

# Programming 2 Report

---



과목명                      Programming 2

학과                         산업공학과

학번                         201611145

이름                         박민성

제출 일자                   2019.12.07

지도 교수님                윤장혁 교수님

## 1. 프로젝트 진행 과정

### 프로젝트 초기

2019년 9월 30일 경부터 개인과제 프로젝트를 진행했습니다. 처음에는 내부적인 코드의 흐름과 GUI가 잘 어울릴 수 있도록 코딩하는게 어려웠지만, 10,11월을 거치면서 실습과 수업을 통해서 차차 하나씩 알아가면서 개인과제를 진행했고, 모르는 것들은 구글링이나 교수님 pdf를 보면서 프로젝트를 진행했습니다.

### 프로젝트 막바지

겹치는 기능들을 하나로 표현하고, 객체로 만들어서 사용했습니다. 확실히 1학기 때 파이썬 언어를 다룰 때 보다 중간에 객체로 표현 할 수 있는 것들이 많았고, 수정이 용이했습니다.

## 중점 사항

개인과제 가이드라인 pdf를 보고 특허 검색 사이트들을 많이 참고하였습니다. 실제로 어떻게 검색이 이루어 지는 과정과 유사하게 코드를 짜려고 노력했습니다.

실제로 검색기능에서 중요한 게 무엇일까? 생각했습니다.

## 2. 코드 설명 및 출력 화면 설명

### 메인 기능 설명



### 1. main 폼

- 1) 다른 검색 기능의 form과 연결해주는 기능
- 2) 빠른 검색 시 검색어를 받아서 form 간 데이터 전송

<초기화면>

## 2. quick\_search 폼

<처음 실행 시 모습>

출원일	특허명	초록	발명가	CPCs
2015.06.11	휴대용 세...	본 발명은 ...	김민석(대...	B60S 3/04...
2010.02.03	기능성 티...	본 발명은 ...	김민영(대...	A41D 1/04...
2012.07.20	광고 노출...	본 발명의 ...	김민혜(대...	G06Q 30/0...
2011.12.30	뇌성마비 ...	본 발명은 ...	김민영(대...	C12Q 1/68...
2010.12.13	반도체 발...	본 발명은 ...	김민호(대...	H01L 29/12...
2015.06.17	녹용을 합...	본 발명은 ...	김민애(대...	A61K 2236...
2013.09.24	TDD 기반 ...	TDD(time ...	김민규(대...	H04L 1/182...
2014.10.21	진폭신호의...	본 발명은 ...	김민수(대...	H04L 27/34
2014.12.23	공기방파제...	공기방파제...	구성철(대...	E02B 3/06
2014.03.10	부유식 가...	본 발명은 ...	김민석(대...	F24F 6/00...
2014.08.26	마찰전기에...	마찰전기에...	강윤성(대...	H02N 1/04...
2015.11.02	토치각도조...	본 발명은 ...	김민주(대...	B23K 37/0...
2014.09.25	동체만의 ...	본 발명에 ...	김민규(대...	B64F 5/00
2015.06.25	매거진 세...	본 발명은 ...	김경만(대...	C11D 1/72...
2012.12.28	알루미늄계...	본 발명은 ...	김민길(대...	C02F 1/28...
2011.08.19	천장 부착...	본 발명은 ...	김민식(대...	F21S 8/04...
2014.05.12	GDI 엔진의...	본 발명은 ...	김민수(대...	F02D 1/16...
2014.10.22	이차 전지...	본 발명은 ...	김민수(대...	H01M 10/0...
2016.03.04	옥외용 비...	본 발명은 ...	김광영(대...	H01P 1/32...
2014.12.24	재생형 바...	본 발명은 ...	곽지혜(대...	H01M 14/0...
2015.02.23	입체영상의...	본 발명은 ...	김민(대한...	G05D 1/02...
2013.07.29	경화성 조...	본 출원은 ...	강대호(대...	C08G 77/0...
2014.04.08	태양광 모...	태양광 모...	김민혁(대...	H01L 31/04...

### 1) 검색 기능

main 폼에서 받아온 검색어를 보고 싶은 항목(등록신청번호, 출원일, 특허명, 초록, 발명가, CPCs)만 검색해주는 기능

출원일	특허명	초록	발명가	CPCs	출원일	특허명	초록	발명가	CPCs
2015.06.11	휴대용 세...	본 발명은 ...	김민석(대...	B60S 3/04...	2014.05.12	GDI 엔진의...	본 발명은 ...	김민수(대...	F02D 1/16...
2010.02.03	기능성 티...	본 발명은 ...	김민영(대...	A41D 1/04...	2014.09.25	동체만의 ...	본 발명에 ...	김민규(대...	B64F 5/00...
2012.07.20	광고 노출 ...	본 발명의 ...	김민혜(대...	G06Q 30/0...	2014.03.10	부유식 가...	본 발명은 ...	김민석(대...	F24F 6/00...
2011.12.30	뇌성마비 ...	본 발명은 ...	김민영(대...	C12Q 1/68...	2014.10.21	진폭신호의...	본 발명은 ...	김민수(대...	H04L 27/34...
2010.12.13	반도체 말...	본 발명은 ...	김민호(대...	H01L 23/12...	2012.12.28	알루미늄계 ...	본 발명은 ...	김민길(대...	C02F 1/28...
2015.06.17	녹용을 함...	본 발명은 ...	김민애(대...	A61K 2236...					
2013.09.24	TDD 기반 ...	TDD/time ...	김민규(대...	H04L 1/18...					
2014.10.21	진폭신호의...	본 발명은 ...	김민수(대...	H04L 27/34...					
2014.12.23	공기방파제...	공기방파제...	구성철(대...	E02B 3/06...					
2014.03.10	부유식 가...	본 발명은 ...	김민석(대...	F24F 6/00...					
2014.08.26	마찰전기예...	마찰전기예...	강운성(대...	H02N 1/04...					
2015.11.02	토치라도조...	본 발명은 ...	김민준(대...	B23K 37/0...					
2014.09.25	동체만의 ...	본 발명에 ...	김민규(대...	B64F 5/00...					
2015.06.25	매거진 세...	본 발명은 ...	김경만(대...	C11D 1/72...					
2012.12.28	알루미늄계 ...	본 발명은 ...	김민길(대...	C02F 1/28...					
2011.08.19	현장 부착...	본 발명은 ...	김민석(대...	F21S 8/04...					
2014.05.12	GDI 엔진의...	본 발명은 ...	김민수(대...	F02D 1/16...					
2014.10.22	미차 견지 ...	본 발명은 ...	김민수(대...	H01M 10/0...					
2016.03.04	육외용 배...	본 발명은 ...	김광영(대...	H01P 1/32...					
2014.12.24	재생형 방...	본 발명은 ...	박지혜(대...	H01M 14/0...					
2015.02.23	입체영상의...	본 발명은 ...	김민(대한...	G05D 1/02...					
2013.07.29	경화성 조...	본 출원은 ...	강대호(대...	C08G 77/0...					
2014.04.08	대양활 모...	대양활 모...	김민현(대...	H01L 31/04...					

## 2) 장바구니 기능

리스트 더블클릭, 버튼 이용 자유롭게 추가, 삭제 가능

등록신청번호 : 1020120000000

출원 날짜 : 2012.12.28

특허명 : 알루미늄계 슬러지 자원화 시스템

초록 : 본 발명은 총인 처리시설에서 발생하는 총인슬러지 및 먹는 물 처리과정에서 발생하는 정수슬러지(상수오니)를 이용한 알루미늄계 슬러지 자원화 시스템에 관한 것으로 4대 강 등을 오염시키는 총인을 정화하여 정화 후 총인 처리시설에서 발생하는 총인 슬러지 및 먹는 물 처리과정에서 발생하는 정수슬러지(상수오니)를 재활용함으로써 슬러지 처리에 따른 2차 오염을 막고 폐기물의 활용에 따른 경제적 이익을 창출하고 정화된 설비를 통해 알루미늄계 슬러지(예컨대 총인슬러지 및 상수오니 등)의 공급에서부터 흡착제의 제조라인을 자동화(일부는 수동도 가능)로 제조하여 양질의 흡착제를 다량으로 생산함을 목적으로 한다. 본 발명에 의한 총인 처리시설에서 발생하는 총인 슬러지 및 먹는 물 처리과정에서 발생하는 정수슬러지(상수오니)를 이용한 알루미늄계 슬러지 자원화 시스템은 산의 첨가를 통해 알루미늄과 규사가 용출된 알루미늄계 슬러지(예컨대 총인슬러지 및 상수오니 등)와 인산의 수열합성반응을 유도하여 제올라이트 반응물(알포계 및 사포계 제올라이트 및 무정형상태의 베르사이트와 실리카겔의 반응물)을 생성하는 반응부(100)와 상기 반응부에서 반응된 반응물을 탈수하는 탈수기(200)와 상기 탈수기에서 탈수된 반응물을 건조하며 흡착제를 성형하는 건조기(300)와 상기 건조기를 통해 건조된 흡착제를 분말로 분쇄하여 분말형 흡착제를 제조하는 분쇄기(400)와 상기 건조를 통해 건조된 흡착제 또는 상기 분쇄기를 통해 분쇄된 분말형 흡착제를 입상형으로 제조하는 펠렛기(500)로 구성된다.

발명인 : 김민길(대한민국);김상중(대한민국);김우규(대한민국);손무정(대한민국);연경호(대한민국);이기형(대한민국);이병진(대한민국);이윤규(대한민국);조재범(대한민국);최돈혁(대한민국);한승철(대한민국)

CPCs : C02F 1/285;C02F 11/12;Y02W 10/20

닫기

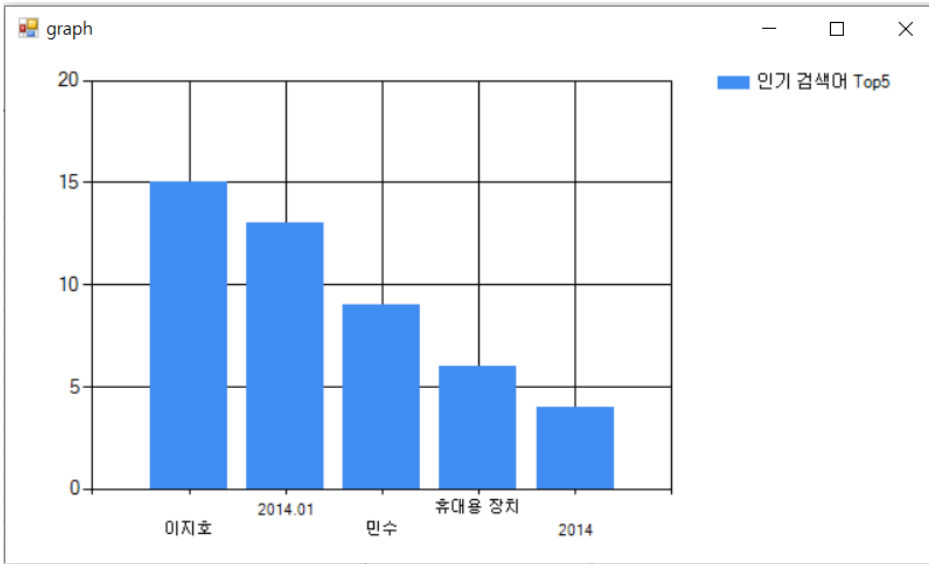
김민현(대... H01L 31/04...

