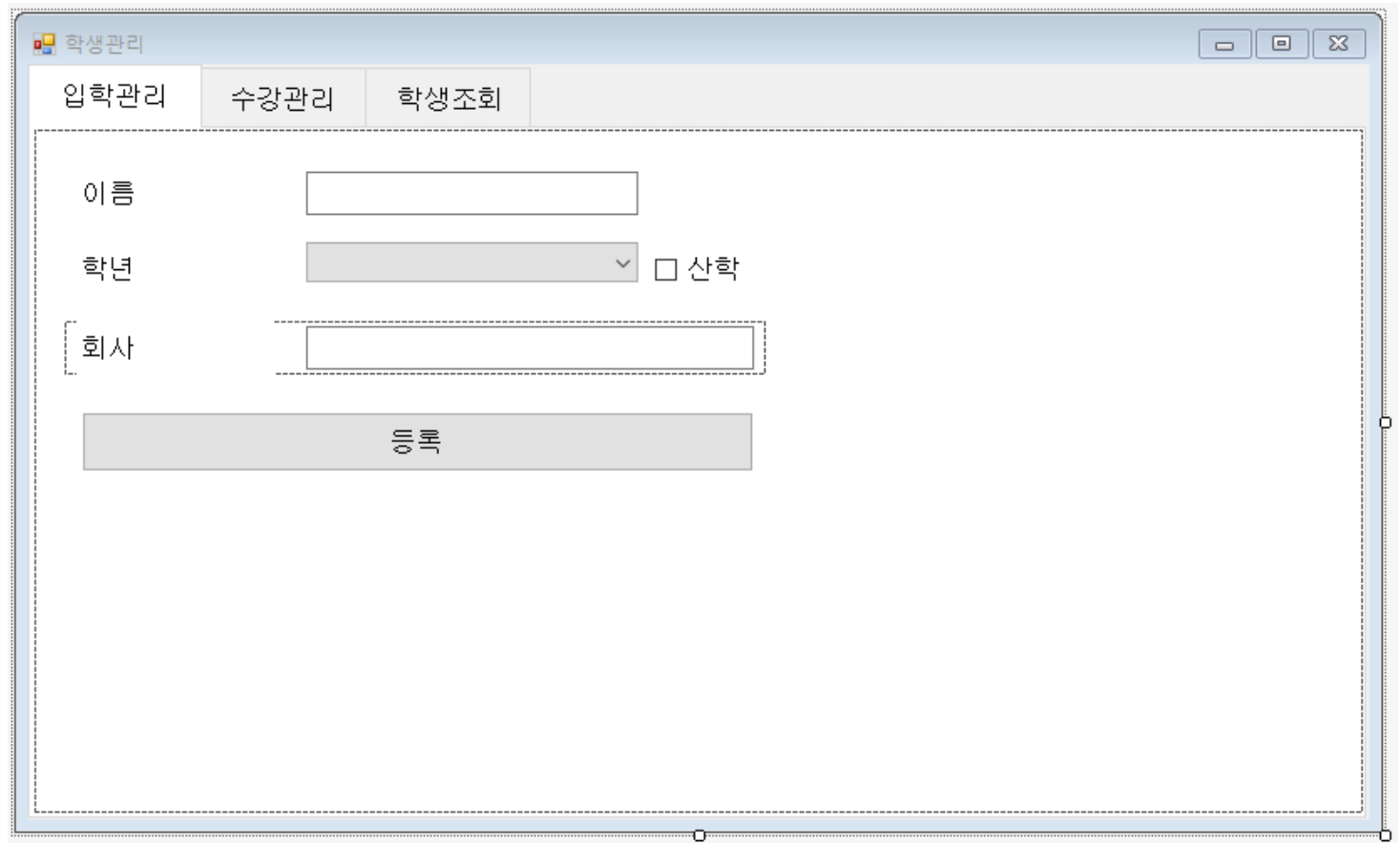
