

פרויקטים שהוכנו על-ידי משתתפי קורס מורים מובילים תש"ע

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי

ללא קבלת אישור מראש מצוות הפיתוח

מערכת ניהול ציונים

איריס צור

שרה וגנר

אהובה שפרלינג

תוכן עניינים

2	תוכן עניינים.....
2	הקדשה/תודה:
3	מבוא.....
3	מטרות הפרויקט.....
3	תאור הפרויקט.....
3	הפרויקט ואני.....
3	כלי פיתוח.....
4	תרשים המחלקות (מידול הפרויקט).....
4	קוד המחלקות.....
10	הרצת הפרויקט.....
12	קוד התפריטים.....

הדרך לעדכן את תוכן העניינים הוא בסיום כתיבת הקבץ לסמן את כל שורות התוכן, בשורת התפריט :
הוספה < הפניה < אינדקס ורשימות < תוכן עניינים
על השאלה האם להחליף תוכן קיים- ענה כן.
לאחר ההחלפה יש לעדכן font, גודל, ריווח שורות.

יש להקפיד ולסמן כותרות בסגנון כותרת 1 וכותרת משנה בסגנון כותרת 2

הקדשה/תודה:

דף זה הינו חופשי וניתן גם לוותר עליו. בדף זה, לכותב ישנה הזדמנות להודות לאלו, שלדעתו, תרמו לו בעשייה, גיבוש וסיום עבודת הגמר. כמו כן, ניתן לצטט או להביע דעה חופשית לגבי העבודה או עשייתה, כגון: "מי שטרח בערב שבת...".

מבוא

מטרות הפרויקט

הפרויקט נועד לשמש מחנכים למתן ציונים לתלמידיהם במקצועות השונים ולקבל ממוצעים, תלמידים מצטיינים ותלמידים נכשלים.

[הפרויקט עוסק בנושא ...](#)
[הפרויקט נועד לענות על הדרישה ל...](#)

תאור הפרויקט

הפרויקט יאפשר למורה להזין ציונים לתלמידיו במקצועות השונים ולקבל נתונים על תלמיד ספציפי או נתונים סטטיסטיים על כיתתו.

הפרויקט יכלול שימוש במערכים חד-ממדיים וכן שימוש ב- Windows Form.
בנוסף כולל הפרויקט שימוש ב- Event, foreach, Object, properties, enum.

[הפרויקט נועד ...](#)
[הפרויקט כולל שימוש במערכים חד-ממדיים / דו-ממדיים...](#)
[יש לפרט את נושאי הלימוד הנוספים הממומשים בפרויקט זה](#)

