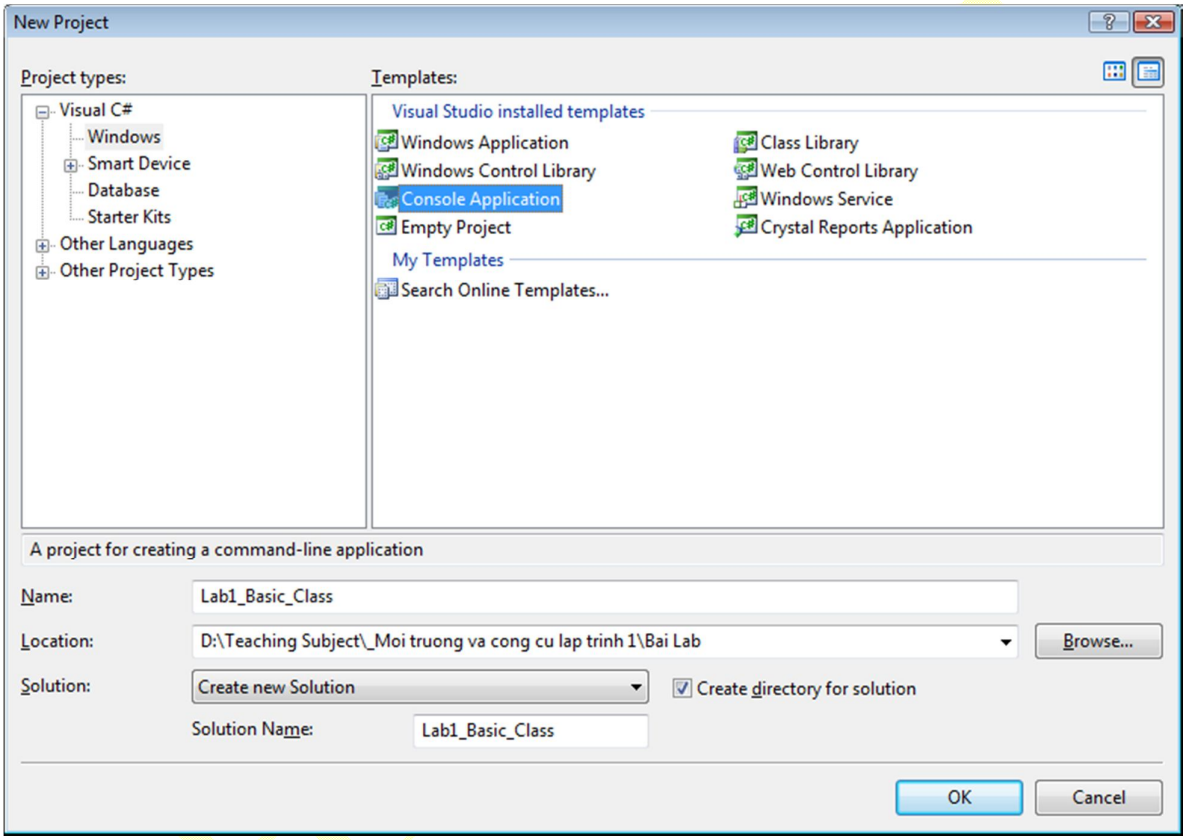
BÀI THỰC HÀNH SỐ 1 (4 tiết)

I. Mục tiêu:

- Xử lý trên chuỗi
- Sử dụng biểu thức chính quy
- Nhóm lớp các đối tượng

II. Hướng dẫn thực hành

Tạo một dự án Console Application mới, đặt tên là Lab1_Basic_Class



1. Thao tác trên chuỗi

Phương thức	Mục đích
Compare	So sánh nội dung của 2 chuỗi
CompareOrdinal	Giống compare nhưng không quan tâm đến ngôn ngữ
Format	Định dạng một chuỗi chứa 1 giá trị khác và chỉ định cách mỗi giá trị nên được định dạng.
IndexOf	Vị trí xuất hiện đầu tiên của 1 chuỗi con hoặc kí tự trong chuỗi
IndexOfAny	Vị trí xuất hiện đầu tiên của bất kì 1 hoặc 1 tập kí tự trong chuỗi
LastIndexOf	Giống indexof , nhưng tìm lần xuất hiện cuối cùng
LastIndexOfAny	Giống indexofAny , nhưng tìm lần xuất hiện cuối cùng