자세히 보기

장바구니 저장

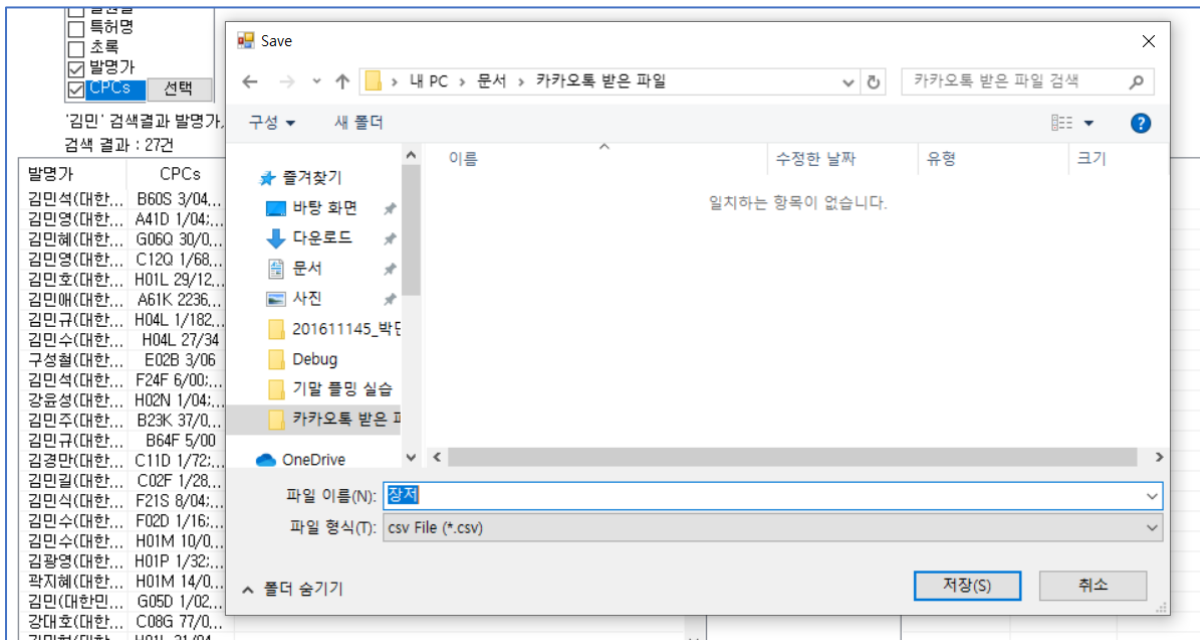
## 3)자세히 보기 기능

리스트 내 항목을 클릭 후, 자세히 보기 버튼을 누르면 검색어는 파란색으로 처리되고 내용의전문이 실행됨 (panel 이용)



#### 4) 통계 자료 제시(graph form 실행)

가장 많은 검색어 top 5를 제공해서 pc 사용자 사이에서 어떠한 검색어가 가장 인기가 많은지 제공



#### 5) csv 저장 기능

Listview의 item들을 csv 형태로 저장

### 3. category\_search 품

**카테고리별 검색**

등록신청번호  출원일

특허명  초록


발명가  CPCs

카테고리에 맞게 검색해주세요

등록신청번호	출원일	특허명	초록	발명가	CPCs

등록신청번호	출원일	특허명	초록	발명가	CPCs

<처음 실행시 모습>

 카테고리별 검색

등록신청번호 
 출원일

특허명 
 초록

발명가 
 CPCs

다음은 [등록신청번호 : 2014 , 특허명 : 부유식 , ] 가 포함된 검색 결과입니다.

등록신청번호	출원일	특허명	초록	발명가	CPCs
1020140000000	2014.12.23	공기방파제 ...	공기방파...	구성철<대...	E02B 3/06
1020140000000	2014.03.10	부유식 가습기	본 발명...	김민석<대...	F24F 6/00;...

[추가](#)

### 1) 카테고리 별 검색

각각 항목 별 원하는 검색어를 넣어 겹치는 자료들을 출력해 준다.

## 2) 장바구니 기능, 3) csv 저장 기능

### 3. presice\_search 품

[illegible]

<처음 실행시 모습>

### 1) 정밀 검색 기능(검색어 보정)

**정밀 검색**

☐ 특허명    ☒ 발명가

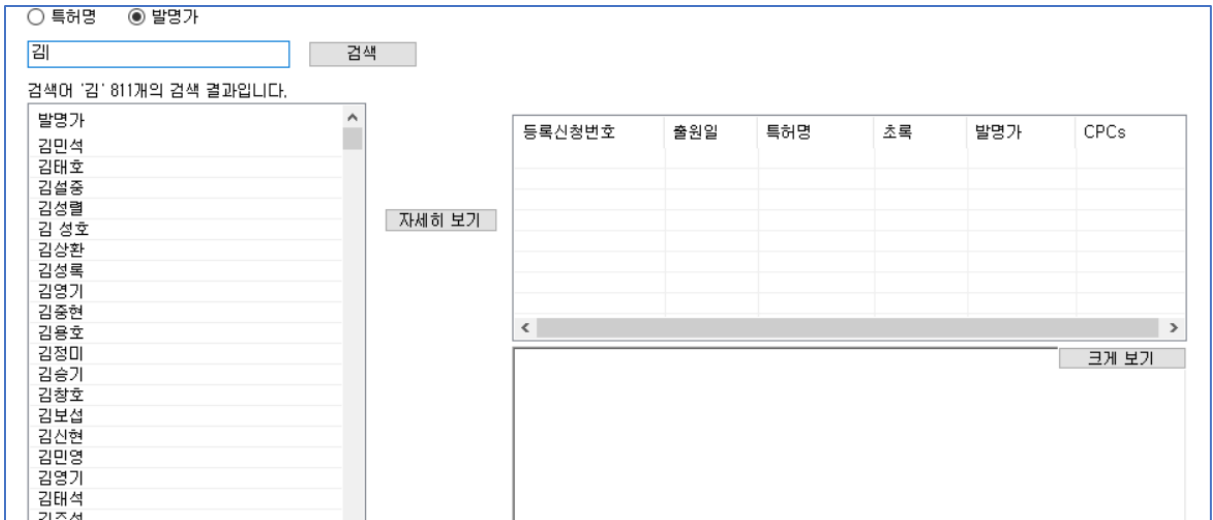
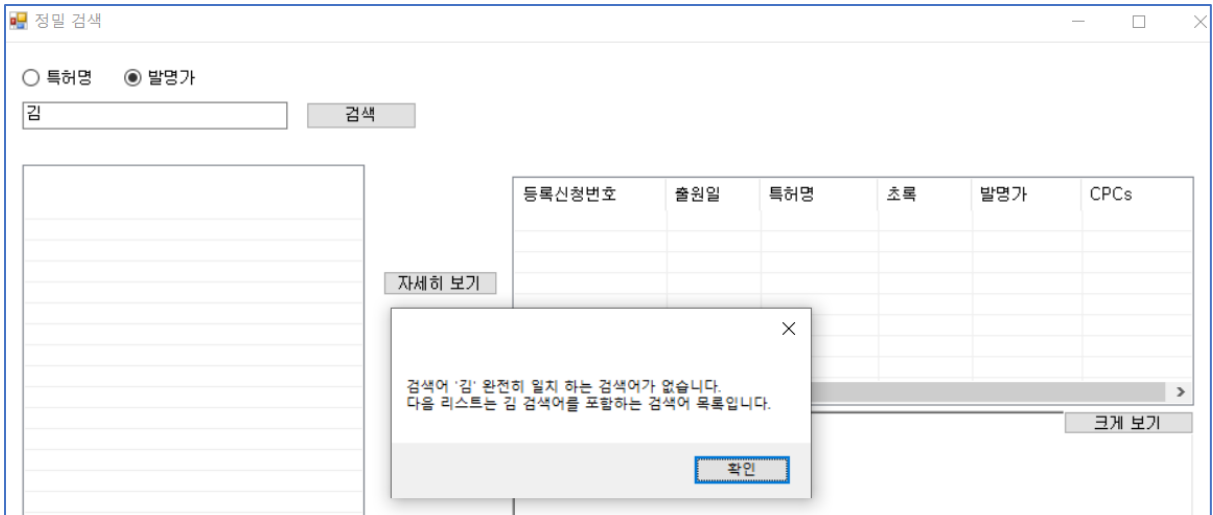
김민      검색

'김민' 정밀 검색 1개의 검색 결과

등록신청번호	출원일	특허명	초록	발명가	CPCs
10201500000000	2015.02.23	입체영상의 ...	본 발명...	김민(대한...	G05D 1/024...

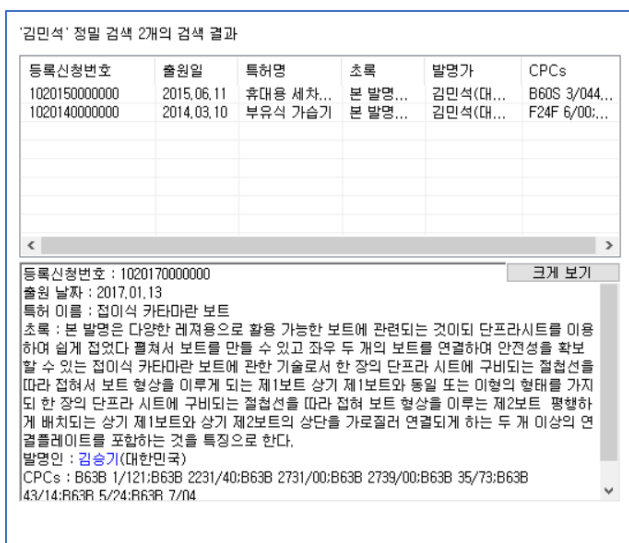
자세히 보기

### 1. 검색어가 완전히 일치 할 때



## 2. 검색어가 완전히 일치하지 않을때

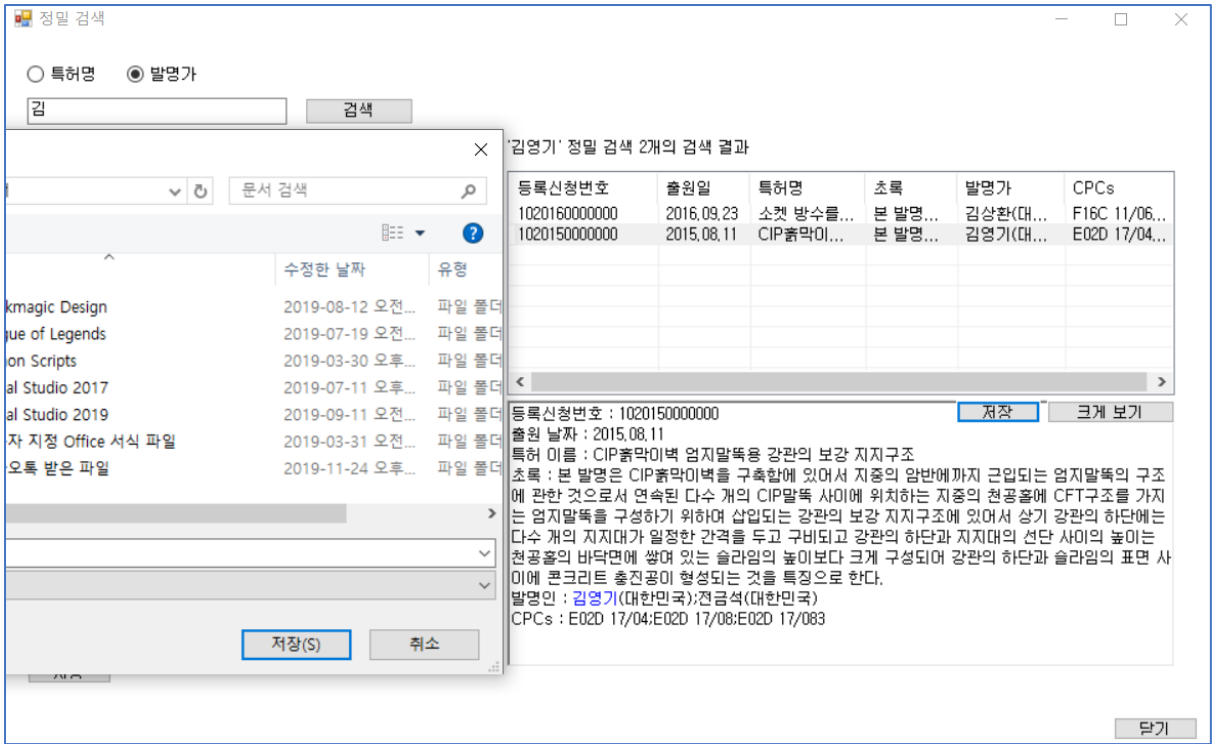
그 검색어가 포함하는 검색어가 존재한다면, 검색어를 포함하는 검색어를 추천해준다.



