

# MAINFORMDESIGNER.cs

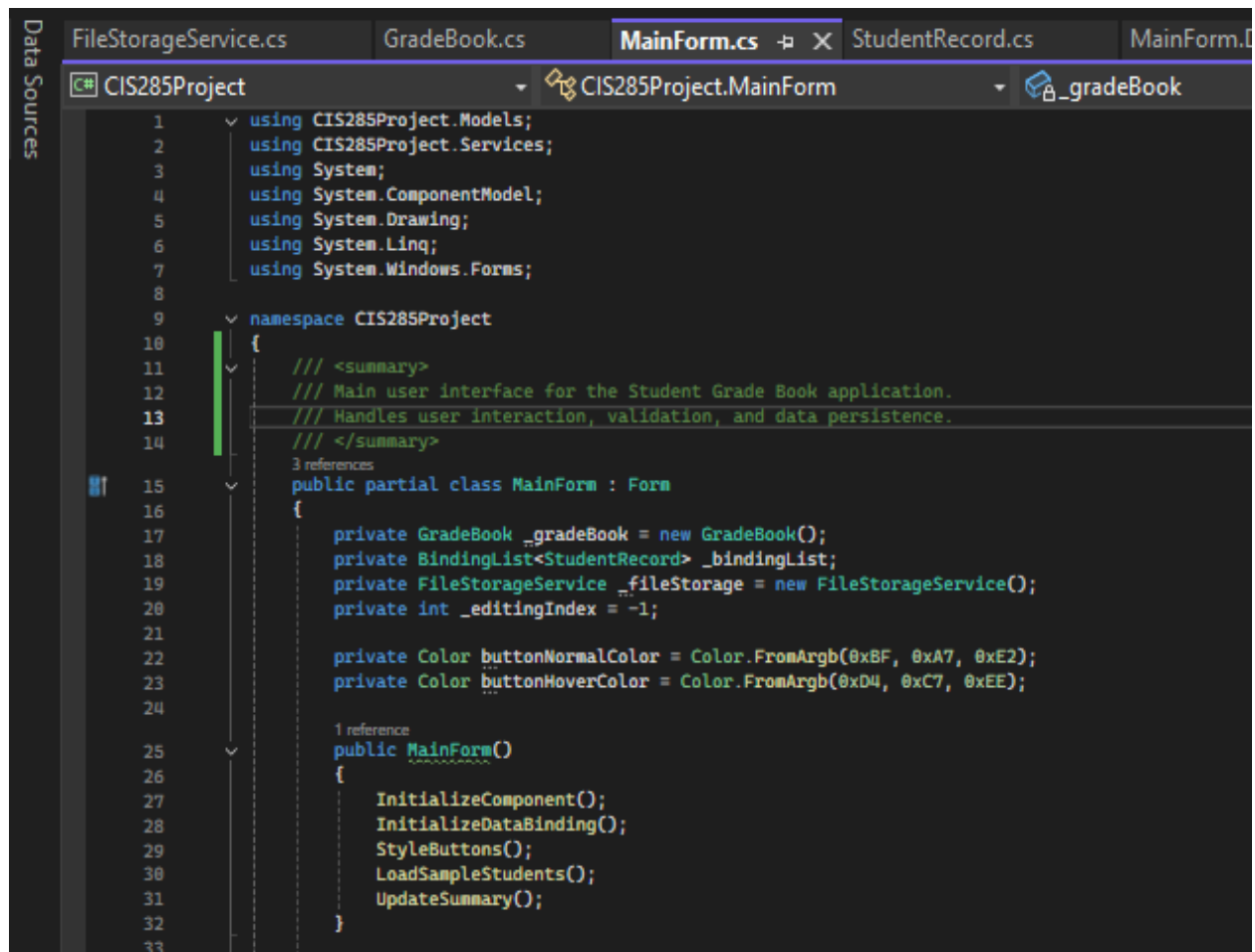
```

FileStorageService.cs  GradeBook.cs  StudentRecord.cs  MainForm.cs  MainForm.Designer.cs
CIS285Project
CIS285Project.MainForm

74
75 // DATAGRIDVIEW
76 this.dgvStudents.Location = new System.Drawing.Point(25, 25);
77 this.dgvStudents.Size = new System.Drawing.Size(800, 350);
78 this.dgvStudents.ReadOnly = true;
79 this.dgvStudents.RowHeadersVisible = false;
80 this.dgvStudents.SelectionMode = System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;
81 this.dgvStudents.AllowUserToAddRows = false;
82 this.dgvStudents.AllowUserToDeleteRows = false;
83 this.dgvStudents.MultiSelect = false;
84 this.dgvStudents.ScrollBars = System.Windows.Forms.ScrollBars.Both;
85 this.dgvStudents.EnableHeadersVisualStyles = false;
86 this.dgvStudents.ColumnHeadersDefaultCellStyle.BackColor =
87 System.Drawing.Color.FromArgb(0x8F, 0xA7, 0xE2);
88 this.dgvStudents.ColumnHeadersDefaultCellStyle.ForeColor =
89 System.Drawing.Color.White;
90 this.dgvStudents.BackgroundColor = System.Drawing.Color.White;
91
92 this.dgvStudents.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
93     this.colName, this.colG1, this.colG2, this.colG3, this.colG4,
94     this.colG5, this.colG6, this.colG7, this.colG8, this.colAvg
95 });
96
97 this.dgvStudents.SelectionChanged += new System.EventHandler(this.dgvStudents_SelectionChanged);
98
99 // Column bindings
100 this.colName.DataPropertyName = "Name";
101 this.colG1.DataPropertyName = "Grade1";
102 this.colG2.DataPropertyName = "Grade2";
103 this.colG3.DataPropertyName = "Grade3";
104 this.colG4.DataPropertyName = "Grade4";
105 this.colG5.DataPropertyName = "Grade5";
106 this.colG6.DataPropertyName = "Grade6";
107 this.colG7.DataPropertyName = "Grade7";
108 this.colG8.DataPropertyName = "Grade8";
109 this.colAvg.DataPropertyName = "Average";
110
111 // Column headers
112 this.colName.HeaderText = "Student Name";
113 this.colName.Width = 140;
114
115 this.colG1.HeaderText = "Grade 1"; this.colG1.Width = 60;
116 this.colG2.HeaderText = "Grade 2"; this.colG2.Width = 60;
117 this.colG3.HeaderText = "Grade 3"; this.colG3.Width = 60;
118 this.colG4.HeaderText = "Grade 4"; this.colG4.Width = 60;
119 this.colG5.HeaderText = "Grade 5"; this.colG5.Width = 60;
120 this.colG6.HeaderText = "Grade 6"; this.colG6.Width = 60;
121 this.colG7.HeaderText = "Grade 7"; this.colG7.Width = 60;
122 this.colG8.HeaderText = "Grade 8"; this.colG8.Width = 60;
123
124 this.colAvg.HeaderText = "Average";
125 this.colAvg.Width = 80;
126
127 // BUTTONS
128 this.btnAdd.Location = new System.Drawing.Point(850, 25);
129 this.btnAdd.Size = new System.Drawing.Size(220, 40);
130 this.btnAdd.Text = "Add Student";
131 this.btnAdd.Click += new System.EventHandler(this.btnAdd_Click);
132 this.btnAdd.MouseEnter += new System.EventHandler(this.Button_MouseEnter);
133 this.btnAdd.MouseLeave += new System.EventHandler(this.Button_MouseLeave);
134
135 this.btnEdit.Location = new System.Drawing.Point(850, 75);
136 this.btnEdit.Size = new System.Drawing.Size(220, 40);
137 this.btnEdit.Text = "Edit Student";
138 this.btnEdit.Click += new System.EventHandler(this.btnEdit_Click);
139 this.btnEdit.MouseEnter += new System.EventHandler(this.Button_MouseEnter);
140 this.btnEdit.MouseLeave += new System.EventHandler(this.Button_MouseLeave);
141
142 this.btnDelete.Location = new System.Drawing.Point(850, 125);
143 this.btnDelete.Size = new System.Drawing.Size(220, 40);
144 this.btnDelete.Text = "Delete Student";
145 this.btnDelete.Click += new System.EventHandler(this.btnDelete_Click);
146 this.btnDelete.MouseEnter += new System.EventHandler(this.Button_MouseEnter);
147
70 %  No issues found

