C# LINQ 使用例

[C#][C# サンプルコード]

・国立国会図書館 API

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Reflection;
using System.Text;
using System.Threading.Tasks;
using System.Xml;
using System.Xml.Linq;

namespace Book
{
    class Program
    {
        static void Main(string[] args)
        {
            var p = new Program();
            if (args.Length < 2)
            {
                Console.WriteLine("must 2 arguments.");
                return;
            }
            string option = args[0];

            string xml = null;
            if (option == "-f")
            {
                xml = File.ReadAllText(args[1],Encoding.UTF8);
            }
            else if (option == "-k")
            {
                var param = new List<KeyValuePair<string, string>>();
                for (int i=1; i<args.Length; i++)
                {
                    string[] kv = args[i].Split('=');
                    param.Add(new KeyValuePair<string, string>(kv[0], kv?[1]));
                }
                xml = p.GetXml(param);
            }
            else if (option == "-t")
            {
                var param = new List<KeyValuePair<string, string>>();
                param.Add(new KeyValuePair<string, string>("title", "Python"));
                xml = p.GetXml(param);
            }
            var books = ParseXml(xml);

            LinqTest(books);
        }

        private static void LinqTest(List<Book> books)
        {

            // QueryExp(books);
            MethodExp(books);

        }

        /// <summary>
        /// メソッド構文
        /// </summary>
        /// <param name="books"></param>
        private static void MethodExp(List<Book> books)
        {
            var result = books
                        .Where( book => book.Title.IndexOf("計算") > 0 )
                        .OrderByDescending( book=> book.Title)
                        .Select( book => book.Title)

             ;
```

```
    foreach (string title in result)
    {
        Console.WriteLine(title);
    }
}

/// <summary>
/// クエリ構文
/// </summary>
/// <param name="books"></param>
private static void QueryExp(List<Book> books)
{
    var result = from book in books
                 where book.Title.IndexOf("計算") > 0
                 orderby book.Title descending
                 select book.Title
     ;

    // 遅延評価
    // books.ForEach(book => book.Title = book.Title.Replace("Python", "HOGE"));

    foreach (string title in result)
    {
        Console.WriteLine(title);
    }
}

private static List<Book> ParseXml(string xml)
{
    XNamespace xmlns = "http://www.loc.gov/zing/srw/";
    XNamespace xmlns2 = "info:srw/schema/1/dc-v1.1";
    XNamespace xmlns3 = "http://purl.org/dc/elements/1.1/";

    var root = XElement.Load(new XmlTextReader(new StringReader(xml)));

    var records = from record in root.Elements(xmlns + "records")
                                     .Elements(xmlns + "record")
                                     .Elements(xmlns + "recordData")
                                     .Elements(xmlns2 + "dc")
                  select record
                  ;

    var books = new List<Book>();
    foreach (XElement record in records)
    {

        var book = new Book();
        books.Add(book);

        foreach(PropertyInfo field in book.GetType().GetProperties())
        {
            foreach (var elms in record.Elements(xmlns3 + field.Name.ToLower()))
            {
                field.SetMethod?.Invoke(book, new object[] { elms.Value });
                break;
            }
        }

    }
    return books;
}

class Book
{
    public string Title { get; set; } = "";
    public string Creator { get; set; } = "";
    public string Subject { get; set; } = "";
    public string Publisher { get; set; } = "";
    public string Language { get; set; } = "";

    public override string ToString()
    {
        var buf = new StringBuilder();
        buf.Append($"{this.GetType().Name}{{");
        foreach(PropertyInfo p in  this.GetType().GetProperties())
        {
            buf.Append($"{p.Name}={p.GetValue(this)},");
        }

        buf.Append($"}}");
        return buf.ToString();
    }
```

```
        }

        public string GetUrl(List<KeyValuePair<string, string>> param)
        {
                                                                    string          url          =
"http://iss.ndl.go.jp/api/sru?operation=searchRetrieve&recordPacking=xml&query={0}";
            string qp = string.Join(" OR ",
                    param.Select(kvp => WebUtility.UrlEncode(String.Format("{0}=\"{1}\"", kvp.Key,
kvp.Value))));
            return string.Format(url, qp);
        }

        public string GetXml(List<KeyValuePair<string, string>> param)
        {
            string url = GetUrl(param);

            using (var client = new WebClient())
            {
                using (var reader = new StreamReader(client.OpenRead(url)))
                {
                    string xml = reader.ReadToEnd();
                    Console.WriteLine(xml);
                    return xml;
                }
            }
        }
    }
}
```