## 2) 이름, 특허 별 특허 목록 제공

왼쪽 리스트에서 이름을 클릭시 그 사람에 해당 되는 특허목록만 출력

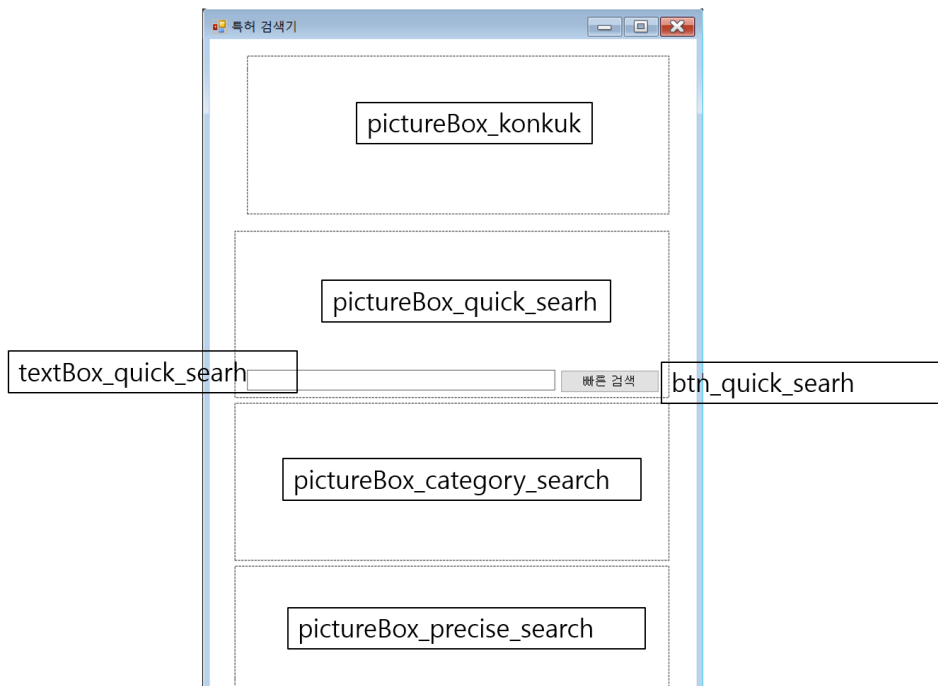




### 3) 자세히 보기 기능 제공

### 4) csv 저장 기능 제공

## 1) Main Form



```

15 public main()
16 {
17     InitializeComponent();
18     pictureBox_category_search.Load("./카테고리검색.png");
19     pictureBox_category_search.SizeMode = PictureBoxSizeMode.StretchImage;
20
21     pictureBox_quick_search.Load("./빠른검색.png");
22     pictureBox_quick_search.SizeMode = PictureBoxSizeMode.StretchImage;
23
24     pictureBox_precise_search.Load("./정밀검색.png");
25     pictureBox_precise_search.SizeMode = PictureBoxSizeMode.StretchImage;
26
27     pictureBox_konkuk.Load("./konkuk.jpg");
28     pictureBox_konkuk.SizeMode = PictureBoxSizeMode.StretchImage;
29 }

```

15 ~ 29번째 줄 : picture Box에 이미지를 넣어주고, 이미지 사이즈를 각각 picture Box와 맞춰줍니다. Main Form에는 4개의 picture box를 사용했습니다.

```

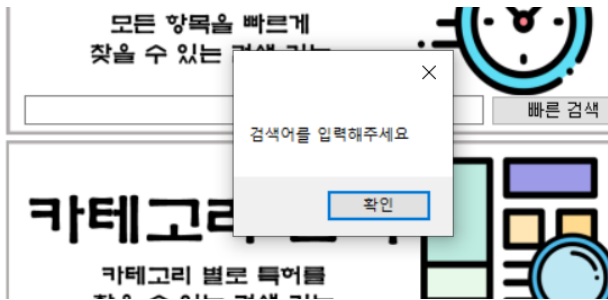
31 private void Btn_quicksearch_Click(object sender, EventArgs e)
32 {
33     if (textBox_quicksearch.Text != "")
34     {
35         quick_search Quick_search = new quick_search();
36         Quick_search.Search_word = textBox_quicksearch.Text;
37         Quick_search.ShowDialog();
38     }
39     else
40     {
41         MessageBox.Show("검색어를 입력해주세요");
42     }
43 }

```

31 ~ 43번째 줄: 검색어가 있다면 바로 Quick\_search form이 실행 됩니다.

textBox의 text가 없으면 검색어를 입력해주세요 라는 MessageBox가 나오면서 Quick\_search form이 실행되지 않게 했습니다.

또한 textBox의 Text를 Quick\_search 폼 객체로 넘겨줘서 검색어 정보를 활용 할 수 있게 해줍니다.



```

44 private void TextBox_quicksearch_KeyDown(object sender, KeyEventArgs e)
45 {
46     if (e.KeyCode == Keys.Enter)
47     {
48         Btn_quicksearch_Click(sender, e);
49     }
50 }

```

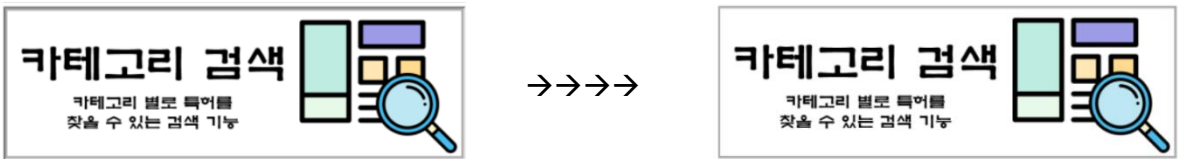
44~ 50번째줄: quicksearch 버튼을 누르지 않아도 textBox에 검색어를 입력한후 Enter Key를 누르면 버튼을 누르는 것과 같은 효과를 주었습니다.

```

51 private void PictureBox_category_search_MouseMove(object sender, MouseEventArgs e)
52 {
53     pictureBox_category_search.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
54 }
55 참조 1개
56 private void PictureBox_category_search_MouseLeave(object sender, EventArgs e)
57 {
58     pictureBox_category_search.BorderStyle = System.Windows.Forms.BorderStyle.None;
59 }
60 참조 1개
61 private void PictureBox_precise_search_MouseMove(object sender, MouseEventArgs e)
62 {
63     pictureBox_precise_search.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
64 }
65 참조 1개
66 private void PictureBox_precise_search_MouseLeave(object sender, EventArgs e)
67 {
68     pictureBox_precise_search.BorderStyle = System.Windows.Forms.BorderStyle.None;
69 }

```

51 ~ 66번째줄: 마우스가 그림위에 있는 것과 없는 것의 차이를 두어서 인터페이스의 편의를 제공했습니다. 마우스가 PictureBox에 올라가면 BorderStyle이 Fixed3D로 바뀌고 마우스가 떠나면 None으로 바뀝니다.



```

67 private void PictureBox_category_search_MouseClick(object sender, MouseEventArgs e)
68 {
69     category_search Category_search = new category_search();
70     Category_search.Show();
71 }
72 참조 1개
73 private void PictureBox_precise_search_MouseClick(object sender, MouseEventArgs e)
74 {
75     precise_search Precise_search = new precise_search();
76     Precise_search.Show();
77 }
78 }
79 }

```

67 ~ 79번째줄: pictureBox\_category\_search 를 클릭하면 category\_search 폼이 실행되고 pictureBox\_precise\_search 를 클릭하면 precise\_search 폼이 실행되게 하였습니다.

## 2) quick\_search Form

빠른 검색으로 각 항목에 '검색어'가 포함되어 있으면 모두 찾아줍니다.

The screenshot displays the '빠른 검색' (Quick Search) form. At the top, a header bar contains the title '빠른 검색'. Below it, a checkbox list titled 'checkBoxList\_Header' includes options: '등록신청번호', '출원일', '특허명', '초록', '발명가', and 'CPCs'. A '선택' (Select) button is next to the list. A 'btn\_add' button is positioned to the right of the list. Below the list, a text prompt reads '↑ 원하시는 항목을 클릭해주세요' (Click the item you want) followed by '빠른 검색' (Quick Search). The main area is a large list box labeled 'listView\_search'. To the right of the list box are three buttons: 'btn\_plus' (labeled '추가' - Add), '삭제' (Delete), and 'btn\_minus'. At the bottom of the list box are two buttons: 'btn\_all\_save' (labeled '모두 저장' - Save All) and 'btn\_carefully' (labeled '자세히 보기' - View Details). The bottom of the form features a '통계 초기화' (Reset Statistics) button, a 'btn\_born' button, a '통계 자료 보기' (View Statistics Data) button, a 'btn\_tonggae' button, a 'listView\_save' list box, a 'btn\_close' button (labeled '닫기' - Close), a 'btn\_back' button, a 'btn\_spool' button, a '장바구니 저장' (Save Cart) button, and a '초기화면으로 돌아가기' (Return to Home Screen) button. A 'richTextBox1' is located on the left side of the form.

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.IO;

```

## 1 ~ 10번째 줄

10번째 줄 -> StreamReader, Directory, File 을 사용 하기 위해 using 해준다.

```

12 namespace _201611145_박민성final
13 {
14     참조 4개
15     public partial class quick_search : Form
16     {
17         private List<string> app_num;
18         private List<string> app_date;
19         private List<string> title;
20         private List<string> abstract_;
21         private List<string> inventors;
22         private List<string> CPCs;
23         private string search_word;
24         public List<int> Num_save = new List<int>();

```

## 12 ~ 23번째 줄

16~21번째 줄 -> 각 csv 파일의 열에 있는 정보를 각각따로 리스트에 담아주기 위해서 전역변수로 설정했습니다.

22번째 줄 -> Main Form에서 받은 검색어를 저장해줍니다.

23번째 줄-> 검색어를 포함하는 정보의 인덱스를 저장하기 위해 설정

```

24     public string Search_word
25     {
26     set
27     {
28         this.search_word = value;
29     }
30     get
31     {
32         return search_word;
33     }
34 }

```

## 24 ~ 34번째 줄

Set- get을 통해서 Main Form에서 받아온 검색어를 저장해줍니다.

```

35 public quick_search()
36 {
37     InitializeComponent();
38     StreamReader sr = new StreamReader("../개인과제\data.csv", Encoding.Default, true);
39     List<string> app_num = new List<string>();
40     this.app_num = app_num;
41     List<string> app_date = new List<string>();
42     this.app_date = app_date;
43     List<string> title = new List<string>();
44     this.title = title;
45     List<string> abstract_ = new List<string>();
46     this.abstract_ = abstract_;
47     List<string> inventors = new List<string>();
48     this.inventors = inventors;
49     List<string> CPCs = new List<string>();
50     this.CPCs = CPCs;
51     while (!sr.EndOfStream)
52     {
53         string line = sr.ReadLine();
54         string[] data = line.Split(',');
55         app_num.Add(data[0]);
56         app_date.Add(data[1]);
57         title.Add(data[2]);
58         abstract_.Add(data[3]);
59         inventors.Add(data[4]);
60         CPCs.Add(data[5]);
61     }
62     sr.Close();
63 }

```

### 35 ~ 63번째 줄

38번째 줄 -> StreamReader로 개인과제 csv를 불러옵니다.

39~50번째 줄 -> 각 리스트 생성, 전역변수 저장.

51~62번째 줄 -> 한줄씩 읽어서 각 데이터를 리스트에 저장해줍니다.

### Btn\_add\_Click 64 ~ 235번째 줄

항목을 선택하고 검색을 시작하는 이벤트입니다.

```

64 private void Btn_add_Click(object sender, EventArgs e)
65 {
66     listView_search.Columns.Clear();
67     listView_search.Items.Clear();
68     listView_save.Columns.Clear();
69     listView_save.Items.Clear();
70     Num_save.Clear();
71     for (int i = 0; i < inventors.Count(); i++)
72     {
73         if (inventors[i].IndexOf(Search_word) != -1 || title[i].IndexOf(Search_word) != -1
74             || app_date[i].IndexOf(Search_word) != -1 || abstract_[i].IndexOf(Search_word) != -1 || CPCs[i].IndexOf(Search_word) != -1
75             || app_num[i].IndexOf(Search_word) != -1)
76         {
77             Num_save.Add(i);
78         }
79         else
80             continue;
81     }
82     if (Num_save.Count == 0)
83     {
84         MessageBox.Show("'" + Search_word + "' 를 포함하는 검색어가 존재하지 않습니다. 초기화면으로 돌아갑니다.");
85         this.Close();
86     }

```

### 64 ~ 85번째 줄

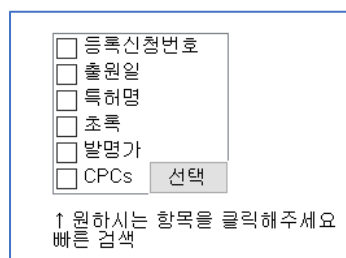
66 ~ 70번째 줄 -> 클릭마다 Num\_save와 Search listView, Save listView(장바구니)를 초기화 해줍니다. 저는 보고 싶은 항목만 볼수있게 클릭을 해서 그 항목들만 볼 수 있게 구현을 해서, 칼럼까지 초기화 해줍니다.

71 ~ 81번째줄 -> 검색어가 포함되어 있기만 한다면 인덱스를 Num\_save에 저장해줍니다.

82 ~ 86번째줄 -> 검색어가 모든 자료에 포함되어 있지 않다면, this.Close()로 Main Form으로 돌아갑니다.

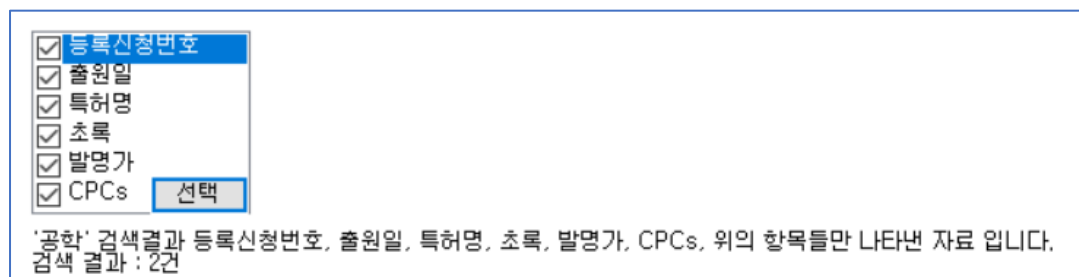
```
87 else if (Num_save.Count != 0)
88 {
89     foreach (string i in checkedListBox_header.CheckedItems)
90     {
91         listView_search.Columns.Add(i);
92         listView_save.Columns.Add(i);
93     }
94     if (checkedListBox_header.CheckedItems.Count != 0)
95     {
96         string string_result = "" + Search_word + " 검색결과 ";
97         for (int x = 0; x < checkedListBox_header.CheckedItems.Count; x++)
98         {
99             string_result = string_result + checkedListBox_header.CheckedItems[x].ToString() + ", ";
100         }
101         string_result = string_result + "위의 항목들만 나타낸 자료 입니다.";
102         label_result.Text = string_result;
103         label_search_count.Text = "검색 결과 : " + Num_save.Count().ToString() + "건";
104     }
```

### 87 ~ 104번째줄



89 ~ 93번째줄 -> Check한 항목만 칼럼에 넣어줍니다.

94 ~ 104번째줄 -> checkBox에 체크가 되어있다면 검색 결과의 수를 알려줍니다.