```

## MAINFORM.cs (MULTIPLE SCREENSHOTS)



```
1  using CIS285Project.Models;
2  using CIS285Project.Services;
3  using System;
4  using System.ComponentModel;
5  using System.Drawing;
6  using System.Linq;
7  using System.Windows.Forms;
8
9  namespace CIS285Project
10 {
11     /// <summary>
12     /// Main user interface for the Student Grade Book application.
13     /// Handles user interaction, validation, and data persistence.
14     /// </summary>
15     3 references
16     public partial class MainForm : Form
17     {
18         private GradeBook _gradeBook = new GradeBook();
19         private BindingList<StudentRecord> _bindingList;
20         private FileStorageService _fileStorage = new FileStorageService();
21         private int _editingIndex = -1;
22
23         private Color buttonNormalColor = Color.FromArgb(0xBF, 0xA7, 0xE2);
24         private Color buttonHoverColor = Color.FromArgb(0xD4, 0xC7, 0xEE);
25
26         1 reference
27         public MainForm()
28         {
29             InitializeComponent();
30             InitializeDataBinding();
31             StyleButtons();
32             LoadSampleStudents();
33             UpdateSummary();
34         }
35     }
36 }
```

```
FileStorageService.cs  GradeBook.cs  MainForm.cs  StudentRecord.cs  MainForm.Designer.cs
CIS285Project  CIS285Project.MainForm  _gradeBook

65 }
66
67
68 // Live validation for grade textboxes
69 8 references
70 private void GradeTextBox_TextChanged(object sender, EventArgs e)
71 {
72     if (sender is TextBox tb)
73     {
74         if (string.IsNullOrWhiteSpace(tb.Text))
75         {
76             tb.BackColor = Color.White;
77             return;
78         }
79         if (double.TryParse(tb.Text, out double value) && value >= 0 && value <= 100)
80             tb.BackColor = Color.White;
81         else
82             tb.BackColor = Color.MistyRose;
83     }
84 }
85
86 // -----
87 // LOAD 19 SAMPLE STUDENTS (8 GRADES)
88 // -----
89 1 reference
90 private void LoadSampleStudents()
91 {
92     _gradeBook.Students.Clear();
93     _gradeBook.AddStudent(new StudentRecord
94     {
95         Name = "Jake Arrieta",
96         Grade1 = 85,
97         Grade2 = 92,
98         Grade3 = 88,
99         Grade4 = 91,
100        Grade5 = 89,
101        Grade6 = 94,
102        Grade7 = 90,
103        Grade8 = 93
104    });
105     _gradeBook.AddStudent(new StudentRecord
106     {
107         Name = "Anthony Rizzo",
108         Grade1 = 78,
109         Grade2 = 82,
110         Grade3 = 80,
111         Grade4 = 75,
112         Grade5 = 79,
113         Grade6 = 81,
114         Grade7 = 77,
115         Grade8 = 83
116     });
117     _gradeBook.AddStudent(new StudentRecord
118     {
119         Name = "Troy Polanalu",
120         Grade1 = 94,
121         Grade2 = 90,
122         Grade3 = 96,
123         Grade4 = 98,
124         Grade5 = 99,
125         Grade6 = 97,
126         Grade7 = 95,
127         Grade8 = 99
128     });
129     _gradeBook.AddStudent(new StudentRecord
130     {
131         Name = "Hines Ward",
132         Grade1 = 67,
133         Grade2 = 72,
```

```
340         Grade7 = 87,  
341         Grade8 = 100,  
342  
343     });  
344  
345     _bindingList = new BindingList<StudentRecord>(_gradeBook.Students);  
346     dgvStudents.DataSource = _bindingList;  
347  
348     lblStatus.Text = "Loaded 19 active students.";  
349 }  
350  
351   
352   
353   
354   
355   
356   
357   
358   
359   
360   
361   
362   
363   
364   
365   
366   
367   
368   
369   
370   
371   
372   
373   
374   
375   
376   
377   
378   
379   
380   
381   
382   
383   
384   
385   
386   
387   
388   
389   
390   
391   
392   
393   
394   
395   
396   
397   
398   
399   
400   
401   
402   
403   
404   
405   
406   
407   
408   
409   
410   
411   
412   
413   
414   
415   
416   
417   
418   
419   
420   
421   
422   
423   
424   
425   
426   
427   
428   
429   
430   
431   
432   
433   