PadLeft	Canh phải chuỗi điền chuỗi bằng cách thêm 1 kí tự được chỉ định lặp lại vào đầu chuỗi
PadRigth	Canh trái chuỗi điền chuỗi bằng cách thêm 1 kí tự được chỉ định lặp lại vào cuối chuỗi
Replace	Thay thế kí tự hay chuỗi con trong chuỗi với 1 kí tự hoặc chuỗi con khác
Split	Chia chuỗi thành 2 mảng chuỗi con, ngắt bởi sự xuất hiện của một kí tự nào đó
Substring	Trả về chuỗi con bắt đầu ở một vị trí chỉ định trong chuỗi.
ToLower	Chuyển chuỗi thành chữ thường
ToUpper	Chuyển chuỗi thành chữ in
Trim	Bỏ khoảng trắng ở đầu và cuối chuỗi

Định dạng xuất chuỗi:

Specifier	Type	Format	Output (double 1.2345)	Output (int -12345)
c	currency	{0:c}	£1.23	-£12,345.00
d	decimal (whole number)	{0:d}	System.FormatException	-12345
e	exponent / scientific	{0:e}	1.234500e+000	-1.234500e+004
f	fixed point	{0:f}	1.23	-12345.00
g	general	{0:g}	1.2345	-12345
n	number	{0:n}	1.23	-12,345.00
r	round trippable	{0:r}	1.23	System.FormatException
x	hexadecimal	{0:x4}	System.FormatException	ffffcfc7

Ví dụ 1.1 : Xử lý trên chuỗi

```
1:  using System;
2:  using System.Collections.Generic;
3:  using System.Text;
4:
5:  namespace Lab1
6:  {
7:      class Program
```

```
 8:      {
 9:          static void Main(string[] args)
10:          {
11:
12:              string str1, str2, str;
13:              str1 = "Xin chao";
14:              str2 = " Sinh vien Khoa CNTT";
15:
16:              //1.1 Ví dụ Ghép chuỗi
17:              Console.WriteLine("Noi chuoi:");
18:              str = str1 + str2;
19:              Console.WriteLine("\t" + str);
20:
21:              //1.2 Ví dụ Chiều dài chuỗi
22:              int len = str.Length;
23:              Console.WriteLine("Chieu dai:" + len);
24:              //1.3 Ví dụ Lấy chuỗi con
25:              str1 = str.Substring(8);
26:              Console.WriteLine("\tstr.Substring(8):" + str1);
27:              str1 = str.Substring(8, 10);
28:              Console.WriteLine("\tstr.Substring(8, 10):" + str1);
29:              //1.4 Ví dụ Thay thế chuỗi
30:              Console.WriteLine("Thay the chuoi:Replace");
31:              str1 = str.Replace('T', 'K');
32:              Console.WriteLine("\t" + str1);
33:              str1 = str.Replace("Khoa CNTT", "Lop CTK31");
34:              Console.WriteLine("\t" + str1);
35:              //1.5 Ví dụ So sánh chuỗi
36:              Console.WriteLine("So sanh 2 chuoi:");
37:              str1 = "ppp";
38:              str2 = "ccc";
39:              int res = String.Compare(str1, str2);
40:              Console.WriteLine("\tFirst result:" + res.ToString());
41:              str2 = "ttt";
42:              res = String.Compare(str1, str2);
43:              Console.WriteLine("\tSecond result:" + res.ToString());
44:              str1 = "ttt";
45:              res = String.Compare(str1, str2);
46:              Console.WriteLine("\tThird result:" + res.ToString());
47:              //1.6 Ví dụ Định dạng xuất chuỗi
48:              Console.WriteLine("Dinh dang xuat chuoi:");
49:              double d = 1.2456d;
50:              str1 = String.Format("{0:c}", d);
51:              Console.WriteLine("\t" + str1);
52:              str1 = String.Format("{0:f}", d);
53:              Console.WriteLine("\t" + str1);
54:          }
55:      }
56:  }
```