```
105 if (checkedListBox_header.CheckedItems.Contains("등록신청번호"))
106 {
107     foreach (int i in Num_save)
108     {
109         ListViewItem search = new ListViewItem((double.Parse(app_num[i])).ToString());
110         if ((checkedListBox_header.CheckedItems.Contains("출원일")))
111         {
112             search.SubItems.Add(app_date[i]);
113         }
114         if ((checkedListBox_header.CheckedItems.Contains("특허명")))
115         {
116             search.SubItems.Add(title[i]);
117         }
118         if ((checkedListBox_header.CheckedItems.Contains("초록")))
119         {
120             search.SubItems.Add(abstract_[i]);
121         }
122         if ((checkedListBox_header.CheckedItems.Contains("발명가")))
123         {
124             search.SubItems.Add(inventors[i]);
125         }
126         if ((checkedListBox_header.CheckedItems.Contains("CPCs")))
127         {
128             search.SubItems.Add(CPCs[i]);
129         }
130         listView_search.Items.Add(search);
131     }
132 }
```

### 105 ~ 132번째줄

체크 되는 항목이 가변적이기 때문에 어떻게 처리를 할지 고민을 하다가 코드가 복잡해지고 길어졌습니다. 등록 신청번호가 있을 때, ListViewItem 객체 search에 app\_num과 인덱스 i를 이용해서 정보를 넣어줍니다. 또 subitem에는 각각의 항목들을 넣어줍니다.

```
133     else if (checkedListBox_header.CheckedItems.Contains("출원일"))
134     {
135         foreach (int i in Num_save)
136         {
137             ListViewItem search = new ListViewItem(app_date[i]);
138             if ((checkedListBox_header.CheckedItems.Contains("특허명")))
139             {
140                 search.SubItems.Add(title[i]);
141             }
142             if ((checkedListBox_header.CheckedItems.Contains("초록")))
143             {
144                 search.SubItems.Add(abstract_[i]);
145             }
146             if ((checkedListBox_header.CheckedItems.Contains("발명가")))
147             {
148                 search.SubItems.Add(inventors[i]);
149             }
150             if ((checkedListBox_header.CheckedItems.Contains("CPCs")))
151             {
152                 search.SubItems.Add(CPCs[i]);
153             }
154             listView_search.Items.Add(search);
155         }
156     }
```

```
157     else if (checkedListBox_header.CheckedItems.Contains("특허명"))
158     {
159         foreach (int i in Num_save)
160         {
161             ListViewItem search = new ListViewItem(title[i]);
162             if ((checkedListBox_header.CheckedItems.Contains("초록")))
163             {
164                 search.SubItems.Add(abstract_[i]);
165             }
166             if ((checkedListBox_header.CheckedItems.Contains("발명가")))
167             {
168                 search.SubItems.Add(inventors[i]);
169             }
170             if ((checkedListBox_header.CheckedItems.Contains("CPCs")))
171             {
172                 search.SubItems.Add(CPCs[i]);
173             }
174             listView_search.Items.Add(search);
175         }
176     }
```

```
177     else if (checkedListBox_header.CheckedItems.Contains("초록"))
178     {
179         foreach (int i in Num_save)
180         {
181             ListViewItem search = new ListViewItem(abstract_[i]);
182             if ((checkedListBox_header.CheckedItems.Contains("발명가")))
183             {
184                 search.SubItems.Add(inventors[i]);
185             }
186             if ((checkedListBox_header.CheckedItems.Contains("CPCs")))
187             {
188                 search.SubItems.Add(CPCs[i]);
189             }
190             listView_search.Items.Add(search);
191         }
192     }
```



```

193     else if (checkedListBox_header.CheckedItems.Contains("발명가"))
194     {
195         foreach (int i in Num_save)
196         {
197             ListViewItem search = new ListViewItem(inventors[i]);
198             if ((checkedListBox_header.CheckedItems.Contains("CPCs")))
199             {
200                 search.SubItems.Add(CPCs[i]);
201             }
202             listView_search.Items.Add(search);
203         }
204     }
205

```

```

206     else if (checkedListBox_header.CheckedItems.Contains("CPCs"))
207     {
208         foreach (int i in Num_save)
209         {
210             ListViewItem search = new ListViewItem(CPCs[i]);
211             listView_search.Items.Add(search);
212         }
213     }

```

### 133 ~ 213번째 줄

105~132번째 줄과 반복적인 로직이 사용됩니다.

```

214     else
215     {
216         MessageBox.Show("보고 싶은 항목을 체크해주세요");
217     }
218     for (int i = 0; i < listView_search.Columns.Count; i++)
219     {
220         listView_search.Columns[i].Width = 85;
221         listView_save.Columns[i].Width = 85;
222         listView_save.Columns[i].TextAlign = HorizontalAlignment.Center;
223         listView_search.Columns[i].TextAlign = HorizontalAlignment.Center;
224     }
225
226 }
227 if (checkedListBox_header.CheckedItems.Count != 0)
228 {
229     using (System.IO.StreamWriter file = new System.IO.StreamWriter("./통계.csv", true, System.Text.Encoding.GetEncoding("ks_c_5601-1987")))
230     {
231         file.WriteLine("{0}", Search_word);
232         file.Close();
233     }
234 }
235

```

### 214 ~ 235번째 줄

214 ~ 217번째 줄 -> checkBox에 체크된 항목이 없으면 메시지 박스가 나타납니다.

218 ~ 225번째 줄 -> 각각 리스트 뷰에 들어간 칼럼의 넓이와 정렬을 시켜줍니다.

227 ~ 235번째 줄 -> 검색어 순위자료로 쓰기 위해서 통계라는 csv 파일에 검색된 검색어를 저장해줍니다. 이 자료는 나중에 그래프 form에서 씁니다.

```

236     private void Btn_back_Click(object sender, EventArgs e)
237     {
238         this.Close();
239     }

```

### btn\_back\_Click

클릭시 현재 품을 닫아주는 기능을 하는 버튼 이벤트입니다.

## 236 ~ 239번째줄

닫기 버튼으로 클릭시 품이 닫히고 초기화면으로 돌아갑니다.

## btn\_plus\_Click 240~286번째줄

listView\_search(검색 결과)에서 장바구니에 항목을 추가할 때 쓰는 버튼 이벤트입니다.

```
240 private void Btn_plus_Click(object sender, EventArgs e)
241 {
242     if (listView_search.FocusedItem == null)
243     {
244         MessageBox.Show("추가 할 항목을 클릭해 주세요.");
245     }
246     else
247     {
248         try
249         {
250             int a = 0;
251             for (int i = 0; i < listView_save.Items.Count; i++)
252             {
253                 if (listView_save.Items[i].SubItems[0].Text == listView_search.SelectedItems[0].SubItems[0].Text &&
254                     listView_save.Items[i].SubItems[1].Text == listView_search.SelectedItems[0].SubItems[1].Text &&
255                     listView_save.Items[i].SubItems[2].Text == listView_search.SelectedItems[0].SubItems[2].Text &&
256                     listView_save.Items[i].SubItems[3].Text == listView_search.SelectedItems[0].SubItems[3].Text &&
257                     listView_save.Items[i].SubItems[4].Text == listView_search.SelectedItems[0].SubItems[4].Text &&
258                     listView_save.Items[i].SubItems[5].Text == listView_search.SelectedItems[0].SubItems[5].Text)
259                 {
260                     a = a + 1;
261                 }
262             }
263         }
264     }
```

## 240 ~ 264번째줄

242~245번째줄 -> listView에서 클릭된항목(Focus)가 없으면 "추가 할 항목을 클릭해 주세요"라는 메시지 박스가 실행됩니다.

246~ 264번째줄 -> 장바구니에 중복이 안되게 처리해줬습니다. 등록번호는 다 구분되지만 이름이 같은 경우도 있을 수도 있고, 발명의 이름이 같을 수도 있기 때문에 모든 항목이 같아야 같은 항목으로 처리해줬습니다. 간단하게 int a를 사용해서 코드를 짰습니다.

```

265         if (a == 0)
266         {
267             ListViewItem save = new ListViewItem(listView_search.FocusedItem.SubItems[0].Text);
268             for (int i = 1; i < listView_search.Columns.Count; i++)
269             {
270                 save.SubItems.Add(listView_search.FocusedItem.SubItems[i].Text);
271             }
272
273             listView_save.Items.Add(save);
274             listView_search.FocusedItem.Focused = false;
275         }
276         else
277         {
278             MessageBox.Show("이미 장바구니에 있는 항목입니다.");
279         }
280     }
281     catch (Exception)
282     {
283         MessageBox.Show("이미 장바구니에 있는 항목입니다.");
284     }
285 }
286

```

## 265 ~ 286번째줄

a가 0이라면 중복되지 않는 것처럼 처리해서 장바구니에 item을 add해줍니다. 여기서 catch를 해준 이유는 제가 column의 개수가 가변적이기 때문에 하나하나 코드를 짜면 코드가 길어진다고 생각해서 오류가 나면 장바구니에 들어가지 않게 설정해주었습니다.

## Btn\_minus\_Click

장바구니에서 item을 제거해줍니다.

```

287 private void Btn_minus_Click(object sender, EventArgs e)
288 {
289     if (listView_save.FocusedItem == null)
290     {
291         MessageBox.Show("삭제할목록을 체크해주세요.");
292     }
293     else if (listView_save.FocusedItem != null)
294     {
295         listView_save.FocusedItem.Remove();
296         if (listView_save.Items.Count != 0)
297         {
298             listView_save.FocusedItem.Focused = false;
299         }
300     }
301 }

```

## 287 ~ 301번째줄

장바구니에 선택된 item이 있다면 제거해줍니다.

296 ~ 299번째줄 -> 이 코드가 없으면 선택된 아이템을 한번 지우고 삭제버튼을 계속 클릭하면 하나씩 아이템이 지워집니다.

## Private void Save

ListView에 있는 item들을 csv파일로 저장해줍니다.

```
302 private void Save(ListView save)
303 {
304     if (save.Items.Count == 0)
305     {
306         MessageBox.Show("저장할 검색 기록이 없습니다.");
307     }
308     else
309     {
310         SaveFileDialog saveFileDialog1 = new SaveFileDialog();
311         saveFileDialog1.Filter = "csv File|*.csv";
312         saveFileDialog1.Title = "Save";
313         saveFileDialog1.ShowDialog();
314         if (saveFileDialog1.FileName != "")
315         {
316             System.IO.FileStream fs = (System.IO.FileStream)saveFileDialog1.OpenFile();
317             switch (saveFileDialog1.FilterIndex)
318             {
319                 case 1:
320                     string pprint = "";
321                     for (int j = 0; j < save.Columns.Count; j++)
322                     {
323                         pprint = pprint + save.Columns[j].Text + ", ";
324                     }
325                     pprint = pprint + "\n";
326                     Byte[] bBytes = Encoding.Default.GetBytes(pprint);
327                     fs.Write(bBytes, 0, bBytes.Length);
328                     for (int i = 0; i < save.Items.Count; i++)
329                     {
330                         string sTmp = "";
331                         for (int j = 0; j < save.Items[i].SubItems.Count; j++)
332                             sTmp += save.Items[i].SubItems[j].Text + ", ";
333                         sTmp += "\n";
334                         Byte[] Bytes = Encoding.Default.GetBytes(sTmp);
335                         fs.Write(Bytes, 0, Bytes.Length);
336                     }
337                     break;
338             }
339             fs.Close();
340         }
341     }
342 }
```

### 302 ~ 341번째줄

304 ~ 307번째줄 - > 아이템이 없으면 메시지를 띄워서 ListView가 비었다고 알려줍니다.

308 ~ 339번째줄 -> SaveFileDialog를 이용합니다. 선택하지 않아도 자동으로 csv로 저장되게 설정해줍니다.

ListView save에 item들을 csv형태로 저장해줍니다. Csv는 쉼표로 자료를 구분하기 때문에 한 행에 요소들을","로 합쳐줍니다.

## ListView\_search\_MouseDoubleClick, ListView\_save\_MouseDoubleClick

listview에 item을 직접 더블 클릭해도 장바구니 추가,삭제 버튼을 누르는 것처럼 작동되게 했습니다.

## Btn\_spool\_Click, Btn\_all\_save\_Click

위에 Save를 이용하여서 검색기록, 장바구니를 저장합니다.

