* 2019년 1학기 윈도우 프로그래밍 과제

1. 주제
2. 내가 ○○○○○○ 라면?
3. 윈도우 어플리케이션으로 개발 할 것 – 데이터 처리를 위한 한 개 이상의 클래스를 구현하여 사용할 것
4. 첨부된 양식에 정리하여 제출
5. ‘개발 내용’ 란에는 다음과 같은 사항을 정리할 것

1) 구현한 소스코드 제시 – 핵심 되는 코드에는 반드시 주석을 작성할 것

2) 프로젝트를 구성하는 소스 프로그램의 작업내용을 간단히 정리할 것.

1. ‘구현 결과’ 란에는 실행결과를 처리과정에 따라 정리할 것.
2. 제출기간
3. 6월10일 까지 스마트 캠퍼스

|  |  |  |  |
| --- | --- | --- | --- |
| **프로젝트 결과보고서** | | | |
|  | | | |
| 프로젝트 주제명 | 내가 세무사 라면 | | |
| 교과목명 | 윈도우 프로그래밍 | 학번 | 20135151 |
| 개발 기간 | 2019.05.16. ~ 2019.05.31 | 이름 | 이갑성 |
| 제출일 | 2019년 06월 02일 |  |  |
|  | | | |
| **Ⅰ. 개발배경** | | | |
| 세금 계산을 하는데 있어서 세금의 종류마다 계산하는 방법이 다 다르다. 특정 세금은 보유기간, 나이 등에 따라 추가적으로 세율이 달라지는 경우가 있고, 1세대 1주택에 해당되는 경우 추가적인 계산을 해야 되는 경우도 있다. 따라서 세금을 계산하는데 있어 일반적인 사람들은 이런 것들을 다 기억하기는 쉽지 않아 계산을 하는데 어려움이 따른다. 개발자 본인은 세금 계산을 하는데 있어 일반 사람들이 이러한 복잡한 것들을 다 알지 않고도 편리하게 금액만 기입하고 추가적으로 해당하는 것들을 선택하여 세금 계산을 할 수 있게 하고 싶어 개발하게 되었다. | | | |
|  | | | |
| **Ⅱ. 개발 내용** | | | |
| **[startForm.cs]**  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 TransferTax  {  public partial class startForm : Form  {    public startForm()  {  InitializeComponent();  }  //메인 메뉴기 나오게 해준다.  private void button1\_Click(object sender, EventArgs e)  {  Close();  }  //종료하는 버튼  private void button2\_Click(object sender, EventArgs e)  {  if (MessageBox.Show("프로그램을 종료하시겠습니까?", "세금계산기", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)  {  Application.Exit();  }  }  }  }  **[menuForm.cs]**  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 TransferTax  {  public partial class menuForm : Form  {  compForm cf;  startForm sf; //처음시작할때 뜨는 폼 선언  exitForm ef; //프로그램 종료시 뜨는 폼 선언  achiForm af;  public menuForm()  {    InitializeComponent();  }  //메뉴폼이 뜨기전에 시작할때 뜨는 폼을 우선적으로 띄어준다.  private void menuForm\_Load(object sender, EventArgs e)  {  sf = new startForm();  sf.Size = new System.Drawing.Size(523, 560);  sf.ShowDialog();    }    //양도소득세 버튼을 눌렀을때  private void trnsBtn\_Click(object sender, EventArgs e) //양도소득세 버튼  {  TransForm tf = new TransForm();  tf.Show();  }    //종료버튼을 눌렀을때  private void clsBtn\_Click(object sender, EventArgs e)  {  if (MessageBox.Show("프로그램을 종료하시겠습니까?", "세금계산기", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)  {  ef = new exitForm();  ef.ShowDialog();  }  }  //취득세 버튼 눌렀을떄  private void acquBtn\_Click(object sender, EventArgs e)  {  af = new achiForm();  af.ShowDialog();  }  //종합부동산세 버튼 눌렀을때  private void compBtn\_Click(object sender, EventArgs e)  {  cf = new compForm();  cf.ShowDialog();  }  }  }  **[achiForm.cs]**  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 TransferTax  {  public partial class achiForm : Form  {  int achi, result;  achiDetailForm adf; //취득가액 상세폼인 adf 선언  achiCal ac; //취득세 계산 클래스 ac 선언  public achiForm()  {  InitializeComponent();  ac = new achiCal();  }  //취득가액을 상세입력하기 위해 상세 폼을 띄우는 버튼  private void achiDetailBtn\_Click(object sender, EventArgs e)  {  int total;  adf = new achiDetailForm(); //취득가액 상세폼 adf 생성  this.AddOwnedForm(adf); //취득세 폼(achiForm)이 취득가액 상세폼 adf를 소유  adf.ShowDialog();  