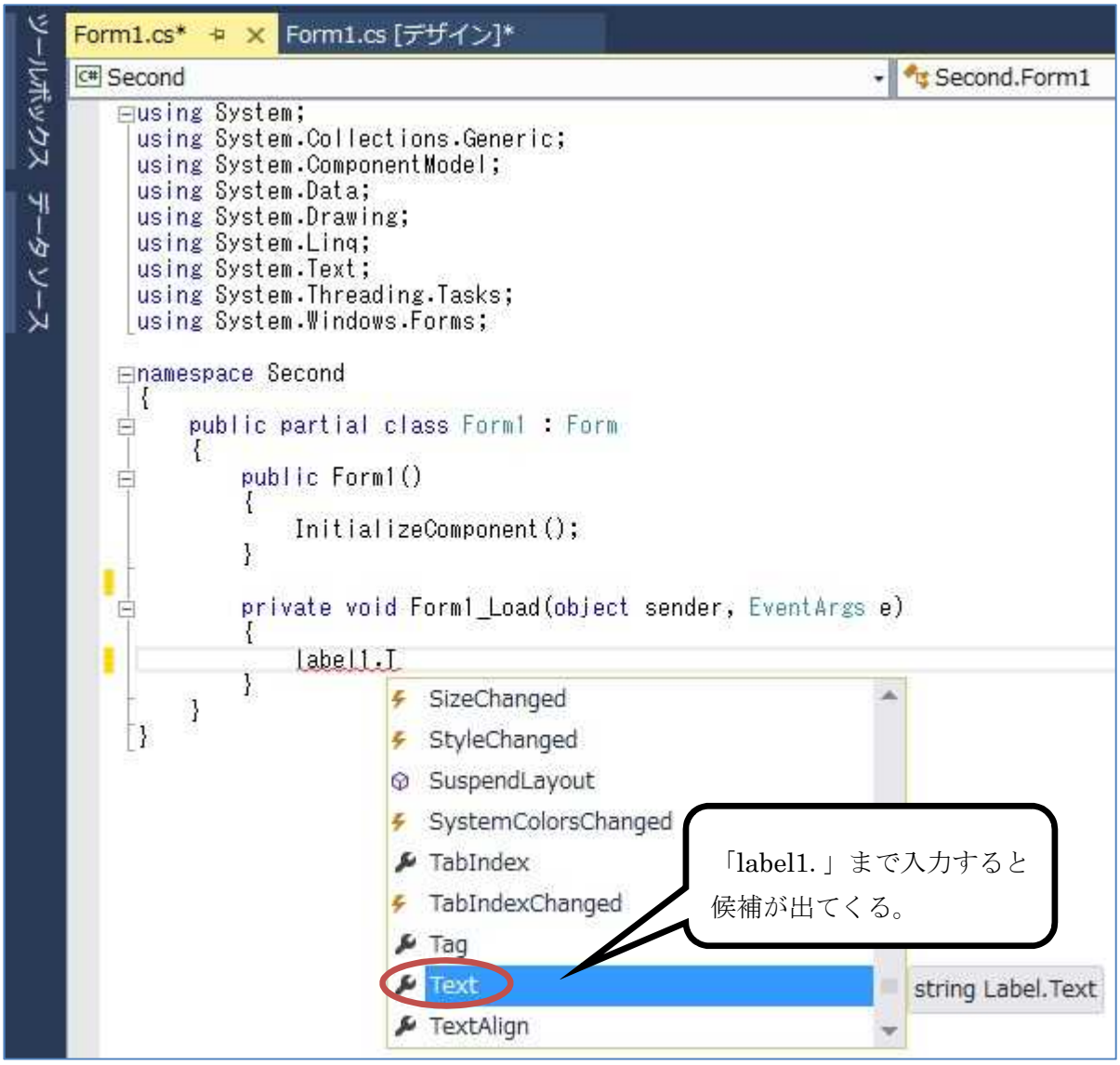


フォーム全体を選択し，プロパティの『Text』を「Form1」から「Second Appli」に書き換える。

3) コーディング

フォームをダブルクリックして，Form1. cs のコード（プログラム）を表示する。Form1_Load メソッドのブロック内に Form1 が読み込まれた際の処理を記述していく。