```

343 private void ListView_search_MouseDoubleClick(object sender, MouseEventArgs e)
344 {
345     Btn_plus_Click(sender, e);
346 }
참조 1개
347 private void ListView_save_MouseDoubleClick(object sender, MouseEventArgs e)
348 {
349     Btn_minus_Click(sender, e);
350 }
참조 1개
351 private void Btn_spool_Click(object sender, EventArgs e)
352 {
353     Save(listView_save);
354 }
참조 1개
355 private void Btn_all_save_Click(object sender, EventArgs e)
356 {
357     Save(listView_search);
358 }

```

## Btn\_toggae\_Click

통계자료 (graph)폼을 띄울지 말지 결정해주는 버튼 이벤트입니다

```

359 private void Btn_tonggae_Click(object sender, EventArgs e)
360 {
361     StreamReader sr = new StreamReader("./통계.csv", Encoding.Default, true);
362     List<string> toggae = new List<string>();
363     while (!sr.EndOfStream)
364     {
365         string line = sr.ReadLine();
366         string[] data = line.Split(',');
367         toggae.Add(data[0]);
368     }
369     if (toggae.Count == 0)
370     {
371         MessageBox.Show("자료가 없습니다.");
372     }
373     else
374     {
375         graph toggae_garph = new graph();
376         toggae_garph.Show();
377     }
378     sr.Close();
379 }

```

## 359 ~ 379번째 줄

361 ~ 368번째 줄 -> 통계 csv 파일을 불러서 비었는지 확인합니다.

369 ~ 372번째 줄 -> 통계자료가 없으면 자료가 없다고 MessageBox를 띄워줍니다.

373 ~ 379번째 줄 -> 통계자료가 있으면 graph 폼을 띄워서 통계 자료를 보여줍니다.

## Btn\_carefully\_Click

특히 내용을 크게 보여주고 자세히 보여줍니다.

검색어를 모두 파랗게 처리해서 보기 좋게 제공합니다.

☒ 등록신청번호  
☒ 출원일  
☒ 특허명  
☒ 초록  
☒ 발명가  
☒ CPCs

'산업' 검색결과 등록신청번호, 출원일, 특허명  
 검색 결과 : 23건

등록신청번호	출원일	특허명
1020120094231	2012.08.28	미세회로 ...
1020117019287	2010.03.31	전원 시스...
1020130017695	2013.02.19	네트워크를...
1020150122328	2015.08.31	다공성 고...
1020120059417	2012.06.01	식품원료 ...
1020120146456	2012.12.14	레이저를 ...
1020130059548	2013.05.27	인쇄공정을...
1020140161801	2014.11.19	프로바이오...
1020150161372	2015.11.17	알칼리 도...
1020157013415	2014.08.06	산업용 로봇
1020120107917	2012.09.27	환경 친화...
1020140166293	2014.11.26	휴대용 표...
1020140074447	2014.06.18	가족사료 ...
1020150184249	2015.12.22	폐유와 폐...
1020110036497	2011.04.20	산업용 음...
1020110078570	2011.08.08	페어 글래...
1020140022913	2014.02.26	차광성이 ...
1020100004677	2010.01.19	휘발성 페...
1020160174727	2016.12.20	자기 가교...
1020160069704	2016.06.03	산업부산물...
1020100104863	2010.10.26	해조류 막...
1020150153643	2015.11.03	납축전지의...

등록신청번호 : 1020157013415  
 출원 날짜 : 2014.08.06  
 특허명 : **산업용 로봇**  
 초록 : 본 발명은 소정의 처리 공정 전의 기판이 탑재된 핸드와 상이한 핸드에 이 처리 공정 후의 기판을 탑재하는 것이 가능해도 구성을 간소화하는 것이 가능한 **산업용** 로봇을 제공한다. **산업용** 로봇(1)은 1개의 기판이 탑재되는 기판 탑재부(20)를 하나 갖는 제1 핸드부(21)와 복수개의 기판 탑재부(20)를 갖는 제2 핸드부(22)를 구비하는 제1 핸드(11) 및 제2 핸드(12)와 제1 핸드(11)가 그 선단측에 회동 가능하게 연결되는 제1 마암(13)과 제2 핸드(12)가 그 선단측에 회동 가능하게 연결되는 제2 마암(14)과 제1 마암(13)의 기단부측 및 제2 마암(14)의 기단부측이 회동 가능하게 연결되는 마암 지지부(15)를 구비하고 있다. 이 **산업용** 로봇(1)에서는 제1 핸드부(21)와 제2 핸드부(22)가 개별로 회동 가능하게 되어 있고 제1 마암(13)과 제2 마암(14)이 개별로 회동 가능하게 되어 있다.  
 발명인 : 가미노 시게유키(일본);기타하라 야스유키(일본)  
 CPCs : B65G 49/07;H01L 21/67742

닫기

```

380 private void Btn_carefully_Click(object sender, EventArgs e)
381 {
382     try
383     {
384         int k = 0;
385         richTextBox1.Clear();
386         if (listView_save.SelectedItems != null && listView_search.SelectedItems != null)
387         {
388             if (listView_search.SelectedItems != null)
389             {
390                 {
391                     k = Num_save[listView_search.SelectedIndex[0]];
392                 }
393             }
394             else if (listView_save.SelectedItems != null)
395             {
396                 k = Num_save[listView_save.SelectedIndex[0]];
397             }
398             string string_target = "등록신청번호 : " + double.Parse(app_num[k]).ToString() + "\n" +
399             "출원 날짜 : " + app_date[k] + "\n" +
400             "특허명 : " + title[k] + "\n" +
401             "초록 : " + abstract_[k] + "\n" +
402             "발명인 : " + inventors[k] + "\n" +
403             "CPCs : " + CPCs[k] + "\n";

```

```

403         int var_index = 0;
404         List<int> save_int = new List<int>();
405         for (int i = 0; i < string_target.Length; i++)
406         {
407             string_target.IndexOf(Search_word, var_index);
408             if (save_int.Contains(string_target.IndexOf(Search_word, var_index)) == false &&
409                 string_target.IndexOf(Search_word, var_index) != -1)
410             {
411                 save_int.Add(string_target.IndexOf(Search_word, var_index));
412             }
413             var_index = var_index + 1;
414         }
415         int target_length = Search_word.Length;
416         richTextBox1.AppendText(string_target);
417
418         foreach (int uu in save_int)
419         {
420             richTextBox1.Select(uu, target_length);
421             richTextBox1.SelectionColor = Color.Blue;
422         }
423         richTextBox1.Visible = true;
424         btn_close.Visible = true;
425         panel1.Visible = true;
426     }
427     else
428     {
429         MessageBox.Show("자세히 보고 싶은 항목을\n체크해주세요");
430     }
431 }
432 }
433 catch (Exception)
434 {
435     MessageBox.Show("항목을 먼저 체크해주세요");
436 }
437 }
438 }

```

### 380 ~ 438번째 줄

384 ~ 422번째 줄 -> 장바구니에서 클릭해서 자세히 보는 경우, 검색 결과에서 클릭해서 자세히 보는 경우 두 가지를 모두 다뤘습니다.

Num\_save에 리스트뷰에 들어가 있는 아이템들의 원래 인덱스를 저장해놔기 때문에 이를 응용했습니다.

String\_target을 richTextBox에 저장하고, 검색어(Search\_word의 길이를 이용해서 string의 인덱스를 찾고 그 길이만큼만 파란색으로 처리하고, 검색어가 하나만 포함된게 아닐수도 있기 때문에 반복문을 돌려서 모든 해당 검색어를 찾아서 파랗게 처리합니다.

423 ~ 425번째 줄 -> panel이 숨겨져 있기 때문에 이 버튼을 클릭하면 visible을 true로 바꿔줍니다.

버튼도 숨겨놔기 때문에 마찬가지로 보여줍니다.

### Btn\_close\_Click

클릭하게 되면 panel을 숨겨지게 만듭니다.

```

439     private void Btn_close_Click(object sender, EventArgs e)
440     {
441         panel1.Visible = false;
442     }

```

## Btn\_born\_Click\_1

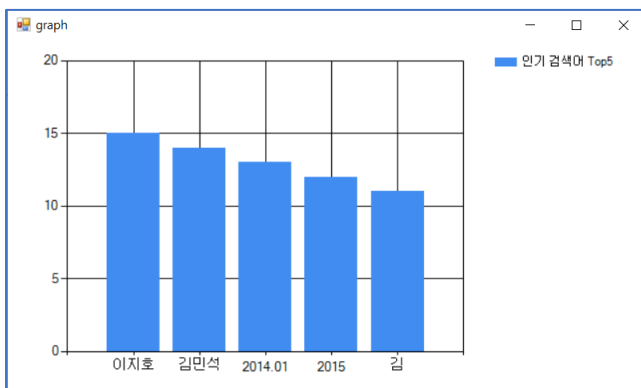
통계수치를 초기화 시켜줍니다.

```
443 private void Btn_born_Click_1(object sender, EventArgs e)
444 {
445     string file = "./";
446     string[] txtList = Directory.GetFiles(file, "./통계.csv");
447
448     foreach (string f in txtList)
449     {
450         File.Delete(f);
451     }
452     MessageBox.Show("통계 수치가 초기화 되었습니다.");
453     using (System.IO.StreamWriter file2 = new System.IO.StreamWriter("./통계.csv", true,
454         System.Text.Encoding.GetEncoding("ks_c_5601-1987")))
455     {
456     }
457 }
458 }
```

### 443 ~ 458번째 줄

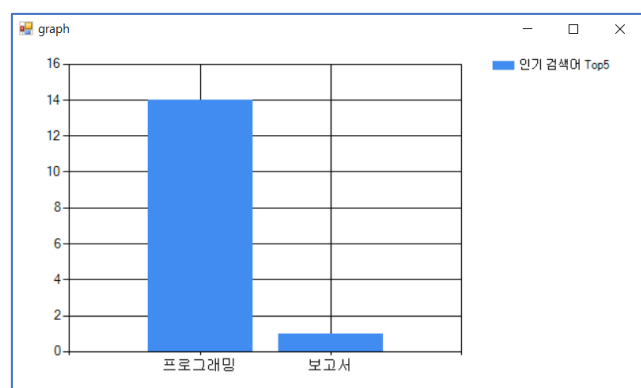
기존에 있던 통계.csv 파일을 삭제해줍니다.

File.Delete를 이용해서 간단하게 진행합니다.



<기존>

->



<삭제 후 재검색>



### 3) category\_search Form

카테고리별로 검색을 지원합니다.

```

14 public partial class category_search : Form
15 {
16     private List<string> app_num;
17     private List<string> app_date;
18     private List<string> title;
19     private List<string> abstract_;
20     private List<string> inventors;
21     private List<string> CPCs;
22     참조 1개
23     public category_search()
24     {
25         InitializeComponent();
26         StreamReader sr = new StreamReader("../개인과제\data.csv", Encoding.Default, true);
27         List<string> app_num = new List<string>();
28         this.app_num = app_num;
29         List<string> app_date = new List<string>();
30         this.app_date = app_date;
31         List<string> title = new List<string>();
32         this.title = title;
33         List<string> abstract_ = new List<string>();
34         this.abstract_ = abstract_;
35         List<string> inventors = new List<string>();
36         this.inventors = inventors;
37         List<string> CPCs = new List<string>();
38         this.CPCs = CPCs;
39         while (!sr.EndOfStream)
40         {
41             string line = sr.ReadLine();
42             string[] data = line.Split(',');
43             app_num.Add(data[0]);
44             app_date.Add(data[1]);
45             title.Add(data[2]);
46             abstract_.Add(data[3]);
47             inventors.Add(data[4]);
48             CPCs.Add(data[5]);
49         }
50     }
51 }

```

14 ~ 49번째 줄

Quicksearch와 동일하게 csv와 불러와서 data를 넣어줍니다.

## Btn\_category\_Click

검색 버튼입니다.