total = int.Parse(achiTxt.Text);  achi = total;  ac.setAchi(achi);  achiTxt.Text = String.Format("{0:#,###}", achi);  }  //해당되는 취득유형을 선택함에 따라 세율을 결정해주는 이벤트  private void button1\_Click(object sender, EventArgs e)  {  int select;  select = comboBox1.SelectedIndex;  if (select == 0)  {  ac.setPer(0.01);  }  else if (select == 1)  {  ac.setPer(0.02);  }  else if (select == 2)  {  ac.setPer(0.03);  }  else if (select == 3)  {  ac.setPer(0.028);  }  else if (select == 4)  {  ac.setPer(0.035);  }  else if (select == 5)  {  ac.setPer(0.028);  }  else if (select == 6)  {  ac.setPer(0.028);  }  else if (select == 7)  {  ac.setPer(0.023);  }  result = ac.achiResult();  resultBox.Text = String.Format("{0:#,###}", result);  }  //메인메뉴로 돌아가기 위한 폼 종료  private void button2\_Click(object sender, EventArgs e)  {  Close();  }    }  }  **[achiDetailForm.cs]**  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 TransferTax  {  public partial class achiDetailForm : Form  {  int[] money; //각각의 텍스트박스에 입력된 금액을 저장하는 money배열  int total; //총 취득가액을 저장할 total 변수  public achiDetailForm()  {  InitializeComponent();  money = new int[7];  for (int i = 0; i < money.Length; i++)  {  money[i] = 0;  }  total = 0;  }  //취득가액 외에 포함된 금액이 있으면 체크박스에 체크  //체크하면 금액 입력 텍스트박스와 버튼이 나온다.  private void checkBox1\_CheckedChanged(object sender, EventArgs e)  {  CheckBox ch = sender as CheckBox;  if (ch.Name.Equals(checkBox1.Name))  {  box2.Visible = !box2.Visible;  btn2.Visible = !btn2.Visible;  }  else if (ch.Name.Equals(checkBox2.Name))  {  box3.Visible = !box3.Visible;  btn3.Visible = !btn3.Visible;  }  else if (ch.Name.Equals(checkBox3.Name))  {  box4.Visible = !box4.Visible;  btn4.Visible = !btn4.Visible;  }  else if (ch.Name.Equals(checkBox4.Name))  {  box5.Visible = !box5.Visible;  btn5.Visible = !btn5.Visible;  }  else if (ch.Name.Equals(checkBox5.Name))  {  box6.Visible = !box6.Visible;  btn6.Visible = !btn6.Visible;  }  else if (ch.Name.Equals(checkBox6.Name))  {  box7.Visible = !box7.Visible;  btn7.Visible = !btn7.Visible;  }  }  //해당하는 금액을 텍스트박스에 입력하고 그 금액을 저장하기위한 입력버튼  private void btn1\_Click(object sender, EventArgs e)  {  int num;  Button bt = sender as Button;  if (bt.Name.Equals(btn1.Name))  {  if (box1.Text.Length == 0)  {  MessageBox.Show("취득가액을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box1.Text);  money[0] = num;  box1.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn2.Name))  {  if (box2.Text.Length == 0)  {  MessageBox.Show("건설자금 이자를 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box2.Text);  money[1] = num;  box2.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn3.Name))  {  if (box3.Text.Length == 0)  {  MessageBox.Show("할부이자, 연체료를 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box3.Text);  money[2] = num;  box3.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn4.Name))  {  if (box4.Text.Length == 0)  {  MessageBox.Show("용역비, 수수료를 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box4.Text);  money[3] = num;  box4.