「label1. Text = ″ラベルへの文字列の表示″;」の 1 行を入力する。この 1 文で 1 個目のラベルに「ラベルへの文字列の表示」という**文字列リテラル**（左右の二重引用符『″』で挟まれた中身）が設定され，プログラムの起動時に Form1 が読み込まれた際に，ラベルの位置に表示される。なお，p.1 のプログラム記述上の注意に従うこと。

続けて，2 個目のラベルに設定する文字列を「label2. Text =」から始まる 1 文に記述する。この文は 5 行に渡って記述しているが，この様にセミコロンに到達するまで複数行でも構わない。また，複数の文字列リテラルの間に書いた「+」は**プラス演算子**と呼ばれ，文字列を連結する機能を有する。なお，『¥n』はエスケープシーケンスと呼ばれるものの一つで，その位置での改行を行う機能がある。

途中にある（123 + 456）は 2 つの数値を足し合わせてその結果を文字列として表示する。これを括弧でくくらずに 123 + 456 としてしまうと，文字列 123 と 456 との連結となってしまう。2 行目から 5 行目の括弧の中で用いられている『-』，『*』，『/』，『%』はそれぞれ，減算，乗算，除算，剰余算を行う為の演算子である。

Form1.cs　Form1.cs [デザイン]

Second　　Second.Form1

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

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

        private void Form1_Load(object sender, EventArgs e)
        {
            label1.Text = "ラベルへの文字列の表示";