```
50 private void Btn_category_Click(object sender, EventArgs e)
51 {
52     listView_search.Items.Clear();
53     if (textBox_app_num.Text.Length + textBox_app_date.Text.Length + textBox_title.Text.Length +
54         textBox_inventors.Text.Length + textBox_abstract.Text.Length + textBox_CPCs.Text.Length == 0)
55     {
56         MessageBox.Show("검색어를 입력해주세요");
57     }
58     else
59     {
60         try
61         {
62             List<int> Num_save = new List<int>();
63             if (textBox_app_num.Text.Length == 0)
64             {
65                 for (int i = 0; i < inventors.Count(); i++)
66                 {
67                     if (inventors[i].IndexOf(textBox_inventors.Text) != -1 && title[i].IndexOf(textBox_title.Text) != -1
68                         && app_date[i].IndexOf(textBox_app_date.Text) != -1 && abstract_[i].IndexOf(textBox_abstract.Text) != -1 &&
69                         CPCs[i].IndexOf(textBox_CPCs.Text) != -1
70                         && app_num[i].IndexOf(textBox_app_num.Text) != -1)
71                     {
72                         Num_save.Add(i);
73                     }
74                     else
75                         continue;
76                 }
77             }
78             else
79             {
80                 for (int i = 0; i < inventors.Count(); i++)
81                 {
82                     if (inventors[i].IndexOf(textBox_inventors.Text) != -1 && title[i].IndexOf(textBox_title.Text) != -1
83                         && app_date[i].IndexOf(textBox_app_date.Text) != -1 && abstract_[i].IndexOf(textBox_abstract.Text) != -1
84                         && CPCs[i].IndexOf(textBox_CPCs.Text) != -1
85                         && app_num[i].IndexOf(double.Parse(textBox_app_num.Text).ToString()) != -1)
86                     {
87                         Num_save.Add(i);
88                     }
89                     else
90                         continue;
91                 }
92             }
93             foreach (int i in Num_save)
94             {
95                 ListViewItem save = new ListViewItem(double.Parse(app_num[i]).ToString());
96
97                 save.SubItems.Add(app_date[i]);
98                 save.SubItems.Add(title[i]);
99                 save.SubItems.Add(abstract_[i]);
100                 save.SubItems.Add(inventors[i]);
101                 save.SubItems.Add(CPCs[i]);
102                 listView_search.Items.Add(save);
103             }
104             string string_explain = ("다음은 [");
105             if (textBox_app_num.Text.Length != 0)
106             {
107                 string_explain = string_explain + "등록신청번호 : " + textBox_app_num.Text + " , ";
108             }
109             if (textBox_app_date.Text.Length != 0)
110             {
111                 string_explain = string_explain + "출원일 : " + textBox_app_date.Text + " , ";
112             }
113             if (textBox_title.Text.Length != 0)
114             {
115                 string_explain = string_explain + "특허명 : " + textBox_title.Text + " , ";
116             }
117             if (textBox_abstract.Text.Length != 0)
118             {
```

```

119         string_explain = string_explain + "초록 : " + textBox_abstract.Text + " , ";
120     }
121     if (textBox_inventors.Text.Length != 0)
122     {
123         string_explain = string_explain + "발명가 : " + textBox_inventors.Text + " , ";
124     }
125     if (textBox_CPOs.Text.Length != 0)
126     {
127         string_explain = string_explain + "CPOs : " + textBox_CPOs.Text;
128     }
129     string_explain += string_explain + "] 가 포함된 검색 결과입니다.";
130     label_explain.Text = string_explain + "검색 결과 : " + Num_save.Count().ToString() + " 건";
131 }
132 catch (Exception)
133 {
134     MessageBox.Show("등록신청 번호에는 숫자만 입력해주세요");
135 }
136 }
137 }

```

## 50 ~ 137번째줄

52 ~ 57번째줄 -> 검색어가 없으면 검색어를 입력해주세요 MessageBox가 출력됩니다.

58 ~ 92번째줄 -> quick search와 비슷하지만 여기서는 각각의 검색어를 각각 찾아서 검색어가 포함되어 있으면 Num\_save에 인덱스를 저장해줍니다.

93 ~ 131번째줄 -> Num\_save에 저장된 인덱스를 활용해서 listview에 검색 결과를 추가해주고 검색결과 수와 유형을 label에 각 각 넣어줍니다.

132 ~ 135번째줄 -> Exception Handling을 통해서 등록신청번호에 한글이 들어간다면 Message 박스에 등록신청 번호에는 숫자만 입력해주세요를 띄웁니다.

## Private void Save

Csv 저장 기능

```
138 private void Save(ListView save)
139 {
140     if (save.Items.Count == 0)
141     {
142         MessageBox.Show("저장할 검색 기록이 없습니다.");
143     }
144     else
145     {
146         SaveFileDialog saveFileDialog1 = new SaveFileDialog();
147         saveFileDialog1.Filter = "csv File|*.csv";
148         saveFileDialog1.Title = "Save";
149         saveFileDialog1.ShowDialog();
150         if (saveFileDialog1.FileName != "")
151         {
152             System.IO.FileStream fs = (System.IO.FileStream)saveFileDialog1.OpenFile();
153             switch (saveFileDialog1.FilterIndex)
154             {
155                 case 1:
156                     string pprint = "";
157                     for (int j = 0; j < save.Columns.Count; j++)
158                     {
159                         pprint = pprint + save.Columns[j].Text + ", ";
160                     }
161                     pprint = pprint + "\n";
162                     Byte[] bBytes = Encoding.Default.GetBytes(pprint);
163                     fs.Write(bBytes, 0, bBytes.Length);
164                     for (int i = 0; i < save.Items.Count; i++)
165                     {
166                         string sTmp = "";
167                         for (int j = 0; j < save.Items[i].SubItems.Count; j++)
168                             sTmp += save.Items[i].SubItems[j].Text + ", ";
169                         sTmp += "\n";
170                         Byte[] Bytes = Encoding.Default.GetBytes(sTmp);
171                         fs.Write(Bytes, 0, Bytes.Length);
172                     }
173                     break;
174             }
175             fs.Close();
176         }
177     }
178 }
```

138 ~ 178번째 줄

Quick search save와 동일하게 csv를 저장해주는 매서드입니다.

## Btn\_search\_save\_Click

## Btn\_save\_save\_Click

클릭시 save 매서드에 각각 listview를 넣어 진행해줍니다.

```
179 private void Btn_search_save_Click(object sender, EventArgs e)
180 {
181     Save(listView_search);
182 }
183 참조 1개
184 private void Btn_save_save_Click(object sender, EventArgs e)
185 {
186     Save(listView_save);
187 }
```

179 ~ 187번째 줄

## Btn\_plus\_Click

Btn\_plus 클릭시 장바구니 listview 로 항목들이 저장됩니다.

```
188 private void Btn_plus_Click(object sender, EventArgs e)
189 {
190     if (listView_search.FocusedItem == null)
191     {
192         MessageBox.Show("추가 할 항목을 클릭해 주세요.");
193     }
194     else
195     {
196         int a = 0;
197         for (int i = 0; i < listView_save.Items.Count; i++)
198         {
199             if (listView_save.Items[i].SubItems[0].Text == listView_search.SelectedItems[0].SubItems[0].Text &&
200                 listView_save.Items[i].SubItems[1].Text == listView_search.SelectedItems[0].SubItems[1].Text &&
201                 listView_save.Items[i].SubItems[2].Text == listView_search.SelectedItems[0].SubItems[2].Text &&
202                 listView_save.Items[i].SubItems[3].Text == listView_search.SelectedItems[0].SubItems[3].Text &&
203                 listView_save.Items[i].SubItems[4].Text == listView_search.SelectedItems[0].SubItems[4].Text &&
204                 listView_save.Items[i].SubItems[5].Text == listView_search.SelectedItems[0].SubItems[5].Text)
205             {
206                 a = a + 1;
207             }
208         }
209         if (a == 0)
210         {
211             ListViewItem save = new ListViewItem(listView_search.FocusedItem.SubItems[0].Text);
212             for (int i = 1; i < listView_search.Columns.Count; i++)
213             {
214                 save.SubItems.Add(listView_search.FocusedItem.SubItems[i].Text);
215             }
216             listView_save.Items.Add(save);
217             listView_search.FocusedItem.Focused = false;
218         }
219         else
220         {
221             MessageBox.Show("이미 장바구니에 있는 항목입니다.");
222         }
223     }
224 }
225
226 }
```

## 188 ~ 226번째 줄

Quick search의 추가 버튼과 동일한 기능을 합니다. 동일 항목이 저장되는 것을 막아주기 위해 장바구니에 있는 아이템들과 현재 선택된 항목을 비교해 줍니다. Quick search에서와 다른점은 column의 개수가 동일하기 때문에 칼럼의 수로 인한 인덱스 에러가 여기서는 발생하지 않기 때문에 고려하지 않았습니다.

## Btn\_minus\_Click

장바구니에서 항목을 제거해주는 버튼 클릭 이벤트입니다.

```
227 private void Btn_minus_Click(object sender, EventArgs e)
228 {
229     if (listView_save.FocusedItem == null)
230     {
231         MessageBox.Show("삭제할 목록을 체크해주세요.");
232     }
233     else if (listView_save.FocusedItem != null)
234     {
235         listView_save.FocusedItem.Remove();
236         if (listView_save.Items.Count != 0)
237         {
238             listView_save.FocusedItem.Focused = false;
239         }
240     }
241 }
```

## 227 ~ 241번째 줄

Quick search form 과 동일한 기능을 해줍니다. 장바구니에서 항목을 삭제해줍니다. 그리고 focus 또한 풀어줍니다.

```
242 private void ListView_search_MouseDoubleClick(object sender, MouseEventArgs e)
243 {
244     Btn_plus_Click(sender, e);
245 }
246 참조 1개
247 private void ListView_save_MouseDoubleClick(object sender, MouseEventArgs e)
248 {
249     Btn_minus_Click(sender, e);
250 }
251 참조 1개
```

## 242 ~ 249번째 줄

추가/삭제 버튼외에도 항목을 더블클릭하는걸로 추가/삭제 할 수 있게 구현했습니다.

**TextBox\_app\_num\_KeyDown TextBox\_app\_date\_KeyDown TextBox\_title\_KeyDown**

**TextBox\_abstract\_KeyDown TextBox\_inventors\_KeyDown TextBox\_CPCs\_KeyDown**

검색시에 엔터 버튼을 눌러도 검색버튼이 눌리는 것처럼 구현했습니다.

```
250 private void TextBox_app_num_KeyDown(object sender, KeyEventArgs e)
251 {
252     if (e.KeyCode == Keys.Enter)
253     {
254         Btn_category_Click(sender, e);
255     }
256 }
257 참조 1개
258 private void TextBox_app_date_KeyDown(object sender, KeyEventArgs e)
259 {
260     if (e.KeyCode == Keys.Enter)
261     {
262         Btn_category_Click(sender, e);
263     }
264 }
265 참조 1개
266 private void TextBox_title_KeyDown(object sender, KeyEventArgs e)
267 {
268     if (e.KeyCode == Keys.Enter)
269     {
270         Btn_category_Click(sender, e);
271     }
272 }
273 참조 1개
274 private void TextBox_abstract_KeyDown(object sender, KeyEventArgs e)
275 {
276     if (e.KeyCode == Keys.Enter)
277     {
278         Btn_category_Click(sender, e);
279     }
280 }
281 참조 1개
282 private void TextBox_inventors_KeyDown(object sender, KeyEventArgs e)
283 {
284     if (e.KeyCode == Keys.Enter)
285     {
286         Btn_category_Click(sender, e);
287     }
288 }
```

```

285     private void TextBox_CPCs_KeyDown(object sender, KeyEventArgs e)
286     {
287         if (e.KeyCode == Keys.Enter)
288         {
289             Btn_category_Click(sender, e);
290         }
291     }

```

250 ~ 291번째 줄

#### Btn\_close\_Click

Form을 닫고 Main Form으로 돌아갑니다.

```

292     private void Btn_close_Click(object sender, EventArgs e)
293     {
294         this.Close();
295     }
296 }
297

```

292 ~ 297번째 줄

```

297     private void Category_search_Load(object sender, EventArgs e)
298     {
299         for (int i = 1; i < inventors.Count; i++)
300         {
301             ListViewItem save = new ListViewItem(app_num[i]);
302
303             save.SubItems.Add(app_date[i]);
304             save.SubItems.Add(title[i]);
305             save.SubItems.Add(abstract_[i]);
306             save.SubItems.Add(inventors[i]);
307             save.SubItems.Add(CPCs[i]);
308             listView_search.Items.Add(save);
309         }
310     }
311 }
312

```

297 ~ 312번째 줄

전체 자료를 로드해줍니다. - 기본기능

#### 4) precise\_search Form

위에 두 폼과 달리 정확히 일치하는 특허명/사람이름을 찾아줍니다.