434   
435   
436   
437   
438   
439   
440   
441   
442   
443   
444   
445   
446   
447   
448   
449   
450   
451   
452   
453   
454   
455   
456   
457   
458   
459   
460   
461   
462   
463   
464   
465   
466   
467   
468   
469   
470   
471   
472   
473   
474   
475   
476   
477   
478   
479   
480   
481   
482   
483   
484   
485   
486   
487   
488   
489   
490   
491   
492   
493   
494   
495   
496   
497   
498   
499   
500   
501   
502   
503   
504   
505   
506   
507   
508   
509   
510   
511   
512   
513   
514   
515   
516   
517   
518   
519   
520   
521   
522   
523   
524   
525   
526   
527   
528   
529   
530   
531   
532   
533   
534   
535   
536   
537   
538   
539   
540   
541   
542   
543   
544   
545   
546   
547   
548   
549   
550   
551   
552   
553   
554   
555   
556   
557   
558   
559   
560   
561   
562   
563   
564   
565   
566   
567   
568   
569   
570   
571   
572   
573   
574   
575   
576   
577   
578   
579   
580   
581   
582   
583   
584   
585   
586   
587   
588   
589   
590   
591   
592   
593   
594   
595   
596   
597   
598   
599   
600   
601   
602   
603   
604   
605   
606   
607   
608   
609   
610   
611   
612   
613   
614   
615   
616   
617   
618   
619   
620   
621   
622   
623   
624   
625   
626   
627   
628   
629   
630   
631   
632   
633   
634   
635   
636   
637   
638   
639   
640   
641   
642   
643   
644   
645   
646   
647   
648   
649   
650   
651   
652   
653   
654   
655   
656   
657   
658   
659   
660   
661   
662   
663   
664   
665   
666   
667   
668   
669   
670   
671   
672   
673   
674   
675   
676   
677   
678   
679   
680   
681   
682   
683   
684   
685   
686   
687   
688   
689   
690   
691   
692   
693   
694   
695   
696   
697   
698   
699   
700   
701   
702   
703   
704   
705   
706   
707   
708   
709   
710   
711   
712   
713   
714   
715   
716   
717   
718   
719   
720   
721   
722   
723   
724   
725   
726   
727   
728   
729   
730   
731   
732   
733   
734   
735   
736   
737   
738   
739   
740   
741   
742   
743   
744   
745   
746   
747   
748   
749   
750   
751   
752   
753   
754   
755   
756   
757   
758   
759   
760   
761   
762   
763   
764   
765   
766   
767   
768   
769   
770   
771   
772   
773   
774   
775   
776   
777   
778   
779   
780   
781   
782   
783   
784   
785   
786   
787   
788   
789   
790   
791   
792   
793   
794   
795   
796   
797   
798   
799   
800   
801   
802   
803   
804   
805   
806   
807   
808   
809   
810   
811   
812   
813   
814   
815   
816   
817   
818   
819   
820   
821   
822   
823   
824   
825   
826   
827   
828   
829   
830   
831   
832   
833   
834   
835   
836   
837   
838   
839   
840   
841   
842   
843   
844   
845   
846   
847   
848   
849   
850   
851   
852   
853   
854   
855   
856   
857   
858   
859   
860   
861   
862   
863   
864   
865   
866   
867   
868   
869   
870   
871   
872   
873   
874   
875   
876   
877   
878   
879   
880   
881   
882   
883   
884   
885   
886   
887   
888   
889   
890   
891   
892   
893   
894   
895   
896   
897   
898   
899   
900   
901   
902   
903   
904   
905   
906   
907   
908   
909   
910   
911   
912   
913   
914   
915   
916   
917   
918   
919   
920   
921   
922   
923   
924   
925   
926   
927   
928   
929   
930   
931   
932   
933   
934   
935   
936   
937   
938   
939   
940   
941   
942   
943   
944   
945   
946   
947   
948   
949   
950   
951   
952   
953   
954   
955   
956   
957   
958   
959   
960   
961   
962   
963   
964   
965   
966   
967   
968   
969   
970   
971   
972   
973   
974   
975   
976   
977   
978   
979   
980   
981   
982   
983   
984   
985   
986   
987   
988   
989   
990   
991   
992   
993   
994   
995   
996   
997   
998   
999   
1000
```