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn5.Name))  {  if (box5.Text.Length == 0)  {  MessageBox.Show("채무인수액을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box5.Text);  money[4] = num;  box5.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn6.Name))  {  if (box6.Text.Length == 0)  {  MessageBox.Show("주택채권 매각차손을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box6.Text);  money[5] = num;  box6.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn7.Name))  {  if (box7.Text.Length == 0)  {  MessageBox.Show("기타를 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box7.Text);  money[6] = num;  box7.Text = String.Format("{0:#,###}", num);  }  }    }  //취득가액과 취득가액 외의 금액의 총액을 계산해주는 이벤트  private void btn8\_Click(object sender, EventArgs e)  {  for (int i = 0; i < money.Length; i++)  {  if (money[i] != 0)  {  total += money[i];  }  }  resultBox.Text = String.Format("{0:#,###}", total);  }  //총 취득가액을 achiForm의 achiTxt에 전달해주고 폼을 종료시켜주는 이벤트  private void rsltBtn\_Click(object sender, EventArgs e)  {  ((achiForm)Owner).achiTxt.Text = total.ToString();  Close();  }    }  }  **[achiCal.cs]**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace TransferTax  {  class achiCal  {  int achi;  double per;  int result;  public achiCal()  {  achi = 0;  result = 0;  per = 0.0;  }  public void setAchi(int achi)  {  this.achi = achi;  }  public void setPer(double per)  {  this.per = per;  }  public int achiResult()  {  result = (int)(achi \* per);    return result;  }  }  }  **[TransForm.cs]**  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 TransferTax  {  public partial class TransForm : Form  {  int trns, acqu, need; //양도가액, 취득가액, 필요경비  transCal tc; //계산을 도와줄 transCal 클래스의 tc 선언  acquDetailForm ad; //취득가액을 세부적으로 입력할 수 있는 폼 acquDetailForm ad선언  public TransForm()  {  InitializeComponent();  tc = new transCal();    }  //양도가액 입력하는 버튼  private void trnsBtn\_Click(object sender, EventArgs e)  {  trns = 0;  if (trnsTxt.Text.Length == 0)  {  MessageBox.Show("양도가액 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  trns = int.Parse(trnsTxt.Text);  tc.setTrns(trns);  trnsTxt.Text = String.Format("{0:#,###}", trns);  }  }    //취득가액의 상세입력을 처리하는 버튼  private void acquDetailBtn\_Click(object sender, EventArgs e)  {  int total;  ad = new acquDetailForm();  this.AddOwnedForm(ad);  ad.ShowDialog();  total = int.Parse(acquTxt.Text);  acqu = total;  tc.setAcqu(acqu);  acquTxt.Text = String.Format("{0:#,###}", acqu);  }  //필요경비 입력하는 버튼  private void needBtn\_Click(object sender, EventArgs e)  {  need = 0;  if (acquTxt.Text.Length == 0)  {  MessageBox.Show("필요경비를 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  need = int.Parse(needTxt.Text);  tc.setNeed(need);  needTxt.Text = String.Format("{0:#,###}", need);  }  }  //양도차익 계산해주는 이벤트  private void minusBtn\_Click(object sender, EventArgs e)  {  int result = tc.minuCal();  minusTxt.Text = String.Format("{0:#,###}", result);  }  //선택한 보유기간에 따른 공제율을 tc객체에 넣어서 장기보유특별공제액을 계산해주는 이벤트  private void