2. Biểu thức chính quy

2.1 Một số ký tự trong biểu thức chính quy

Ký tự	Ý nghĩa
^	Bắt đầu của chuỗi nhập
$	Kết thúc của chuỗi nhập
.	Bất kì kí tự nào ngoại trừ kí tự xuống dòng(\n)
*	Kí tự trước có thể được lặp lại 0 hoặc nhiều lần
+	Kí tự trước có thể được lặp lại 1 hoặc nhiều lần
?	Kí tự trước có thể được lặp lại 0 hoặc 1 lần
\s	Bất kì kí tự khoảng trắng
\w	Ký tự word (gồm chữ cái và chữ số, dấu gạch dưới _) tương đương [a-zA-Z_0-9]
\W	Ký tự không phải ký tự word tương đương [^a-zA-Z_0-9]
\A	Bắt đầu 1 chuỗi
\z	Kết thúc 1 chuỗi
\S	Bất kì kí tự nào không phải là khoảng trắng
\b	Từ biên
\B	Bất kì vị trí nào không phải là từ biên
\d	Ký tự số 0-9
|	Ký tự ngăn cách so trùng tương đương với phép or (lưu ý cái này nếu muốn kết hợp nhiều điều kiện)
[abc]	Khớp với 1 ký tự nằm trong nhóm là a hay b hay c.
[a-z]	So trùng với 1 ký tự nằm trong phạm vi a-z, dùng dấu - làm dấu ngăn cách.
[^abc]	Sẽ không so trùng với 1 ký tự nằm trong nhóm, ví dụ không so trùng với a hay b hay c.

()	Xác định 1 group (biểu thức con) xem như nó là một yếu tố đơn lẻ trong pattern . Ví dụ: ((a(b))c) sẽ khớp với b, ab, abc.
{n}	n là con số, Khớp đúng với n ký tự đúng trước nó . Ví dụ A{2}: khớp đúng với 2 chữ A.
{n, }	Khớp đúng với n ký tự trở lên đứng trước nó , A{2,} khớp với AA, AAA ...
{m,n}	Khớp đùng với từ m->n ký tự đứng trước nó, A{2,4} khớp với AA,AAA,AAAA.

2.2 Sử dụng biểu thức chính quy

```
using System.Text.RegularExpressions
```

Lớp Regex:

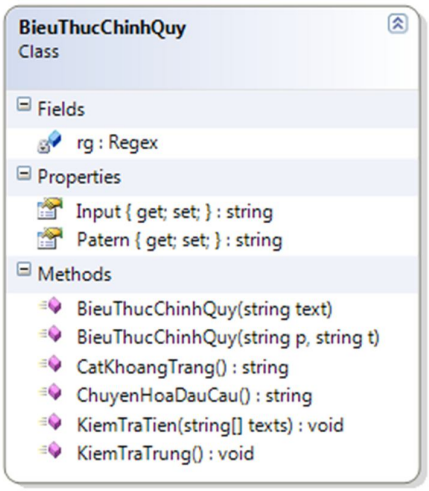
Thuộc tính	Mô tả
Options	Trả về những mục chọn được dùng trong phương thức tạo lập `Regex`.
RightToLeft	true, false: cho biết liệu xem regular expression dò tìm từ phải qua trái hay không?
Phương thức	**Mô tả**
GetGroupNames	Trả về mảng gồm toàn tên nhóm thu được đối với RE.
GetGroupNumbers	Trả về mảng gồm toàn số nhóm thu được tương ứng với tên nhóm trên 1 mảng.
GroupNameFromNumber	Lấy tên nhóm tương ứng với số nhóm được khai báo.
IsMatch	Trả về trị bool cho biết liệu xem RE có tìm thấy một so khớp hay không trên pattern.
Match	Dò tìm trên pattern xem có xuất hiện một RE hay không rồi trả về kết quả một đối tượng Match duy nhất.
Matches	Dò tìm trên pattern xem tất cả các xuất hiện của một RE có hay không rồi trả về tất cả những so khớp thành công.
Replace	Cho thay thế những xuất hiện của một pattern được định nghĩa bởi một RE bởi một chuỗi ký tự thay thế được chỉ định.
Split	Split một pattern thành một mảng gồm những chuỗi con ở những vị trí được chỉ định bởi một so khớp trên RE