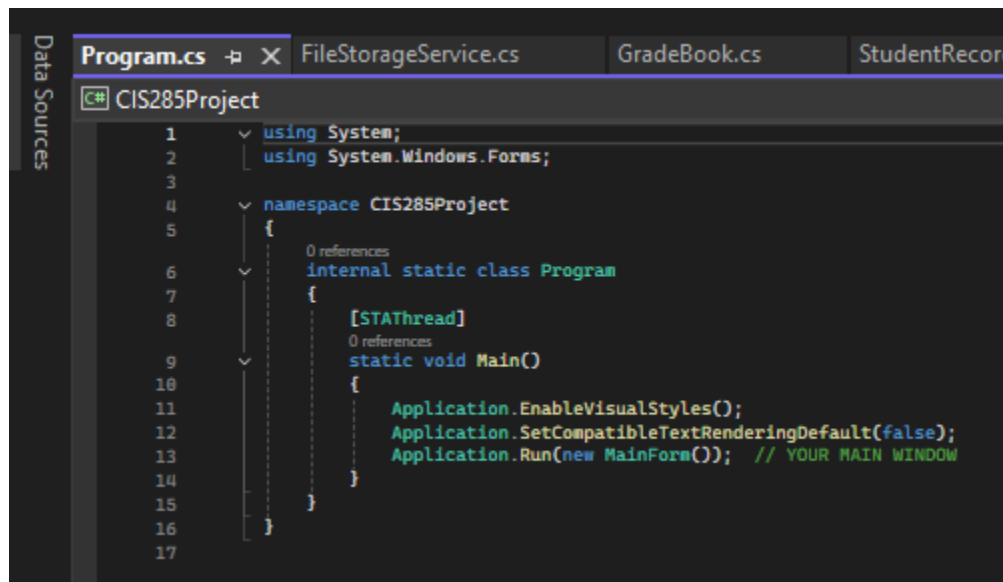
FileStorageService.csGradeBook.csStudentRecord.csMainForm.csMainForm.cs