per1Btn\_Click(object sender, EventArgs e)  {  int select;  select = comboBox1.SelectedIndex;  if (select == 0)  {  tc.setPer1(0);  }  else if (select == 1)  {  tc.setPer1(0.06);  }  else if (select == 2)  {  tc.setPer1(0.08);  }  else if (select == 3)  {  tc.setPer1(0.1);  }  else if (select == 4)  {  tc.setPer1(0.12);  }  else if (select == 5)  {  tc.setPer1(0.14);  }  else if (select == 6)  {  tc.setPer1(0.16);  }  else if (select == 7)  {  tc.setPer1(0.18);  }  else if (select == 8)  {  tc.setPer1(0.20);  }  else if (select == 9)  {  tc.setPer1(0.22);  }  else if (select == 10)  {  tc.setPer1(0.24);  }  else if (select == 11)  {  tc.setPer1(0.26);  }  else if (select == 12)  {  tc.setPer1(0.28);  }  else if (select == 13)  {  tc.setPer1(0.3);  }  int result;  result = tc.longCal();  if (result == 0)  {  longTxt.Text = "0";  }  else  {  longTxt.Text = String.Format("{0:#,###}", result);  }  }  //양도소득금액을 계산해주는 이벤트  private void achvBtn\_Click(object sender, EventArgs e)  {  int result;    result = tc.achCal();  if (result == 0)  {  stndTxt.Text = "0"; //공제액으로 인한 세금을 안내도 되는 상황  }  else  {  achvTxt.Text = String.Format("{0:#,###}", result);  }  }  //양도소득 과세표준을 계산해주는 이벤트  private void stndBtn\_Click(object sender, EventArgs e)  {  int result;  result = tc.stndCal();  if (result == 0)  {  stndTxt.Text = "0"; //공제액으로 인한 세금을 안내도 되는 상황  }  else  {  stndTxt.Text = String.Format("{0:#,###}", result);  }  }  //세율을 처리하는 이벤트  private void per2Btn\_Click(object sender, EventArgs e)  {  int select;  select = comboBox2.SelectedIndex;  if (select == 0)  {  tc.setPer2(0.7);  }  else if (select == 1)  {  tc.setPer2(0.4);  }  else if (select == 2)  {  tc.setPer2(0.15);  }  int result;  result = tc.rsltCal();  rsltTxt.Text = String.Format("{0:#,###}", result);  }    //메인메뉴로 돌아가는 이벤트  private void button1\_Click(object sender, EventArgs e)  {  Close();  }  }  }  **[acquDetailForm.cs]**  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 TransferTax  {  public partial class acquDetailForm : Form  {  int[] money; //각각의 텍스트박에 입력된 금액을 저장하는 money배열  int total; //총 취득가액을 저장할 total 변수  public acquDetailForm()  {  InitializeComponent();  money = new int[8];  for (int i = 0; i < money.Length; i++)  {  money[i] = 0;  }  total = 0;  }  //체크박스를 체크했을때 금액을 입력하는 텍스트 박스와 입력하는 버튼을 보이게 해주는 이벤트  private void checkBox1\_CheckedChanged(object sender, EventArgs e)  {  CheckBox cb = sender as CheckBox;  if (cb.Name.Equals(checkBox1.Name))  {  box2.Visible = !box2.Visible;  btn2.Visible = !btn2.Visible;  }  else if (cb.Name.Equals(checkBox2.Name))  {  box3.Visible = !box3.Visible;  btn3.Visible = !btn3.Visible;  }  else if (cb.Name.Equals(checkBox3.Name))  {  box4.Visible = !box4.Visible;  btn4.Visible = !btn4.Visible;  }  else if (cb.Name.Equals(checkBox4.Name))  {  box5.Visible = !box5.Visible;  btn5.Visible = !btn5.Visible;  }  else if (cb.Name.Equals(checkBox5.Name))  {  box6.Visible = !box6.Visible;  btn6.Visible = !btn6.Visible;  }  else if (cb.Name.Equals(checkBox6.Name))  {  box7.Visible = !box7.Visible;  btn7.Visible = !btn7.Visible;  }  else if (cb.Name.Equals(checkBox7.Name))  {  box8.Visible = !box8.Visible;  btn8.Visible = !btn8.Visible;  }  }  //btn1일때는 취득가액을 입력하게 해주는 이벤트 핸들러임과 동시에  //체크한 체크박스에 해당하는 종류의 금액이 입력되면 그 금액을 money배열에 값을 넣어주는 이벤트  private void btn1\_Click(object sender, EventArgs e)  {  int num;  Button bt = sender as Button;  if (bt.Name.Equals(btn1.Name))  {  if (box1.Text.Length == 0)  {  MessageBox.Show("취득가액을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box1.Text);  money[0] = num;  box1.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn2.Name))  {  if (box2.Text.Length == 0)  {  MessageBox.Show("취득세액을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box2.Text);  money[1] = num;  box2.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn3.Name))  {  if (box3.Text.Length == 0)  {  MessageBox.Show("등록면허세액을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box3.Text);  money[2] = num;  box3.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn4.Name))  {  if (box4.Text.Length == 0)  {  MessageBox.Show("부동산중개보수액을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box4.Text);  money[3] = num;  box4.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn5.Name))  {  if (box5.Text.Length == 0)  {  MessageBox.Show("법무사비용을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box5.Text);  money[4] = num;  box5.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn6.Name))  {  if (box6.Text.Length == 0)  {  MessageBox.Show("소송,화해비용 을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box6.Text);  money[5] = num;  box6.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn7.Name))  {  if (box7.Text.Length == 0)  {  MessageBox.Show("컨설팅비용을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box7.Text);  money[6] = num;  box7.Text = String.Format("{0:#,###}", num);  }  }  else if (bt.Name.Equals(btn8.Name))  {  if (box8.Text.Length == 0)  {  MessageBox.Show("기타란을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  num = int.Parse(box8.Text);  money[7] = num;  box8.Text = String.Format("{0:#,###}", num);  }  }  }  //취득가액과 체크한 항목의 금액을 총 합산하여 총 취득가액에 금액을 표시해주는 이벤트  private void btn9\_Click(object sender, EventArgs e)  {  for (int i = 0; i < money.Length; i++)  {  if (money[i] != 0)  {  total += money[i];  }  }  resultBox.Text = String.Format("{0:#,###}", total);  }  //TransForm에 "2. 취득가액" 텍스트 박스에 금액이 전달되 이벤트  private void rsltBtn\_Click(object sender, EventArgs e)  {  ((TransForm)Owner).acquTxt.Text = total.ToString();  Close();  }  }  }  **[transCal.cs]**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace TransferTax  {  class transCal  {  int total, longTotal;  int trns, acqu, need; //양도가액, 취득가액, 필요경비  double per1, per2;  public transCal() {  total = 0;  trns = 0;  acqu = 0;  need = 0;  per1 = 0.0;  }    public void setTrns(int t)  {  trns = t;  }  public void setAcqu(int a)  {  acqu = a;  }  public void setNeed(int n)  {  need = n;  }  public int minuCal()  {  total = trns - acqu - need;  