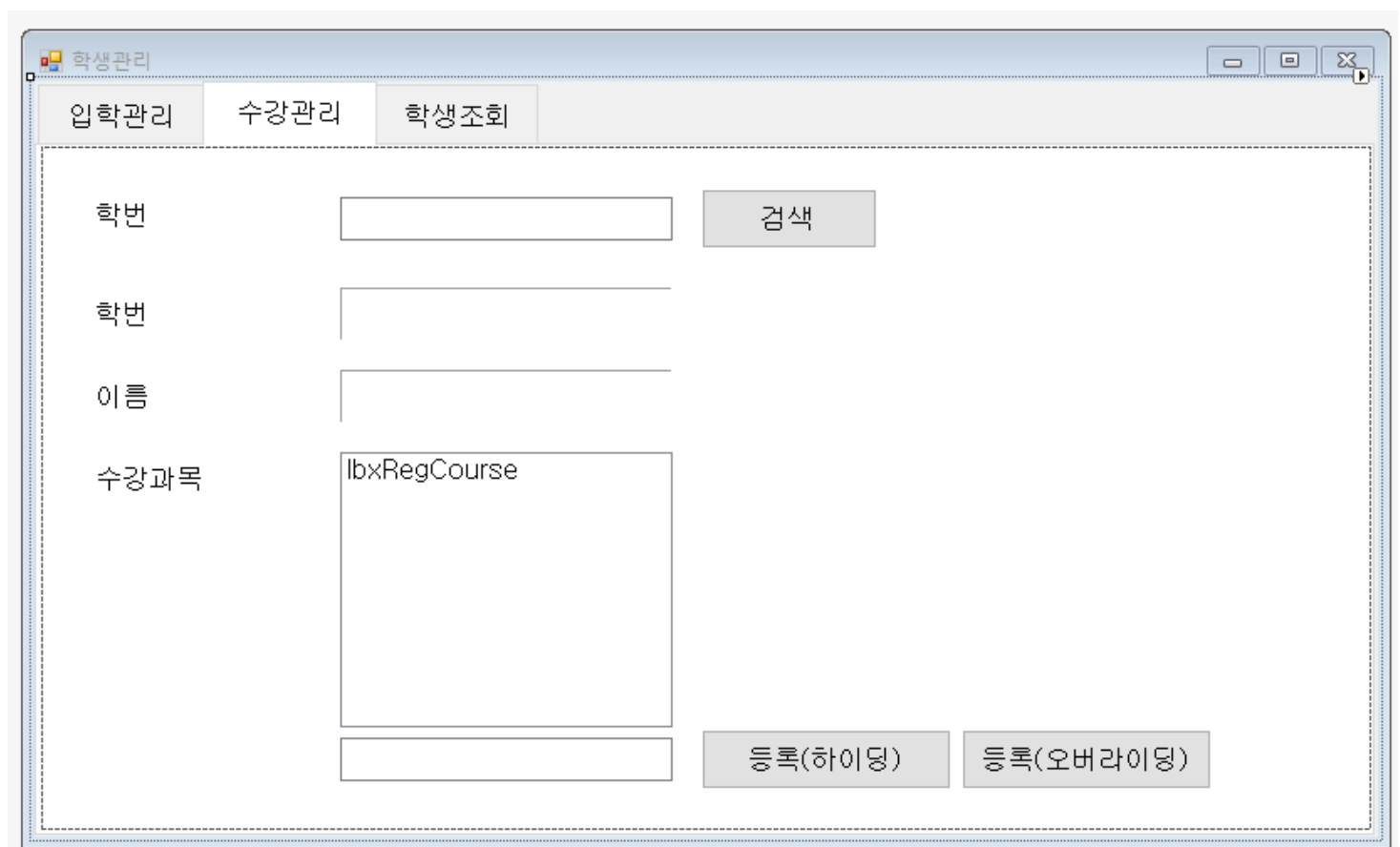