CIS285ProjectCIS285P

529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582

```
    }  
  
    1 reference  
    private void btnAverage_Click(object sender, EventArgs e)  
    {  
        lblClassAverage.Text = _gradeBook.CalculateClassAverage().ToString("F2");  
        lblStatus.Text = "Class average updated.";  
        UpdateSummary();  
    }  
  
    1 reference  
    private void btnSaveFile_Click(object sender, EventArgs e)  
    {  
        SaveFileDialog dialog = new SaveFileDialog();  
        dialog.Filter = "XML Files|*.xml";  
  
        if (dialog.ShowDialog() == DialogResult.OK)  
        {  
            _fileStorage.Save(dialog.FileName, _gradeBook.Students);  
            lblStatus.Text = "File saved.";  
        }  
    }  
  
    1 reference  
    private void btnSaveNotes_Click(object sender, EventArgs e)  
    {  
        if (dgvStudents.CurrentRow == null)  
        {  
            MessageBox.Show("Select a student first.");  
            return;  
        }  
  
        int index = dgvStudents.CurrentRow.Index;  
        _gradeBook.Students[index].Notes = txtNotes.Text;  
  
        lblStatus.Text = "Notes saved.";  
    }  
  
    1 reference  
    private void btnLoadFile_Click(object sender, EventArgs e)  
    {  
        OpenFileDialog dialog = new OpenFileDialog();  
        dialog.Filter = "XML Files|*.xml";  
  
        if (dialog.ShowDialog() == DialogResult.OK)  
        {  
            var list = _fileStorage.Load(dialog.FileName);  
  
            _gradeBook.Students.Clear();  
            _gradeBook.Students.AddRange(list);  
  
            _bindingList = new BindingList<StudentRecord>(_gradeBook.Students);  
            dgvStudents.DataSource = _bindingList;  
  
            UpdateSummary();  
            lblStatus.Text = "File loaded.";  
        }  
    }  
}
```

## PROGRAM.cs



```
1  using System;
2  using System.Windows.Forms;
3
4  namespace CIS285Project
5  {
6      internal static class Program
7      {
8          [STAThread]
9          static void Main()
10         {
11             Application.EnableVisualStyles();
12             Application.SetCompatibleTextRenderingDefault(false);
13             Application.Run(new MainForm()); // YOUR MAIN WINDOW
14         }
15     }
16 }
17
```