return total;  }  public void setPer1(double p1)  {  per1 = p1;  }  public int longCal()  {  longTotal = (int)(total \* per1);  return longTotal;  }  public int achCal()  {  total = total - longTotal;  if (total < 0) //세액감면 받아서 양도소득세가 0원일 경우  {  total = 0;  }  return total;  }  public int stndCal()  {  total = total - 2500000;  if (total < 0) //세액감면 받아서 양도소득세가 0원일 경우  {  total = 0;  }  return total;  }  public void setPer2(double p)  {  per2 = p;  }  public int rsltCal()  {  total = (int)(total \* per2);  return total;  }  }  }  **[compForm.cs]**  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 TransferTax  {  public partial class compForm : Form  {  long pub, stand, result; //종부세는 금액이 크기에 int보다 long형으로 선언  double per;  compCal cc; //종부세 계산해주는 클래스 cc 선언  public compForm()  {  InitializeComponent();  cc = new compCal();  }  //해당하는 체크박스를 체크했을때 콤보박스와 입력 버튼이 등장하는 이벤트  private void checkBox1\_CheckedChanged(object sender, EventArgs e)  {  CheckBox ch = sender as CheckBox;  if (ch.Name.Equals(checkBox1.Name))  {  comboBox1.Visible = !comboBox1.Visible;  btn4.Visible = !btn4.Visible;  }  else if(ch.Name.Equals(checkBox2.Name))  {  comboBox2.Visible = !comboBox2.Visible;  btn5.Visible = !btn5.Visible;  }  }  //  private void btn1\_Click(object sender, EventArgs e)  {  Button bt = sender as Button;    //주택 공시가격 합계액을 입력하는 버튼  if (bt.Name.Equals(btn1.Name))  {  if (pubBox.Text.Length == 0)  {  MessageBox.Show("주택 공시가격 합계액 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  pub = long.Parse(pubBox.Text);  cc.setPub(pub);  pubBox.Text = String.Format("{0:#,###}", pub);  }  }  //공시가격 합계액에 추가 연산 과정을 거쳐 과세표준 텍스트 박스에 금액이 나오는 버튼  else if (bt.Name.Equals(btn2.Name))  {  stand = (long)((pub - 600000000) \* 0.8);  cc.setStand(stand);  stndBox.Text = String.Format("{0:#,###}", stand);  }  //세율을 입력하는 버튼  else if (bt.Name.Equals(btn3.Name))  {  if (perBox.Text.Length == 0)  {  MessageBox.Show("세율을 입력해주세요", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  per = double.Parse(perBox.Text);  cc.setPer1(per);  }  }  //나이 선택시 해당하는 공제율을 추가적으로 연산에 넣어준다.  else if (bt.Name.Equals(btn4.Name))  {  int select;  select = comboBox1.SelectedIndex;  if (select == 0)  {  cc.setPer2(0.1);  }  else if (select == 1)  {  cc.setPer2(0.2);  }  else if (select == 2)  {  cc.setPer3(0.3);  }    }  //해당하는 보유기간에 따른 공제율을 추가적으로 연산에 넣어준다.  else if (bt.Name.Equals(btn5.Name))  {  int select;  select = comboBox2.SelectedIndex;  if (select == 0)  {  cc.setPer3(0.2);  }  else if (select == 1)  {  cc.setPer3(0.4);  }  }  //산출세액 계산하는 버튼  else if (bt.Name.Equals(btn6.Name))  {  result = cc.totalResult();  resultBox.Text = String.Format("{0:#,###}", result);  }  }    //메인메뉴로 돌아가기위한 폼 종료 이벤트  private void button1\_Click(object sender, EventArgs e)  {  Close();  }    }  }  **[compCal.cs]**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace TransferTax  {  class compCal  {  long pub, stand, result, minus1, minus2;  double per1, per2, per3;  public compCal()  {  pub = 0;  stand = 0;  result = 0;  minus1 = 0;  minus2 = 0;  per1 = 0.0;  per2 = 0.0;  per3 = 0.0;  }  public void setPub(long pub)  {  this.pub = pub;  }  public void setStand(long s)  {  stand = s;  }  public void setPer1(double p)  {  per1 = p;  }  public void setPer2(double p)  {  per2 = p;  minus1 = (long)(result \* per2); //1세대1주택자의 연령별에 따른 공제액  }  public void setPer3(double p)  {  per3 = p;  minus2 = (long)(result \* per3); //1세대1주택자의 보유기간별에 따른 공제액  }  public long totalResult()  {  result = (long)(stand \* per1);  result = result - minus1 - minus2;  return result;  }  }  }  **[exitForm.cs]**  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 TransferTax  {  public partial class exitForm : Form  {  public exitForm()  {  InitializeComponent();  }    //"적게 일하고 돈 많이 벌으러 나가기" 버튼 눌렀을때  private void button1\_Click(object sender, EventArgs e)  {  Application.Exit();  }  }  } | | | |
|  | | | |
| **Ⅲ. 구현 결과** | | | |
| **[startForm.cs]**  1. 프로그램을 실행하면 세금계산의 시작 폼이 등장    1-1. **세금계산** 버튼 클릭시 메인 메뉴 폼 등장    1-2. **종료** 버튼 클릭시 종료를 물어보는 메시지 박스가 등장    **[menuForm.cs]**  2. 메인 메뉴에서 계산하고자 하는 세금의 종류를 선택하여 클릭    **[achiForm.cs]**  3. 메인 메뉴에서 **취득세** 버튼 클릭 시 취득세를 계산할 수 있는 폼이 등장    3-1. 취득가액은 **상세입력** 버튼을 클릭하면 취득가액과 그 외에 포함된 금액을 상세하게 입력할 수 있는 폼이 등장.  **[achiDetailForm.cs]**    3-2. 취득가액과 취득가액 외 포함된 항목을 체크하여 금액을 입력해준다.  **총액계산** 버튼을 눌러 총 취득가액의 결과가 나온다. 맨 하단의 **입력하기** 버튼을 누르면 이전 폼에 취득가액이 입력된다.    3-3. 세율은 목록에서 해당되는 취득유형을 선택하여 **입력** 버튼을 누름으로써 취득유형에 따른 세율이 저장된다.    3-4. **입력** 버튼을 누름과 동시에 취득가액과 세율을 계산하여 산출세액이 나오게 된다.    **[TransForm.cs]**  4. 메인 메뉴 폼에서 **양도소득세** 버튼 클릭 시 등장    **[acquDetailForm.cs]**  4-1. 취득가액을 입력시 **상세입력** 버튼을 입력하여 취득가액과 취득가액 외의 금액을 입력할 수 있는 폼이 등장    .  4-2. 취득가액을 입력하고, 취득가액 외에 포함된 것이 있다면 **체크**하고 해당 금액을 입력한 뒤에 **총액계산** 버튼을 누르면 총 취득가액이 나온다.    4-3. 맨 하단의 **입력하기** 버튼을 누르면 이전 폼에서의 취득가액의 금액이 입력된다.    4-4. 필요경비까지 금액을 입력 후, 양도차익의 **결과** 버튼을 클릭하면 양도차익이 계산되어 나온다.    4-5. 공제율은 해당 주택의 보유기간을 **선택**하여 **입력** 버튼을 누르면, 보유기간에 따른 공제율로 입력되어 장기보유특별공제액을 산출한다.    4-6. 장기보유특별공제액까지 구했으면 양도소득금액을 구할 수 있으므로 **결과** 버튼을 눌러주어 양도소득금액이 나오게 한다.    4-7. 양도소득금액에서 양도소득기본공제액을 차감해준 과세표준 값을 산출하기 위해 **결과** 버튼 클릭해준다.    4-8. 세율은 해당 자산에 대한 등기여부와 보유기간을 고려하여 **선택**하여 **입력** 버튼을 눌러주면 그에 따른 세율이 적용되어 산출세액을 알 수 있게 된다.    **[compForm.cs]**  5. 메인 메뉴에서 **종합부동산세** 버튼 클릭 시 등장    5-1. 주택 공시가격 합계액을 **입력**한 후 **결과** 버튼을 누르면 과세표준을 알 수 있다.  해당 세율을 **입력**하여 산출 세액이 나오게 해준다.    5-2. 만약 1세대 1주택자인 경우에는 해당 부분을 **체크**하면 목록이 뜨는데 해당하는 것을 선택하면 공제율이 입력된다. 공제율을 알면 공제액을 구할 수 있기에 **결과** 버튼 클릭시 기존에 있는 산출세액에 공제액을 공제해준다.        **[exitForm.cs]**  6. 메인 메뉴에서 **종료** 버튼 클릭 시 종료를 물어보는 메시지박스가 뜬다.    6-1. **확인** 버튼을 클릭시 “이용해주셔서 감사합니다.” 라는 폼이 등장    6-2. **적게 일하고 많이 벌으러 나가기** 버튼 클릭시 프로그램 종료가 된다. | | | |
|  | | | |
| **Ⅳ. 기대효과** | | | |
| 세금계산에 대해 모르는 일반 사람들이 세무사에게 세금 계산을 맡기지 않고 스스로 세금을 계산하여 국세청에 납입하게 됨으로써 세무사에게 내야 할 수수료를 아낄 수 있으며, 세무사에게 맡기더라도 세무사가 계산한 것에 대한 금액과 자신이 계산한 금액을 비교하여 세무사의 계산이 맞는지 틀린 지 확인 할 수 있다.(간혹 가다 세무사도 사람이기에 실수하는 경우가 발생한다.) | | | |
|  | | | |
| **Ⅴ. 참고자료** | | | |
| - 박문각 공인중개사 2차 기본서 - 부동산 세법  - 박문각 공인중개사 2차 요약집 - 부동산 세법 | | | |