[화면]



A screenshot of a web application window titled "학생관리" (Student Management). The window has three tabs: "입학관리" (Admission Management), "수강관리" (Course Management), and "학생조회" (Student Search). The "입학관리" tab is selected. The form contains the following fields and controls:

- 이름** (Name): A text input field.
- 학년** (Grade): A dropdown menu with a downward arrow.
- 산학** (Industry): A checkbox.
- 회사** (Company): A text input field.
- 등록** (Register): A button.



A screenshot of the same web application window, but with the "학생조회" (Student Search) tab selected. The form contains the following fields and controls:

- 학번** (Student ID): A text input field.
- 검색** (Search): A button.
- 학번** (Student ID): A text input field.
- 이름** (Name): A text input field.
- 수강과목** (Course): A text input field containing the text "lbXRegCourse".
- 등록(하이딩)** (Register (Hiding)): A button.
- 등록(오버라이딩)** (Register (Overriding)): A button.

## Student.cs

```
1  using ...
6
7  namespace School
8  {
9      class Student
10     {
11         private string _id; //학번
12         public string Id
13         {
14             get
15             {
16                 return _id;
17             }
18         }
19
20         private string _name; //이름
21         public string Name
22         {
23             get
24             {
25                 return _name;
26             }
27         }
28
29         private int _grade; //학년
30         public int Grade
31         {
32             get
33             {
34                 return _grade;
35             }
36
37             set
38             {
39                 _grade = value;
40             }
41         }
42
43         //private List<string> _subject; //수강과목
44         protected List<string> _subject; //수강과목
45         public List<string> Subject
46         {
47             get
48             {
49                 return _subject;
50             }
51         }
52
53         public Student(string name, string id)
54         {
55             _name = name;
56             _id = id;
57             _grade = 1;
58             _subject = new List<string>();
59         }
60
61         public Student(string name, string id, int grade)
62             : this(name, id)
63         {
64             Grade = grade;
65         }
66     }
67 }
```

```

66 public bool RegCourse(string subject)
67 {
68     bool result = false;
69
70     if (_subject == null)
71     {
72         _subject = new List<string>();
73     }
74
75     #region COMMENT
76     //LINQ
77     //if (null == _subject.Where(m => m == subject).FirstOrDefault())
78     //{
79     //    _subject.Add(subject);
80     //    result = true;
81     //}
82     #endregion
83
84     string searchSubject = null;
85     foreach(var sub in _subject)
86     {
87         if(sub == subject)
88         {
89             searchSubject = sub;
90             break;
91         }
92     }
93
94     if(searchSubject == null)
95     {
96         _subject.Add(subject);
97         result = true;
98     }
99
100     return result;
101 }
102
103
104 public virtual bool RegCourseEx(string subject)
105 {
106     bool result = false;
107
108     if (_subject == null)
109     {
110         _subject = new List<string>();
111     }
112
113     COMMENT
114
115     string searchSubject = null;
116     foreach (var sub in _subject)
117     {
118         if (sub == subject)
119         {
120             searchSubject = sub;
121             break;
122         }
123     }
124
125     if (searchSubject == null)
126     {
127         _subject.Add(subject);
128         result = true;
129     }
130
131     return result;
132 }
133
134 }
135
136 }
137
138 }
139
140 }
141

```

```
1  using ...
6
7  namespace School
8  {
9      class StudentIndustrialEdu : Student
10     {
11         const int MAX_REG_SUBJECT = 4;
12
13         private string _company;
14         public string Company
15         {
16             get
17             {
18                 return _company;
19             }
20             set
21             {
22                 _company = value;
23             }
24         }
25
26         public StudentIndustrialEdu(string name, string id, int grade, string company) : base(name, id, grade)
27         {
28             _company = company;
29         }
30
31         public StudentIndustrialEdu(string name, string id, string company) : base(name, id)
32         {
33             _company = company;
34         }
35
36         public new bool RegCourse(string subject) //hiding
37         {
38             bool result = false;
39
40             if (_subject == null)
41             {
42                 _subject = new List<string>();
43             }
44
45             if (_subject.Count < MAX_REG_SUBJECT)
46             {
47                 string searchSubject = null;
48                 foreach (var sub in _subject)
49                 {
50                     if (sub == subject)
51                     {
52                         searchSubject = sub;
53                         break;
54                     }
55                 }
56
57                 if (searchSubject == null)
58                 {
59                     _subject.Add(subject);
60                     result = true;
61                 }
62             }
63             return result;
64         }
65     }
66 }
```

```
67 public override bool RegCourseEx(string subject) //overriding
68 {
69     bool result = false;
70
71     if (_subject == null)
72     {
73         _subject = new List<string>();
74     }
75
76     if (_subject.Count < MAX_REG_SUBJECT)
77     {
78         string searchSubject = null;
79         foreach (var sub in _subject)
80         {
81             if (sub == subject)
82             {
83                 searchSubject = sub;
84                 break;
85             }
86         }
87
88         if (searchSubject == null)
89         {
90             _subject.Add(subject);
91             result = true;
92         }
93     }
94     return result;
95 }
96 }
97 }
```