## FILESTORAGESERVICE.cs

```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Debug Any CPU Start
Data Sources
FileStorageService.cs GradeBook.cs MainForm.cs StudentRecord.cs
CIS285Project CIS285Project.Services.FileStorageServi Save(string filePath, List<Stud

10
11 namespace CIS285Project.Services
12 {
13     /// <summary>
14     /// Provides functionality for saving and loading student records
15     /// to and from XML files.
16     /// </summary>
17     2 references
18     public class FileStorageService
19     {
20         /// <summary>
21         /// Saves a list of student records to an XML file.
22         /// </summary>
23         /// <param name="filePath">The file path where the XML file will be saved.</param>
24         /// <param name="students">The list of student records to save.</param>
25         1 reference
26         public void Save(string filePath, List<StudentRecord> students)
27         {
28             var doc = new XDocument(
29                 new XElement("Students",
30                     students.Select(s =>
31                         new XElement("Student",
32                             new XAttribute("Name", s.Name ?? ""),
33                             new XAttribute("Grade1", s.Grade1?.ToString() ?? ""),
34                             new XAttribute("Grade2", s.Grade2?.ToString() ?? ""),
35                             new XAttribute("Grade3", s.Grade3?.ToString() ?? ""),
36                             new XAttribute("Grade4", s.Grade4?.ToString() ?? ""),
37                             new XAttribute("Grade5", s.Grade5?.ToString() ?? ""),
38                             new XAttribute("Grade6", s.Grade6?.ToString() ?? ""),
39                             new XAttribute("Grade7", s.Grade7?.ToString() ?? ""),
40                             new XAttribute("Grade8", s.Grade8?.ToString() ?? ""),
41                             new XElement("Notes", s.Notes ?? "")
42                         )
43                     )
44             );
45             doc.Save(filePath);
46         }
47
48         /// <summary>
49         /// Loads student records from an XML file.
50         /// </summary>
51         /// <param name="filePath">The file path of the XML file to load.</param>
52         /// <returns>A list of student records loaded from the file.</returns>
53         1 reference
54         public List<StudentRecord> Load(string filePath)
55         {
56             var result = new List<StudentRecord>();
57
58             if (!File.Exists(filePath))
59                 return result;
60
61             var doc = XDocument.Load(filePath);
62             foreach (var elem in doc.Root.Elements("Student"))
63             {
64                 var s = new StudentRecord
65                 {
66                     Name = (string)elem.Attribute("Name")
67                 };
68
69                 s.Grade1 = ParseNullableDouble((string)elem.Attribute("Grade1"));
70                 s.Grade2 = ParseNullableDouble((string)elem.Attribute("Grade2"));
71                 s.Grade3 = ParseNullableDouble((string)elem.Attribute("Grade3"));
72                 s.Grade4 = ParseNullableDouble((string)elem.Attribute("Grade4"));
73                 s.Grade5 = ParseNullableDouble((string)elem.Attribute("Grade5"));
74                 s.Grade6 = ParseNullableDouble((string)elem.Attribute("Grade6"));
75                 s.Grade7 = ParseNullableDouble((string)elem.Attribute("Grade7"));
76                 s.Grade8 = ParseNullableDouble((string)elem.Attribute("Grade8"));
77
78                 s.Notes = (string)elem.Element("Notes") ?? "";
79
80                 result.Add(s);
81             }
82         }
83     }
84 }
```

GRADEBOOK.cs



```
FileStorageService.cs  GradeBook.cs  MainForm.cs  StudentRecord.cs  MainF...
C# CIS285Project  CIS285Project.Models.GradeBook  UpdateStuden...

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CIS285Project.Models
8  {
9      /// <summary>
10     /// Manages a collection of student records and provides
11     /// methods for modifying students and calculating class statistics.
12     /// </summary>
13     4 references
14     public class GradeBook
15     {
16         /// <summary>
17         /// Gets the list of students in the grade book.
18         /// </summary>
19         24 references
20         public List<StudentRecord> Students { get; private set; }
21
22         /// <summary>
23         /// Initializes a new instance of the <see cref="GradeBook"/> class.
24         /// </summary>
25         1 reference
26         public GradeBook()
27         {
28             Students = new List<StudentRecord>();
29         }
30
31         /// <summary>
32         /// Adds a new student to the grade book.
33         /// </summary>
34         /// <param name="s">The student record to add.</param>
35         20 references
36         public void AddStudent(StudentRecord s) => Students.Add(s);
37
38         /// <summary>
39         /// Updates an existing student at the specified index.
40         /// </summary>
41         /// <param name="index">The index of the student to update.</param>
42         /// <param name="s">The updated student record.</param>
43         1 reference
44         public void UpdateStudent(int index, StudentRecord s)
45         {
46             if (index >= 0 && index < Students.Count)
47                 Students[index] = s;
48         }
49
50         /// <summary>
51         /// Removes a student from the grade book at the specified index.
52         /// </summary>
53         /// <param name="index">The index of the student to remove.</param>
54         1 reference
55         public void RemoveStudent(int index)
56         {
57             if (index >= 0 && index < Students.Count)
58                 Students.RemoveAt(index);
59         }
60
61         /// <summary>
62         /// Calculates the average grade across all students.
63         /// </summary>
64         /// <returns>The class average.</returns>
65         2 references
66         public double CalculateClassAverage()
67         {
68             if (Students.Count == 0)
69                 return 0;
70
71             return Students.Average(s => s.Average);
72         }
73     }
74 }
```