Lớp Match:

Thuộc tính	Mô tả
Value	Trả về giá trị chuỗi con so khớp được
Index	Trả về vị trí đầu tiên của chuỗi so khớp được trong chuỗi gốc.
Success	Trả về true hoặc false cho biết so khớp thành công hay không?
Groups	Trả về nhóm so khớp của RE. Lớp Group: Thuộc tính: Index, value
Phương thức	**Mô tả**
NextMatch	Trả về Match so khớp được tiếp theo của đối tượng

Ví dụ 1.2: Sử dụng Regex

Xây dựng lớp sau:



Lớp BieuThucChinhQuy.cs

```
 1:  using System;
 2:  using System.Collections.Generic;
 3:  using System.Linq;
 4:  using System.Text;
 5:  using System.Text.RegularExpressions;
 6:  using System.Diagnostics;
 7:  namespace Lab1
 8:  {
 9:      class BieuThucChinhQuy
10:      {
11:          Regex rg;
12:          public string Patern { get; set; }
13:          public string Input { get; set; }
14:          public BieuThucChinhQuy()
15:          {
16:
17:          }
18:          public BieuThucChinhQuy(string p, string t)
19:          {
```

```
20:             Patern = p;
21:             Input = t;
22:             rg = new Regex(Patern, RegexOptions.IgnoreCase);
23:         }
24:         public string CatKhoangTrang()
25:         {
26:             return Regex.Replace(Input, @"(\s{2,})", delegate(Match m)
27:             {
28:                 if (m.Index == 0)
29:                     return "";
30:                 return " ";
31:             });
32:         }
33:         public void KiemTraTien(string[] texts)
34:         {
35:             this.Patern = @"^-?\d+(\.\d{2})?\s((USD)|(\$))";
36:             rg = new Regex(Patern);
37:             foreach (string text in texts)
38:             {
39:                 if (rg.IsMatch(text))
40:                     Console.WriteLine("{0} is a currency value.", text);
41:                 else
42:                   Console.WriteLine("{0} is not a currency value.", text);
43:             }
44:         }
45:         public void KiemTraTrung()
46:         {
47:             this.Patern = @"\b(?<tu>\w+)\s+(\k<tu>)\b";
48:             rg = new Regex(this.Patern, RegexOptions.IgnoreCase);
49:             MatchCollection listmatch = rg.Matches(Input);
50:             foreach (Match m in listmatch)
51:             {
52:                 GroupCollection listgroup = m.Groups;
53:               Console.WriteLine("Gia tri chuoi \"{0}\" lap lai:{1}, {2}",
54:                     listgroup["tu"].Value,
55:                     listgroup[0].Index,
56:                     listgroup[1].Index);
57:             }
58:         }
59:         public string ChuyenHoaDauCau()
60:         {
61:             this.Patern = @"(^[a-z])|(\.\s[a-z])";
62:             rg = new Regex(Patern, RegexOptions.Compiled);
63:             return rg.Replace(Input, delegate(Match m)
64:             {
65:                 string s = m.Value;
66:                 if (m.Index == 0)
67:                     return char.ToUpper(s[0]) + m.Value.Substring(1);
68:                 return s.Substring(0, 2) + char.ToUpper(s[2]);
69:             }
```

```
            );
        }
    }
}
```