```

14 public partial class precise_search : Form
15 {
16     private List<string> app_num;
17     private List<string> app_date;
18     private List<string> title;
19     private List<string> abstract_;
20     private List<string> inventors;
21     private List<string> CPCs;
22     private string search_word;
23     public List<int> Num_save = new List<int>();
24     public List<int> Num_save2 = new List<int>();
25     참조 1개
26     public precise_search()
27     {
28         InitializeComponent();
29         StreamReader sr = new StreamReader("../개인 과제\data.csv", Encoding.Default, true);
30         List<string> app_num = new List<string>();
31         this.app_num = app_num;
32         List<string> app_date = new List<string>();
33         this.app_date = app_date;
34         List<string> title = new List<string>();
35         this.title = title;
36         List<string> abstract_ = new List<string>();
37         this.abstract_ = abstract_;
38         List<string> inventors = new List<string>();
39         this.inventors = inventors;
40         List<string> CPCs = new List<string>();
41         this.CPCs = CPCs;
42
43         while (!sr.EndOfStream)
44         {
45             string line = sr.ReadLine();
46             string[] data = line.Split(',');
47             app_num.Add(data[0]);
48             app_date.Add(data[1]);
49             title.Add(data[2]);
50             abstract_.Add(data[3]);
51             inventors.Add(data[4]);
52             CPCs.Add(data[5]);
53         }
54         sr.Close();
55     }
56 }

```



## 14 ~ 54번째 줄

기존의 폼들과 비슷하지만 Num\_save2가 추가되었습니다.

### Btn\_search\_Click

클릭시 검색이 일어나는 버튼 이벤트입니다.

```
55 private void Btn_search_Click(object sender, EventArgs e)
56 {
57     if (textBox_search.Text != "")
58     {
59         label_save2.Text = "";
60         label_save.Text = "";
61         listView_save2.Clear();
62         listView_save.Items.Clear();
63         Num_save.Clear();
64         Num_save2.Clear();
65         search_word = textBox_search.Text;
66         List<string> target_List = new List<string>();
67         List<string> save_inventors = new List<string>();
68         List<int> save_inventors_num = new List<int>();
69         List<int> save_inventors_num2 = new List<int>();
70         List<string> save_title = new List<string>();
71         if (radio_btn_inventors.Checked == true || radio_btn_title.Checked == true)
72         {
73             if (radio_btn_inventors.Checked == true)
74             {
75                 for (int i = 0; i < inventors.Count; i++)
76                 {
77                     if (inventors[i].IndexOf(";") != -1)
78                     {
79                         string[] ss = inventors[i].Split(';');
80                         for (int ii = 0; ii < ss.Length; ii++)
81                         {
82                             int save = ss[ii].IndexOf('(');
83                             string uu = "";
84                             for (int jj = 0; jj < save; jj++)
85                             {
86                                 uu += ss[ii][jj];
87                             }
88                             save_inventors.Add(uu);
89                             save_inventors_num.Add(i);
90                         }
91                     }
92                 }
93             }
94         }
95     }
96 }
```

```

92         else if (inventors[i].IndexOf(";") == -1)
93         {
94             int save = inventors[i].IndexOf('(');
95             string uu = "";
96             for (int jj = 0; jj < save; jj++)
97             {
98                 uu = uu + inventors[i][jj];
99             }
100             save_inventors.Add(uu);
101             save_inventors_num.Add(i);
102         }
103     }
104     target_List = save_inventors;
105     for (int i = 0; i < target_List.Count; i++)
106     {
107         if (target_List[i] == search_word)
108         {
109             Num_save.Add(save_inventors_num[i]);
110             save_inventors_num2.Add(save_inventors_num[i]);
111         }
112         else if (target_List[i].IndexOf(search_word) != -1)
113         {
114             Num_save2.Add(i);
115         }
116     }
117 }
118 else if (radio_btn_title.Checked == true)
119 {
120     target_List = title;
121     for (int i = 0; i < target_List.Count(); i++)
122     {
123         if (target_List[i] == search_word)
124         {
125             Num_save.Add(i);
126         }
127     }

```

```

128     for (int i = 0; i < target_List.Count(); i++)
129     {
130         if (target_List[i].IndexOf(search_word) != -1)
131         {
132             Num_save2.Add(i);
133         }
134         else
135         {
136             continue;
137         }
138     }
139 }
140 if (Num_save.Count != 0)
141 {
142     if (radio_btn_title.Checked == true)
143     {
144         foreach (int i in Num_save)
145         {
146             ListViewItem save = new ListViewItem(double.Parse(app_num[i]).ToString());
147             save.SubItems.Add(app_date[i]);
148             save.SubItems.Add(title[i]);
149             save.SubItems.Add(abstract_[i]);
150             save.SubItems.Add(inventors[i]);
151             save.SubItems.Add(CPCs[i]);
152             listView_save.Items.Add(save);
153         }
154     }
155     else if (radio_btn_inventors.Checked == true)
156     {
157         foreach (int i in save_inventors_num2)
158         {
159             ListViewItem save = new ListViewItem(double.Parse(app_num[i]).ToString());
160             save.SubItems.Add(app_date[i]);
161             save.SubItems.Add(title[i]);
162             save.SubItems.Add(abstract_[i]);
163             save.SubItems.Add(inventors[i]);
164             save.SubItems.Add(CPCs[i]);
165             listView_save.Items.Add(save);
166         }

```

```

167     }
168     label_save2.Text = "";
169     label_save.Text = "" + search_word + "' 정밀 검색 " + listView_save.Items.Count.ToString() + "개의 검색 결과";
170 }
171 else if (Num_save.Count == 0)
172 {
173     if (Num_save2.Count != 0)
174     {
175         MessageBox.Show("검색어 '" + search_word + "' 완전히 일치 하는 검색어가 없습니다.다음 리스트는 '" + search_word + "' " +
176             "검색어를 포함하는 검색어 목록입니다.");
177         try
178         {
179             if (radio_btn_inventors.Checked == true)
180             {
181                 listView_save2.Columns.Add("발명가", 300);
182                 for (int i = 0; i < Num_save2.Count; i++)
183                 {
184                     int g = 0;
185                     for (int j = 0; j < listView_save2.Items.Count; j++)
186                     {
187                         if (listView_save2.Items[j].SubItems[0].Text.ToString() == target_List[Num_save2[i]].ToString())
188                         {
189                             g++;
190                         }
191                     }
192                     if (g == 0)
193                     {
194                         ListViewItem save = new ListViewItem(target_List[Num_save2[i]]);
195                         listView_save2.Items.Add(save);
196                     }
197                 }
198             }
199         }
200     }
201 }

```

```

202     else if (radio_btn_title.Checked == true)
203     {
204         listView_save2.Columns.Add("특허명", 300);
205         for (int i = 0; i < Num_save2.Count; i++)
206         {
207             if (save_title.IndexOf(title[Num_save2[i]]) == -1)
208             {
209                 save_title.Add(title[Num_save2[i]]);
210             }
211         }
212         foreach (string i in save_title)
213         {
214             ListViewItem save = new ListViewItem(i);
215             listView_save2.Items.Add(save);
216         }
217     }
218     label_save2.Text = "검색어 '" + search_word + "' " + listView_save2.Items.Count.ToString() + "개의 검색 결과입니다.";
219 }
220 catch (Exception ex)
221 {
222     MessageBox.Show(ex.ToString());
223 }
224 }
225 else if (Num_save2.Count == 0)
226 {
227     MessageBox.Show("'" + search_word + "' 검색결과가 없습니다.");
228 }
229 }
230 }
231 else
232 {
233     MessageBox.Show("항목을 체크해주세요.");
234 }
235 }
236 else
237 {
238     MessageBox.Show("검색어를 입력해주세요.");
239 }
240 }

```

## 55 ~ 240번째줄

기본적으로 Radio 버튼이 특허명, 발명가명일때 따로 돌아가게 구현했습니다.

57 ~ 70번째줄 - > 버튼이 눌릴때마다 listview, Num\_save, Num\_save2를 초기화 해주고 라벨의 text도 초기화 해 줍니다.

71 ~ 117번째줄 -> radio 버튼이 발명가로 체크 됐을 때 기본적으로 발명가의 csv 자료 구조는 김민석(대한민국);

박명득(대한민국); 이거나 김민석(대한민국)이어서 이름만 따로 떼어내는 작업을 하였습니다. ;과 (을 기준으로 자료를 스플릿 했습니다.

118 ~ 139번째줄 -> radio 버튼이 특허명일 때 돌아가게했습니다. 일치한다면 Num\_save에 넣고 그 특허명을 포함하는 검색어는 Num\_save2로 넣어서 검색어 교정의 기능을 수행하는데 기초를 다지는 코드입니다.

140 ~ 170번째줄 -> 완전일치 검색어가 있을 때 바로 오른쪽 편에 있는 listview에 자료의 정보가 띄우게 합니다.

171 ~ 240번째줄 -> 완전일치 검색어가 없다면 MessageBox를 띄운 후 radio 버튼이 체크된 항목에 따라 그 검색어를 포함하는 자료들을 왼쪽 listview에 우선적으로 나타내줍니다.

### TextBox\_search\_KeyDown

검색시에 textbox에서 엔터키를 눌러도 검색키가 누르는 것처럼 작용하게 코드를 구성했습니다.

### Listview\_save2\_MouseDoubleClick

Listview에 있는 항목들을 더블클릭할 때 Btn\_plus이 작동하는 것 처럼 구현했습니다.

```
241 private void TextBox_search_KeyDown(object sender, KeyEventArgs e)
242 {
243     if (e.KeyCode == Keys.Enter)
244     {
245         Btn_search_Click(sender, e);
246     }
247 }
248 참조 1개
249 private void ListView_save2_MouseDoubleClick(object sender, MouseEventArgs e)
250 {
251     Btn_plus_Click(sender, e);
252 }
```

241 ~ 251번째줄

### Btn\_plus

왼쪽 리스트뷰에 있는 불일치 검색어들중 포함하는 자료들을 모두 오른쪽 리스트에서 보여줍니다.

☐ 특허명 ☒ 발명가

김 검색

검색어 '김' 811개의 검색 결과입니다.

발명가
김민석
김태호
김설중
김성렬
김 성호
김상환
김성록
김영기
김중현
김용호
김정미
김승기
김창호
김보섭
김신현

자세히 보기

'김신현' 정밀 검색 2개의 검색 결과

등록신청번호	출원일	특허명	초록	발명가	CPCs
1020140040989	2014.04.07	복수개의 내...	본 발명...	김신현(대...	B01F 3/08;...
1020130111602	2013.09.17	광결정을 구...	본 발명...	김신현(대...	G06K 9/00

저장

크게 보기

```

252 private void Btn_plus_Click(object sender, EventArgs e)
253 {
254     try
255     {
256         if (listView_save2.Items.Count != 0)
257         {
258             listView_save.Items.Clear();
259             if (radio_btn_inventors.Checked == true)
260             {
261                 string search_2 = listView_save2.SelectedItems[0].Text;
262                 for (int i = 0; i < inventors.Count; i++)
263                     if (inventors[i].IndexOf(search_2) != -1)
264                     {
265                         ListViewItem save = new ListViewItem(double.Parse(app_num[i]).ToString());
266                         save.SubItems.Add(app_date[i]);
267                         save.SubItems.Add(title[i]);
268                         save.SubItems.Add(abstract_[i]);
269                         save.SubItems.Add(inventors[i]);
270                         save.SubItems.Add(CPCs[i]);
271                         listView_save.Items.Add(save);
272                     }
273             }
274             else if (radio_btn_title.Checked == true)
275             {
276                 string search_2 = listView_save2.SelectedItems[0].Text;
277                 for (int i = 0; i < title.Count; i++)
278                     if (title[i].IndexOf(search_2) != -1)
279                     {
280                         ListViewItem save = new ListViewItem(double.Parse(app_num[i]).ToString());
281                         save.SubItems.Add(app_date[i]);
282                         save.SubItems.Add(title[i]);
283                         save.SubItems.Add(abstract_[i]);
284                         save.SubItems.Add(inventors[i]);
285                         save.SubItems.Add(CPCs[i]);
286                         listView_save.Items.Add(save);
287                     }
288             }
289         }

```

```

290     else
291     {
292         MessageBox.Show("항목을 체크해주세요.");
293     }
294     label_save.Text = "" + listView_save2.SelectedItems[0].Text + " 정밀 검색 " + listView_save.Items.Count.ToString() +
295     "개의 검색 결과";
296     search_word = listView_save2.SelectedItems[0].Text;
297 }
298 else
299 {
300     MessageBox.Show("리스트가 비어 있습니다.");
301 }
302 }
303 catch (Exception)
304 {
305     MessageBox.Show("리스트 중 하나를 선택해주세요.");
306 }
307 }

```

## 252 ~ 307번째줄

252 ~ 272번째줄 -> (발명가 radio button)왼쪽리스트뷰에서 선택된 아이템이 있다면 해당하는 항목의 정보를 오른쪽 리스트뷰에 나타내어 줍니다.

273 ~ 287번째줄 -> 240 ~ 260번째줄과 마찬가지로인 기능이지만 특허명일 때 작동합니다.

288 ~ 307번째줄 -> 항목의 개수와 어떤 항목을 눌렀는지 label에 나타내어 주고 오류와 예외를 처리해주었습니다.

## Btn\_carefully\_Click

앞 폼들과 같이 richTextBox에서 특허 자료를 자세히 보여준다.