## STUDENTRECORD.cs

```
FileStorageService.cs  GradeBook.cs  MainForm.cs  StudentRecord.cs  Mai
Data Sources
CIS285Project  CIS285Project.Models.StudentRecord  Grade4

4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CIS285Project.Models
8  {
9      /// <summary>
10     /// Represents a student record containing the student's name,
11     /// grades, notes, and calculated average.
12     /// </summary>
13     35 references
14     public class StudentRecord
15     {
16         /// <summary>
17         /// The student's full name.
18         /// </summary>
19         23 references
20         public string Name { get; set; }
21
22         /// <summary>First grade.</summary>
23         24 references
24         public double? Grade1 { get; set; }
25
26         /// <summary>Second grade.</summary>
27         24 references
28         public double? Grade2 { get; set; }
29
30         /// <summary>Third grade.</summary>
31         24 references
32         public double? Grade3 { get; set; }
33
34         /// <summary>Fourth grade.</summary>
35         24 references
36         public double? Grade4 { get; set; }
37
38         /// <summary>Fifth grade.</summary>
39         24 references
40         public double? Grade5 { get; set; }
41
42         /// <summary>Sixth grade.</summary>
43         24 references
44         public double? Grade6 { get; set; }
45
46         /// <summary>Seventh grade.</summary>
47         24 references
48         public double? Grade7 { get; set; }
49
50         /// <summary>Eighth grade.</summary>
51         24 references
52         public double? Grade8 { get; set; }
53
54         /// <summary>
55         /// Optional notes associated with the student.
56         /// </summary>
57         6 references
58         public string Notes { get; set; }
59
60         /// <summary>
61         /// Calculates the student's average grade based on all
62         /// non-null grade values.
63         /// </summary>
64         4 references
65         public double Average
66         {
67             get
68             {
69                 var grades = new double?[]
70                 {
71                     Grade1, Grade2, Grade3, Grade4,
72                     Grade5, Grade6, Grade7, Grade8
73                 };
74
75                 var valid = grades.Where(g => g.HasValue).Select(g => g.Value);
76                 return valid.Any() ? valid.Average() : 0;
77             }
78         }
79     }
80 }
```

OUTPUT:

Student Grade Book

Student Name	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8	Average
Jake Arieta	85	92	88	91	89	94	90	93	90.25
Anthony Rizzo	78	82	80	75	79	81	77	83	79.375
Troy Polamalu	94	90	96	98	99	97	95	99	96
Hines Ward	67	72	70	49	71	68	73	70	67.5
Tracy Cortez	58	84	87	60	86	69	75	82	75.125
Charles Oliveira	99	60	88	93	59	91	85	90	83.125
Julio Jones	99	95	100	98	97	99	96	100	98
Drake Maye	73	79	77	82	80	76	81	78	78.25
T.J. Watt	82	85	90	87	88	91	89	92	88
Sean O'Malley	91	89	93	90	92	94	95	96	92.5
Liv Morgan	76	81	79	83	80	78	82	79	79.75
Dave Bautista	65	70	68	62	69	71	74	73	69
Lonzo Ball	92	94	90	93	95	91	96	94	93.125
Emmanuel Mudiay	81	79	85	83	82	84	80	86	82.5
Minkah Fitzpatrick	89	91	92	90	93	94	95	92	92

Add Student

Edit Student

Delete Student

Calculate Average

Save to File

Load from File

Name:

Class Average: **82.32**

Class average updated.

Class Summary

Total Students: 19  
Class Avg: 82.32  
Highest Avg: 98.00  
Lowest Avg: 51.63  
Passing (>=70): 16

Notes for Selected Student:

Save Notes

### Source Code Reference:

This project was developed by our group and was not adapted from any external repository or tutorial project.