            label2.Text = "123 + 456 の計算結果は" + (123 + 456) + "です。¥n"
                        + "123 - 456 の計算結果は" + (123 - 456) + "です。¥n"
                        + "123 * 456 の計算結果は" + (123 * 456) + "です。¥n"
                        + "123 / 456 の計算結果は" + (123 / 456) + "です。¥n"
                        + "123 % 456 の計算結果は" + (123 % 456) + "です。";
        }
    }
}
```

4） プログラムの実行

ここで，フォームデザイナーでフォームの横幅を広げてから，『すべてを保存』のボタンを押した後に，『開始』ボタンを押してプログラムを起動してみる（詳細は第 1 回の教材を参照）。

エラーが出ている場合には，修正してから保存，開始と進む。

確認が済んだら，閉じるボタンを押してプログラムを終了する。

5) オリジナルアイコンの作成

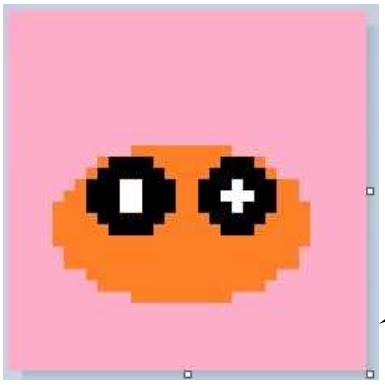
次の段階として，オリジナルアイコンを作成して利用する方法を学ぶ。

[スタートボタン] → [すべてのプログラム] → [アクセサリ] → [ペイント] と辿って，ペイント（Windows に付属する簡易描画ツール）を起動する。

『サイズ変更』のボタンを押す。『単位』を「パーセント」から「ピクセル」（描画する点の数）に変更し，『縦横比を維持する』のチェックを外してから『OK』を押す。

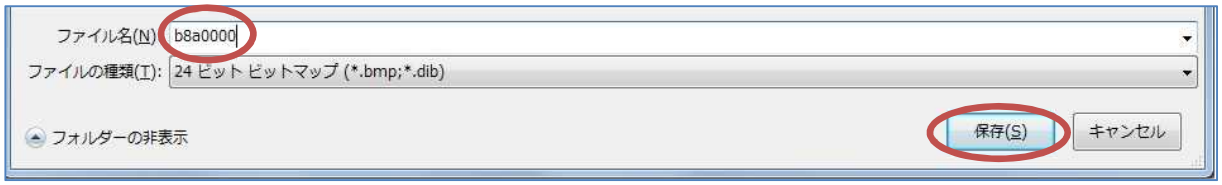
水平方向，垂直方向共に「32」ピクセルに設定する。

『表示』タブを開いて，作業がし易い大きさに拡大して表示する。ここに簡単な絵を描く。



これは教材用の例で，自分のオリジナルな絵を描く。

『ファイル』タブの『名前を付けて保存』で『BMP 画像』を選択する。第 1 回の授業で作成した，H ドライブのマイドキュメントの中の『Prog2』フォルダに，「b8a0???」の様にアカウント名を『ファイル名』欄に入力して，『保存』ボタンを押す（ファイル名には『. bmp』という拡張子が自動的に付加される)。ここで，ペイントを閉じる。