```

308 private void Btn_carefully_Click(object sender, EventArgs e)
309 {
310     try
311     {
312         richTextBox1.Clear();
313
314         if (listView_save.SelectedItems != null)
315         {
316             string string_target = "등록신청번호 : " + listView_save.SelectedItems[0].Text + "\n" +
317             "출원 날짜 : " + listView_save.FocusedItem.SubItems[1].Text + "\n" +
318             "특허 이름 : " + listView_save.FocusedItem.SubItems[2].Text + "\n" +
319             "초록 : " + listView_save.FocusedItem.SubItems[3].Text + "\n" +
320             "발명인 : " + listView_save.FocusedItem.SubItems[4].Text + "\n" +
321             "CPCs : " + listView_save.FocusedItem.SubItems[5].Text + "\n";
322             int var_index = 0;
323             List<int> save_int = new List<int>();
324             for (int i = 0; i < string_target.Length; i++)
325             {
326                 string_target.IndexOf(search_word, var_index);
327                 if (save_int.Contains(string_target.IndexOf(search_word, var_index)) == false &&
328                     string_target.IndexOf(search_word, var_index) != -1)
329                 {
330                     save_int.Add(string_target.IndexOf(search_word, var_index));
331                 }
332                 var_index = var_index + 1;
333             }
334             int target_length = search_word.Length;
335             richTextBox1.AppendText(string_target);
336
337

```

```

338         foreach (int uu in save_int)
339         {
340             richTextBox1.Select(uu, target_length);
341             richTextBox1.SelectionColor = Color.Blue;
342         }
343         richTextBox1.Visible = true;
344     }
345     else
346     {
347         MessageBox.Show("자세히 보고 싶은 항목을 클릭해주세요");
348     }
349 }
350 catch (Exception)
351 {
352     MessageBox.Show("자세히 보고 싶은 항목을 클릭해주세요");
353 }
354 }

```

### 308 ~ 354번째 줄

검색어를 포함한 특허자료를 자세히 보여주고 파랗게 처리해준다. 앞품과 동일로직

### Listview\_save\_MouseDoubleClick

자세히 버튼을 안눌러도 오른쪽에 있는 리스트뷰에서 더블클릭으로 자료를 크게 볼 수 있다.

### Btn\_save\_Click

버튼 클릭 이벤트로 Csv 저장을 해줍니다.

```

355 private void ListView_save_MouseDoubleClick(object sender, MouseEventArgs e)
356 {
357     Btn_carefully_Click(sender, e);
358 }
359
360 참조 1개
361 private void Btn_save_Click(object sender, EventArgs e)
362 {
363     Save(listView_save);
364 }

```

355 ~ 363번째 줄

#### Private void Save

Csv 저장을 해줍니다. ( 앞 로직과 동일)

```

364 private void Save(ListView save)
365 {
366     if (save.Items.Count == 0)
367     {
368         MessageBox.Show("저장할 검색 기록이 없습니다.");
369     }
370     else
371     {
372         SaveFileDialog saveFileDialog1 = new SaveFileDialog();
373         saveFileDialog1.Filter = "csv File|*.csv";
374         saveFileDialog1.Title = "Save";
375         saveFileDialog1.ShowDialog();
376         if (saveFileDialog1.FileName != "")
377         {
378             System.IO.FileStream fs = (System.IO.FileStream)saveFileDialog1.OpenFile();
379             switch (saveFileDialog1.FilterIndex)
380             {
381                 case 1:
382                     string pprint = "";
383                     for (int j = 0; j < save.Columns.Count; j++)
384                     {
385                         pprint = pprint + save.Columns[j].Text + ", ";
386                     }
387                     pprint = pprint + "\n";
388                     Byte[] bBytes = Encoding.Default.GetBytes(pprint);
389                     fs.Write(bBytes, 0, bBytes.Length);
390                     for (int i = 0; i < save.Items.Count; i++)
391                     {
392                         string sTmp = "";
393                         for (int j = 0; j < save.Items[i].SubItems.Count; j++)
394                             sTmp += save.Items[i].SubItems[j].Text + ", ";
395                         sTmp += "\n";
396                         Byte[] Bytes = Encoding.Default.GetBytes(sTmp);
397                         fs.Write(Bytes, 0, Bytes.Length);
398                     }
399                     break;
400             }
401             fs.Close();
402         }
403     }
404 }

```

364 ~ 402번째 줄

오른쪽에 있는 listview의 정보를 저장해줍니다.

#### Btn\_save\_all\_Click

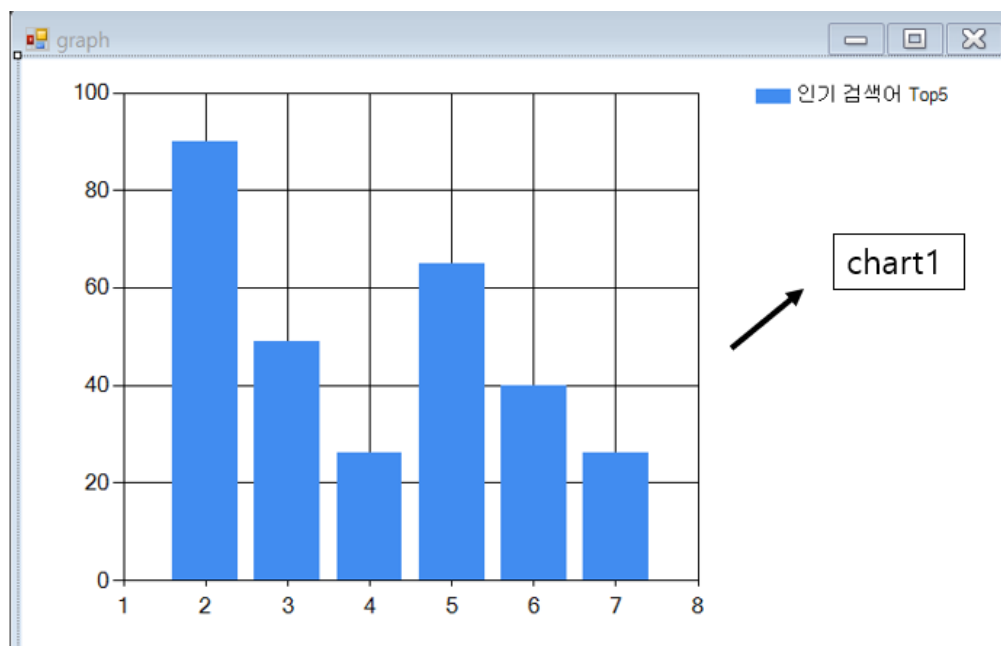
왼쪽에 있는 불일치 검색결과를 모두 저장해줍니다.

### Btn\_close\_Click

폼을 닫고 초기화면 (Main Form으로 돌아갑니다.)

```
406 private void Btn_save_all_Click(object sender, EventArgs e)
407 {
408     Save(listView_save2);
409 }
410
411 참조 1개
412 private void Btn_close_Click(object sender, EventArgs e)
413 {
414     this.Close();
415 }
416 }
417 }
```

## 5) graph Form





```

14 public partial class graph : Form
15 {
16     public Dictionary<string, int> save_dic = new Dictionary<string, int>();
17     참조 1개
18     public graph()
19     {
20         InitializeComponent();
21         StreamReader sr = new StreamReader("../통계.csv", Encoding.Default, true);
22         List<string> toggae = new List<string>();
23         while (!sr.EndOfStream)
24         {
25             string line = sr.ReadLine();
26             string[] data = line.Split(',');
27             toggae.Add(data[0]);
28         }
29         List<int> value_savelist = new List<int>();
30         List<string> key_savelist = new List<string>();
31         for (int j = 0; j < toggae.Count(); j++)
32         {
33             string b = toggae[j].ToString();
34             int c = 0;
35             for (int i = 0; i < toggae.Count(); i++)
36             {
37                 if (b == toggae[i].ToString())
38                 {
39                     c++;
40                 }
41             }

```

```

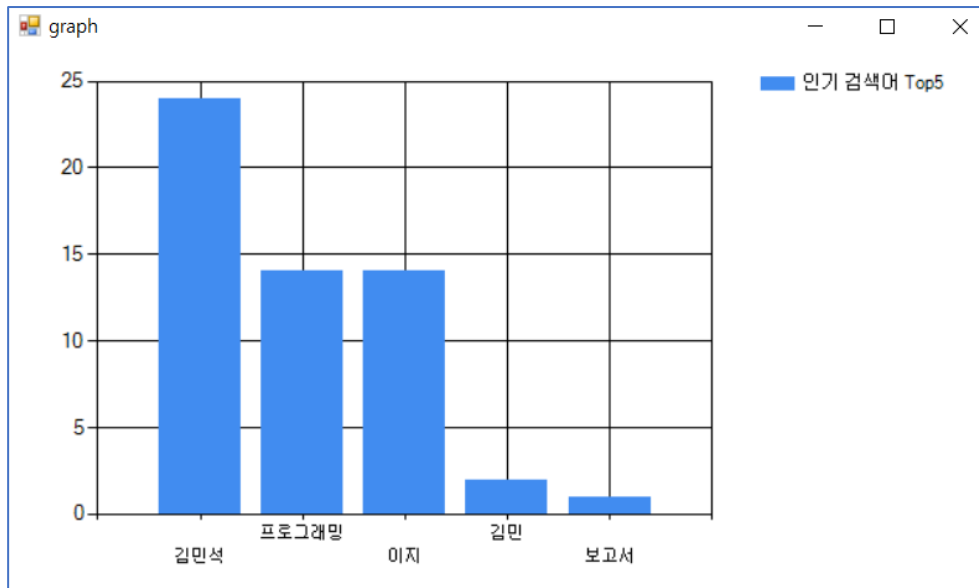
42             if (key_savelist.Contains(toggae[j]) == false)
43             {
44                 value_savelist.Add(c);
45                 key_savelist.Add(toggae[j]);
46             }
47         }
48         for (int ii = 0; ii < value_savelist.Count(); ii++)
49         {
50             save_dic.Add(key_savelist[ii], value_savelist[ii]);
51         }
52         int int_count = 0;
53         chart1.Series["인기 검색어 Top5"].Points.Clear();
54         foreach (var item in save_dic.OrderByDescending(i => i.Value))
55         {
56             chart1.Series["인기 검색어 Top5"].Points.AddXY(item.Key, item.Value);
57             int_count++;
58             if (int_count == 5)
59             { break; }
60         }
61         sr.Close();
62     }
63 }
64 }
65

```

## 14 ~ 65번째 줄

Quick\_search 폼에서 저장된 통계 csv 데이터를 가지고 인기검색어 top5 라는 그래프를 그리게 합니다. 한번 검색할때마다 하나의 데이터가 A열에 들어가게 되는데 그것의 개수를 세준후 많은 순서대로 정렬해서 상위 5개 항목만 그래프로 표현해줍니다.

	A1
	A
4	프로그래밍
5	프로그래밍
6	프로그래밍
7	프로그래밍
8	프로그래밍
9	프로그래밍
10	프로그래밍
11	프로그래밍
12	프로그래밍
13	프로그래밍
14	프로그래밍
15	보고서
16	김민
17	김민
18	김민석
19	김민석
20	김민석
21	김민석
22	김민석
23	김민석
24	김민석
25	김민석



### 3. 결론

C# 프로그래밍 수업을 완전히 듣고 개인과제를 시작한 것이 아니여서 코드에 불필요한 반복이 너무 많은 것 같습니다. Csv 파일에 대한 관리를 용이하게 할 수 있는 과정들을 학습하게 되어서 나중에도 많은 도움이 될 것 같습니다. 다시 이런 코드를 짤 수 있는 기회가 된다면 코드를 좀 더 간략하고 깔끔하게 짤 수 있도록 노력해야 될 것 같습니다.