Lớp Program.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Lab1
{
    class Program
    {
        static void Main(string[] args)
        {
            BieuThucChinhQuy bt = new BieuThucChinhQuy();
            bt.Input = "   xin xin    chao    cac ban ban. chao lop CTK33";
            Console.WriteLine("\nChuoi ban dau:\n" + bt.Input);
            bt.Input = bt.CatKhoangTrang();
            Console.WriteLine("\nSau khi cat khoang trang:\n" + bt.Input);
            bt.KiemTraTrung();
            Console.WriteLine("\nChuoi ket qua:\n" + bt.ChuyenHoaDauCau());
            //Text kiem tra dang tien te USD hoac $
            string[] texts = { "90.00 USD", "-50.5 USD", "700.0$", "500.08 $" };
            Console.WriteLine("\nKiem tra tien te\n");
            bt.KiemTraTien(texts);
        }
    }
}
```

3. Nhóm lớp đối tượng

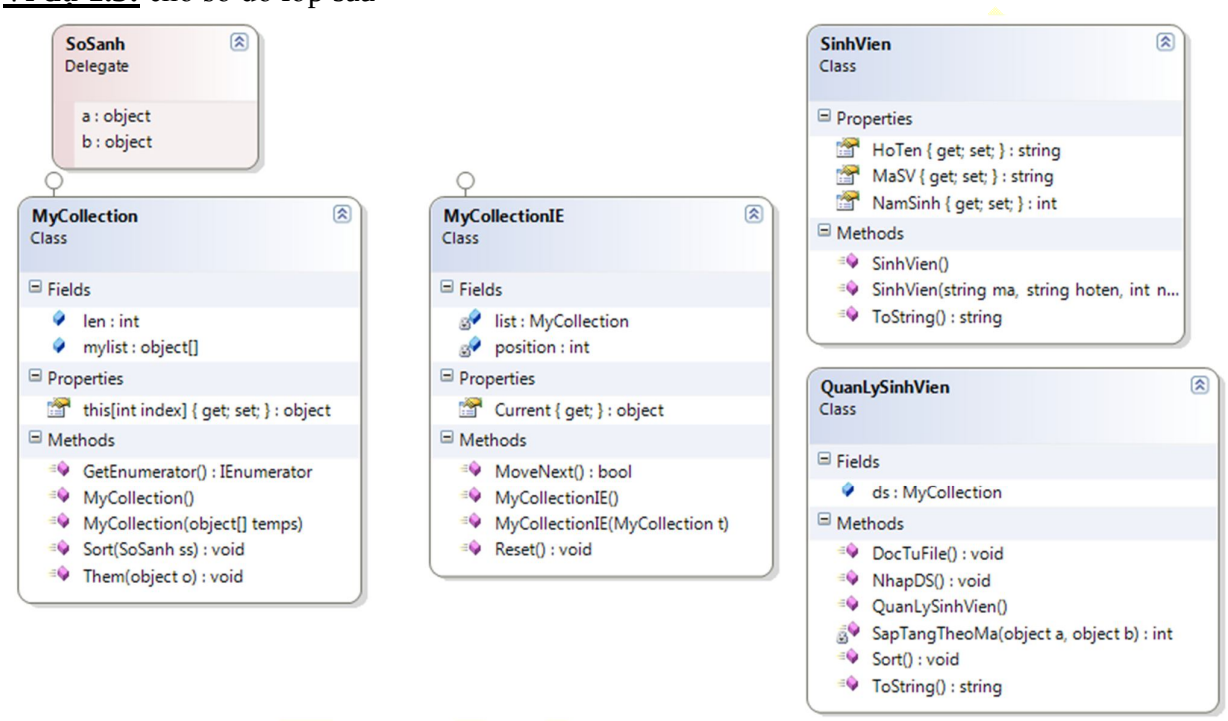
3.1 Lớp ArrayList

Phương thức	Mục đích
CopyTo	Copy cả ArrayList hay 1 phần tử sang mảng 1 chiều
IndexOf	Trả về zero-based index của sự xuất hiện đầu tiên của 1 trị trên ArrayList hoặc trên 1 phần mảng
Insert	Chèn thêm 1 phần tử vào ArrayList tại chỉ mục được chỉ định
LastIndexOf	Trả về zero-based index của sự xuất hiện cuối cùng của 1 trị trên ArrayList hoặc trên 1 phần mảng
Remove	Loại bỏ sự xuất hiện đầu tiên của một specific object ra khỏi ArrayList
RemoveAt	Loại bỏ những phần tử tại chỉ mục được chỉ định của ArrayList
RemoveRange	RemoveRange: gỡ bỏ 1 khoảng phần tử từ ArrayList
Reverse	Đảo ngược thứ tự của các phần tử trên ArrayList.

SetRange	Sao chép các phần tử của một collection chồng lên 1 phần phần tử trên ArrayList
ToArray	Sao chép các phần tử của ArrayList về 1 bản dãy
TrimToSize	Thiết đặt khả năng về số phần tử hiện hành trên ArrayList

3.2 Tạo Collection

Ví dụ 1.3: cho sơ đồ lớp sau



Lớp MyCollection.cs