## Form1.cs

```
1  using ...
10
11  namespace School
12  {
13      public partial class Form1 : Form
14      {
15          const int MAX_GRADE = 4;
16          List<Student> _students;
17          Student _studentRegCourse = null;
18
19          public Form1()
20          {
21              InitializeComponent();
22
23              _students = new List<Student>();
24
25              for (int i = 1; i <= MAX_GRADE; i++)
26              {
27                  cbxEntranceGrade.Items.Add(i.ToString());
28              }
29              cbxEntranceGrade.SelectedIndex = 0;
30          }
31
32          private void chkIndustrialEdu_CheckedChanged(object sender, EventArgs e)
33          {
34              pnlEntranceCompany.Visible = chkIndustrialEdu.Checked;
35          }
36
37          private void btnEntrance_Click(object sender, EventArgs e)
38          {
39              string temp_str = null;
40              int temp_int = 0;
41
42              int grade = 1;
43              int.TryParse(cbxEntranceGrade.SelectedItem.ToString(), out grade);
44
45              int year = DateTime.Now.Year - (grade - 1);
46              string id = string.Empty;
47              if (_students != null && _students.Count > 0)
48              {
49                  temp_str = year.ToString("0000");
50                  temp_str = _students.Where(m => m.Id.StartsWith(temp_str))
51                      .OrderByDescending(m => m.Id)
52                      .Select(m => m.Id)
53                      .FirstOrDefault();
54
55                  if (string.IsNullOrEmpty(temp_str))
56                  {
57                      id = $"{year:0000}0001";
58                  }
59                  else
60                  {
61                      int.TryParse(temp_str.Substring(4, 4), out temp_int);
62                      id = id + $"{year:0000}{temp_int + 1:0000}";
63                  }
64              }
65              else
66              {
67                  id = $"{year:0000}0001";
68              }
69          }
70      }
71  }
```

```

69
70
71 Student student = null;
72 if (chkIndustrialEdu.Checked)
73 {
74     student = new StudentIndustrialEdu(tbxEntranceName.Text, id, grade, tbxEntranceCompany.Text);
75 }
76 else
77 {
78     student = new Student(tbxEntranceName.Text, id, grade);
79 }
80 _students.Add(student);
81
82 lblEntranceResult.Text
83     = $"등록학생의 정보입니다.\r\n학번 : {student.Id}\r\n이름 : {student.Name}\r\n학년 : {student.Grade}\r\n";
84
85 //Type1
86 //if(student is StudentIndustrialEdu)
87 //{
88 //    temp_str = ((StudentIndustrialEdu)student).Company;
89 //    lblEntranceResult.Text += $"[산학과정] 소속회사:{temp_str}";
90 //}
91
92 //Type2
93 StudentIndustrialEdu studentIE = student as StudentIndustrialEdu;
94 if (studentIE != null)
95 {
96     lblEntranceResult.Text += $"[산학과정] 소속회사:{studentIE.Company}";
97 }
98
99 private void btnRegCourseSearch_Click(object sender, EventArgs e)
100 {
101     if (_students == null || _students.Count <= 0)
102     {
103         MessageBox.Show("검색할 학생이 없습니다.");
104         return;
105     }
106
107     _studentRegCourse = null;
108     foreach (var stu in _students)
109     {
110         if (stu.Id == tbxRegCourseSearchId.Text)
111         {
112             _studentRegCourse = stu;
113             break;
114         }
115     }
116
117     if (_studentRegCourse == null)
118     {
119         lblRegCourseId.Text = string.Empty;
120         lblRegCourseName.Text = string.Empty;
121         lbxRegCourse.Items.Clear();
122
123         MessageBox.Show("해당 학생을 찾을 수 없습니다.");
124     }
125     else
126     {
127         lblRegCourseId.Text = _studentRegCourse.Id;
128         lblRegCourseName.Text = _studentRegCourse.Name;
129         lbxRegCourse.Items.Clear();
130         foreach (var sub in _studentRegCourse.Subject)
131         {
132             lbxRegCourse.Items.Add(sub);
133         }
134     }
135 }
136

```

```

137 private void btnRegCourseHiding_Click(object sender, EventArgs e)
138 {
139     if (_studentRegCourse != null)
140     {
141         if (_studentRegCourse.RegCourse(tbxRegCourseName.Text))//hiding
142         {
143             lbxRegCourse.Items.Add(tbxRegCourseName.Text);
144         }
145         else
146         {
147             MessageBox.Show("등록실패");
148         }
149     }
150 }
151
152 private void btnRegCourseOverriding_Click(object sender, EventArgs e)
153 {
154     if (_studentRegCourse != null)
155     {
156         if (_studentRegCourse.RegCourseEx(tbxRegCourseName.Text))//overriding**
157         {
158             lbxRegCourse.Items.Add(tbxRegCourseName.Text);
159         }
160         else
161         {
162             MessageBox.Show("등록실패");
163         }
164     }
165 }
166 }
167

```