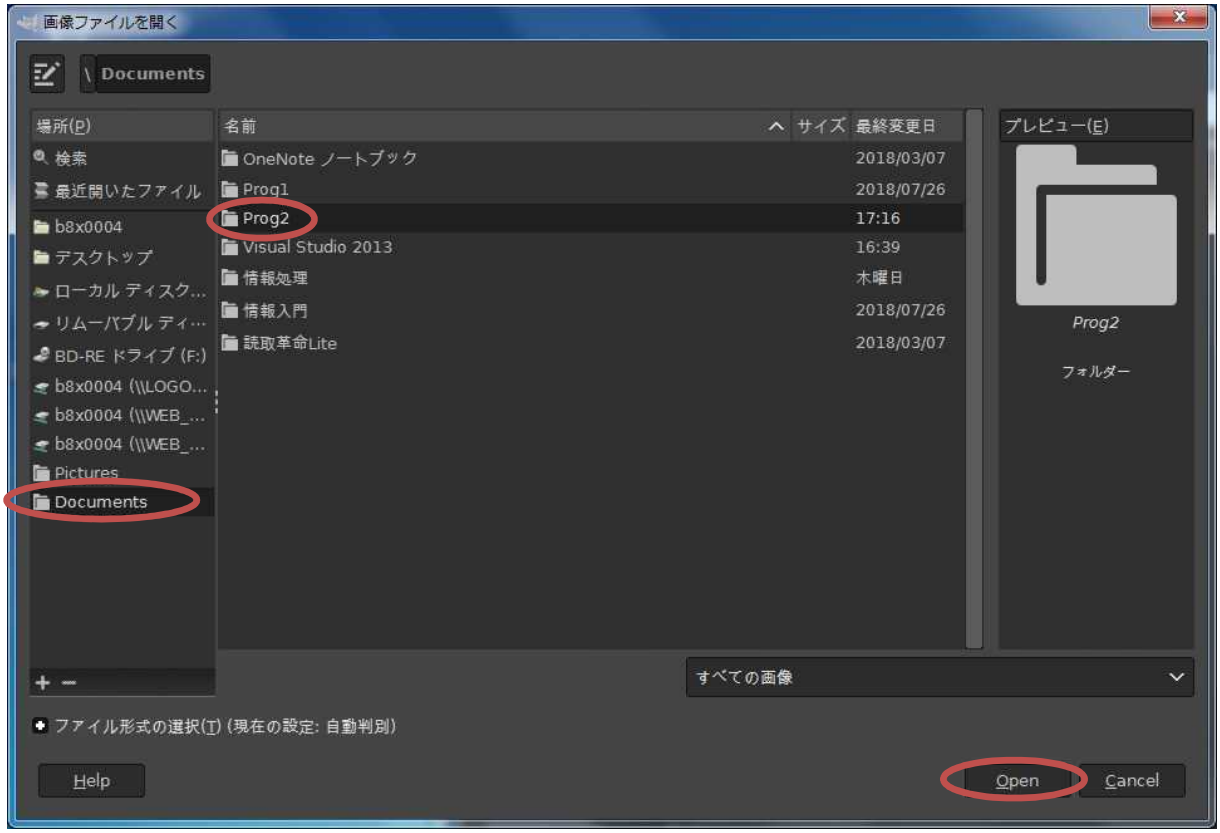


次に，画像ファイルをビットマップ形式からアイコン形式に変換する。この目的の為に，GIMP という高機能描画ツールを利用する。

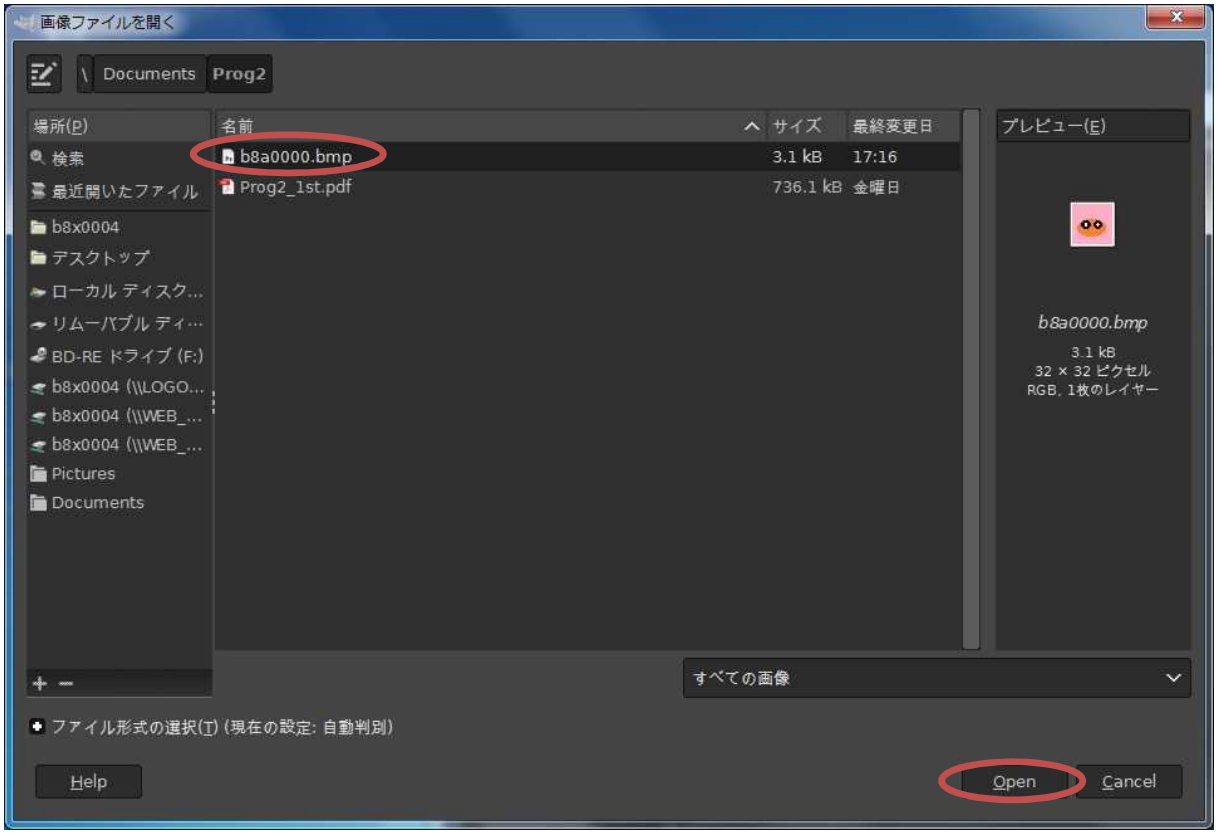
[スタートボタン] → [すべてのプログラム] → [GIMP2. 10. 8]と辿って，GIMP を起動する。

『ファイル』タブを開き,『開く/インポート』を選択する。

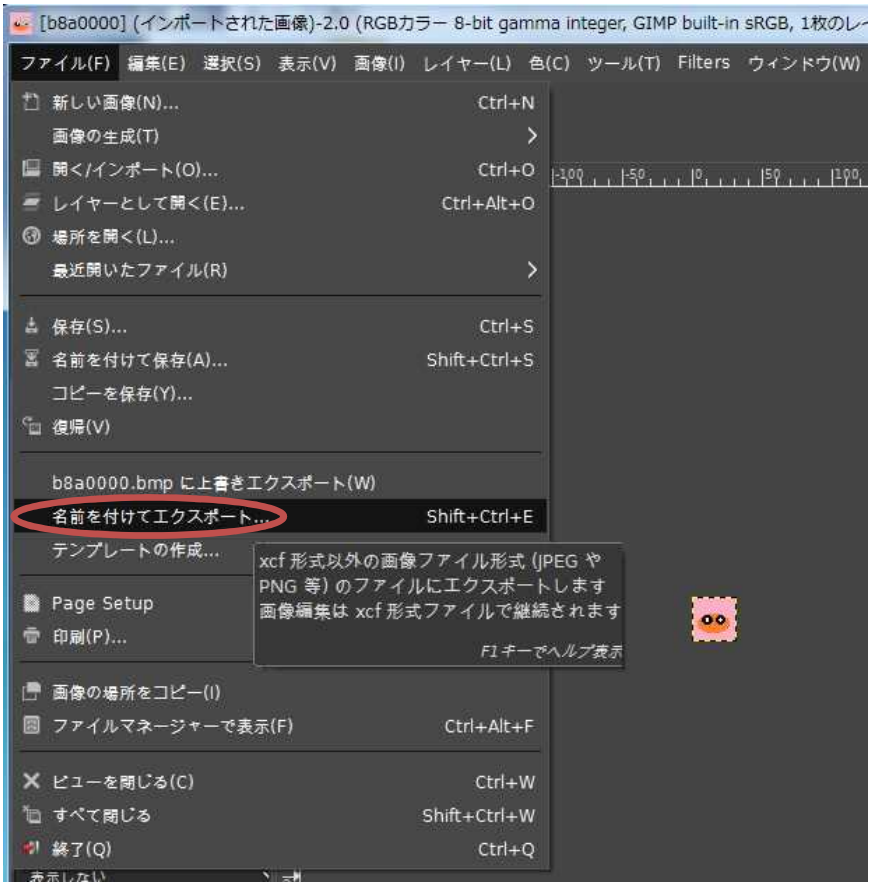
出てきた『画像ファイルを開く』ダイアログボックスで『Documents』を選択すると，『Prog2』フォルダが表示されるので，それを選択して，『Open』を押す。



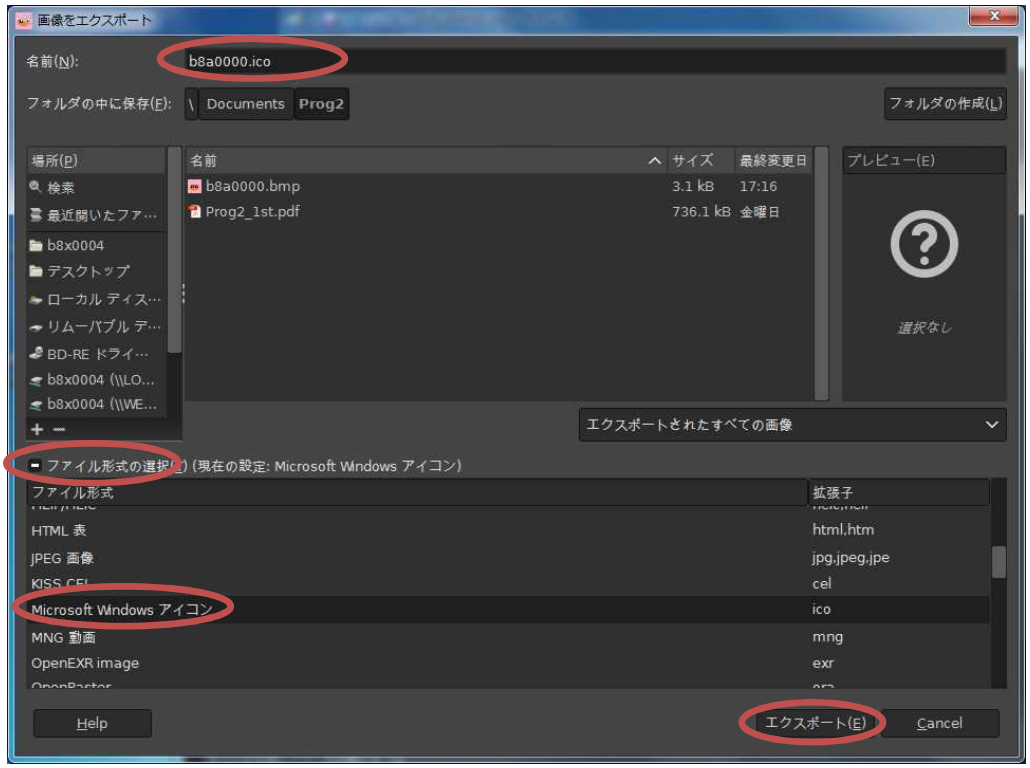
『b8a0???.bmp』を選択して，『Open』を押す。



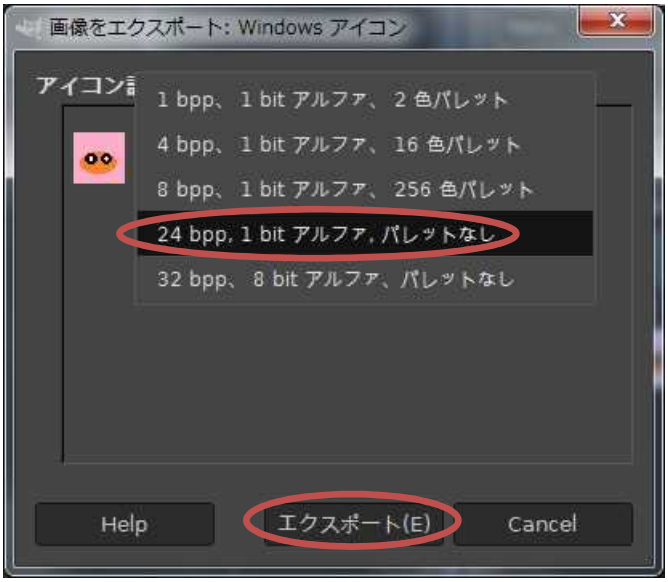
『ファイル』タブを開き，『名前を付けてエクスポート』を選択する。



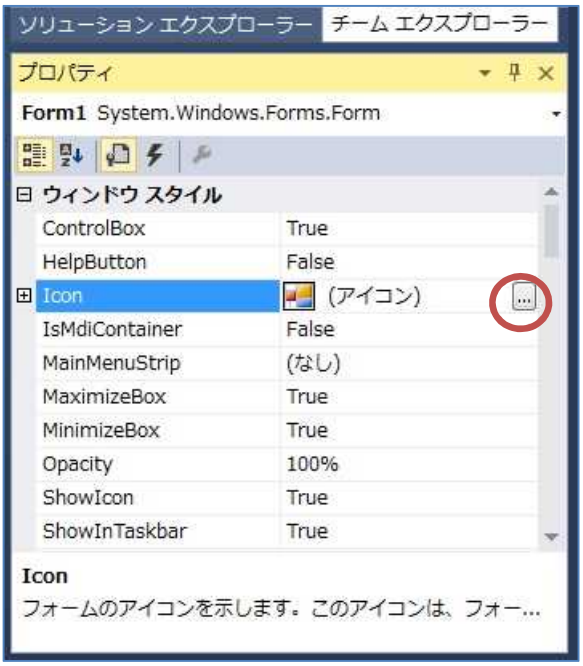
『名前』の欄の「bmp」を「ico」に書き換え，『ファイル形式の選択』を選択して展開し，『Microsoft Windows アイコン』を選択して，『エクスポート』を押す。



出てきた『画像をエクスポート』ダイアログボックスのオプションで『24bpp』を選択して，『エクスポート』を押す。ここで，GIMP を閉じる。



Visual Studio に戻り，フォームを選択して，『プロパティ』 ウィンドウで，『Icon』の右側にある『…』ボタンを押す。



H ドライブのマイドキュメントの中の『Prog2』フォルダを開き，『b8a0???.ico』を選択する。

6) プログラムの実行・最終確認

『すべてを保存』ボタンを押してから，『開始』ボタンを押して，プログラムを実行する。

確認が済んだら，閉じるボタンを押してプログラムを終了する。

提出物：

1) フォームのデザインファイル Form1. Designer.cs をメールに添付して提出する。
2) フォームを含むソースファイル Form1. cs をメールに添付して提出する。
3) オリジナルアイコンファイル b8a0???. ico をメールに添付して提出する。
4) 質問を記述したファイル Questions_2nd.txt に解答を書き込んで保存し，メールに添付して提出する