```
1:  using System;
 2:  using System.Collections.Generic;
 3:  using System.Linq;
 4:  using System.Text;
 5:  using System.Collections;
 6:
 7:  namespace Lab1
 8:  {
 9:      public delegate int SoSanh(object a,object b);
10:      public class MyCollection:IEnumerable
11:      {
12:          public object[] mylist;
13:          public int len=0;
14:          public MyCollection()
15:          {
16:              len = 0;
17:              mylist = new object[100];
18:          }
19:          public MyCollection(object []temps)
```

```
20:          {
21:              mylist = temps;
22:              len = temps.Length;
23:          }
24:          public object this[int index]
25:          {
26:              get { return mylist[index]; }
27:              set { mylist[index] = value; }
28:          }
29:          public void Them(object ob)
30:          {
31:              this.mylist[len++] = ob;
32:          }
33:          public void Sort(SoSanh ss)
34:          {
35:              object temp;
36:              for(int i=0; i<len; i++)
37:                  for (int j = i + 1; j < len; j++)
38:                  if(ss(this[i], this[j])==1)
39:                  {
40:                      temp = this[i];
41:                      this[i] = this[j];
42:                      this[j] = temp;
43:                  }
44:          }
45:          public object TimMax(SoSanh ss)
46:          {
47:              object temp = mylist[0];
48:              foreach (object ob in this)
49:              {
50:                  if (ss(temp, ob) < 0)
51:                      temp = ob;
52:              }
53:              return temp;
54:          }
55:          #region IEnumerable Members
56:          public IEnumerator GetEnumerator()
57:          {
58:              return new MyCollectionIE(this);
59:          }
60:          #endregion
61:      }
```

Lớp MyCollectionIE.cs

```
62:      class MyCollectionIE : IEnumerator
63:      {
64:          MyCollection list;
65:          int position = -1;
66:          public MyCollectionIE()
67:          {
68:          }
```

```
69:          public MyCollectionIE(MyCollection t)
70:          {
71:              this.list = t;
72:          }
73:          #region IEnumerator Members
74:
75:          public object Current
76:          {
77:              get { return list.mylist[position]; }
78:          }
79:          public bool MoveNext()
80:          {
81:              if (position < list.len - 1)
82:              {
83:                  position++;
84:                  return true;
85:              }
86:              else
87:                  return false;
88:          }
89:          public void Reset()
90:          {
91:              position = -1;
92:          }
93:
94:          #endregion
95:      }
96:  }
```

Lớp SinhVien.cs

```
 1:  using System;
 2:  using System.Collections.Generic;
 3:  using System.Linq;
 4:  using System.Text;
 5:  using System.IO;
 6:
 7:  namespace Lab1
 8:  {
 9:      public class SinhVien
10:      {
11:          public string MaSV { get; set; }
12:          public string HoTen { get; set; }
13:          public int NamSinh { get; set; }
14:          public SinhVien()
15:          {
16:
17:          }
18:          public SinhVien(string ma, string hoten, int ns)
19:          {
20:              this.MaSV = ma;
21:              this.HoTen = hoten;
```

```
22:             this.NamSinh = ns;
23:         }
24:         public override string ToString()
25:         {
26:             string s = "{0}|{1}|{2}";
27:             return string.Format(s, MaSV, HoTen, NamSinh);
28:         }
29:     }
30: }
```

Lớp QuanLySinhVien.cs

Dữ liệu file*: DanhSachSinhVien.txt*

*002*Nguyen Van A*1980*
*010*Nguyen Thi Hong*1984*
*011*Nguyen Van H*1989*
*003*Le Hoang Hung*1980*
*005*Le Van C*1987*
*008*Hoang Van C*1982*

```
 1: public class QuanLySinhVien
 2: {
 3:     public MyCollection ds;
 4:
 5:     public QuanLySinhVien()
 6:     {
 7:         ds = new MyCollection();
 8:     }
 9:     public void NhapDS()
10:     {
11:         ds.Them(new SinhVien("002", "Nguyen Van A", 1980));
12:         ds.Them(new SinhVien("001", "Nguyen Van B", 1982));
13:         ds.Them(new SinhVien("004", "Nguyen Thi D", 1989));
14:         ds.Them(new SinhVien("003", "Nguyen Thi C", 1990));
15:     }
16:     public void DocTuFile()
17:     {
18:        StreamReader read = new StreamReader(File.Open("DanhSachSinhVien.txt"
19:                                                 , FileMode.Open));
20:         string s="";
21:         string[] slist;
22:         SinhVien temp;
23:         while ((s = read.ReadLine())!=null)
24:         {
25:             slist = s.Split('*');
26:             temp = new SinhVien(slist[0], slist[1], int.Parse(slist[2]));
27:             ds.Them(temp);
28:         }
29:         read.Close();
30:     }
```

```
31:       public void Sort()
32:       {
33:           this.ds.Sort(SapTangTheoMa);
34:       }
35:       private int SapTangTheoMa(object a, object b)
36:       {
37:           SinhVien sv1 = a as SinhVien;
38:           SinhVien sv2 = b as SinhVien;
39:           return sv1.MaSV.CompareTo(sv2.MaSV);
40:       }
41:       public override string ToString()
42:       {
43:           string s = "";
44:           foreach(SinhVien sv in ds)
45:           {
46:               s += "\n" + sv;
47:           }
48:           return s;
49:       }
50:       public SinhVien TimMax()
51:       {
52:           return this.ds.TimMax(delegate(object a, object b)
53:           {
54:               SinhVien sv1 = a as SinhVien;
55:               SinhVien sv2 = b as SinhVien;
56:               return sv1.NamSinh.CompareTo(sv2.NamSinh);
57:           }) as SinhVien;
58:
59:       }
60:   }
```

Lớp Program.cs

```
 1:  using System;
 2:  using System.Collections.Generic;
 3:  using System.Text;
 4:
 5:  namespace Lab1
 6:  {
 7:      class Program
 8:      {
 9:          static void Main(string[] args)
10:          {
11:              QuanLySinhVien qlsv = new QuanLySinhVien();
12:              qlsv.DocTuFile();
13:              Console.WriteLine("Danh sach sinh vien:\n" + qlsv);
14:              qlsv.Sort();
15:              Console.WriteLine("Danh sach tang theo ma:\n" + qlsv);
16:              Console.WriteLine("Sinh vien nam sinh Max:" + qlsv.TimMax());
17:          }
18:      }
19:  }
```

III. Bài tập:

1. Thao tác trên chuỗi

   Viết chương trình trình bày cách sử dụng các phương thức trên chuỗi: Clone, Join, PadLeft, PadRight, Split, ….

2. Bổ sung các phương thức sau vào lớp BieuThucChinhQuy (***ví dụ 1.2***) để:
   - 2.1. Kiểm tra số điện thoại di động
   - 2.2. Kiểm tra địa chỉ website
   - 2.3. Kiểm tra địa chỉ IP
   - 2.4. Kiểm tra tên biến (ngôn ngữ: C++).
   - 2.5. Kiểm tra địa chỉ email của yahoo hoặc gmail.
   - 2.6. Loại bỏ chữ lặp lại liên tiếp.
   - 2.7. Chuyển chuỗi hoa thành thường.(Ví dụ: xin Chao K33 → xin chao k33)
   - 2.8. Chuyển chuỗi thành ký tự đầu mỗi chữ là hoa (ví dụ: Nguyễn văn an → Nguyễn Văn An).

3. Bổ sung các phương thức vào ***ví dụ 1.3*** thực hiện các thao tác sau:
   - 3.1. Sử dụng biểu thức chính quy tìm danh sách nhân viên có họ “Nguyen”.
   - 3.2. Sử dụng biểu thức chính quy thay thế nhân viên sinh năm 1980 thành 1990
   - 3.3. Tìm nhân viên sinh năm min.
   - 3.4. Sắp danh sách nhân viên giảm theo năm sinh.
   - 3.5. Xóa nhân viên theo mã nhân viên.
   - 3.6. Sử dụng biểu thức chính quy tìm danh sách nhân viên có tên bắt đầu bằng “H” kết thúc bằng “ng”.
   - 3.7. Tổ chức chương trình thành menu.