הפרויקט ואני

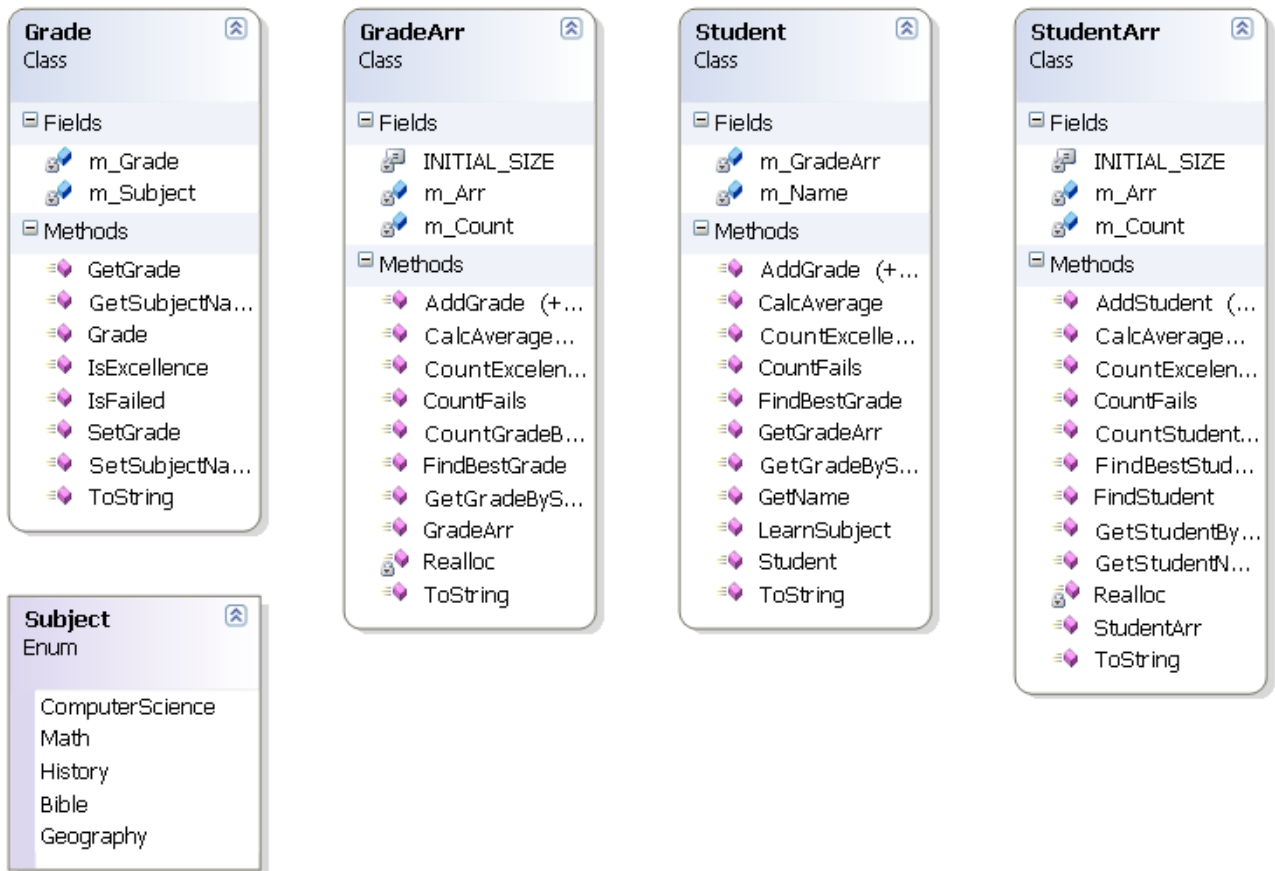
הפרויקט נבחר כך שיוכל להציג בצורה נוחה עבודה עם Windows Form.

[מספר מילים על מדוע בחרת לבנות פרויקט זה, באילו קשיים נתקלת, מה למדת במהלך העבודה](#)

כלי פיתוח

הפרויקט נבנה ב- Visual studio 2008 ונכתב בשפת C#.

תרשים המחלקות (מידול הפרויקט)



קוד המחלקות

```
public enum Subject {ComputerScience, Math, History, Bible, Geography };
public class Grade
{
    private Subject m_Subject;
    private int m_Grade;
    public Grade(Subject subject, int grade)
    {
        this.m_Subject = subject;
        this.m_Grade = grade;
    }
    public Subject GetSubjectName()
    {
        return this.m_Subject;
    }
    public void SetSubjectName(Subject subject)
    {
        this.m_Subject = subject;
    }
    public int GetGrade()
    {
        return this.m_Grade;
    }
    public void SetGrade(int grade)
    {
        this.m_Grade = grade;
    }
}
```

```

public bool IsFailed()
{
    return this.m_Grade < 56;
}
public bool IsExcellence()
{
    return this.m_Grade > 90;
}
public override string ToString()
{
    return string.Format("Subject: {0} - Grade: {1}",
        this.m_Subject, this.m_Grade);
}
}

public class GradeArr
{
    private const int INITIAL_SIZE = 10;
    private Grade[] m_Arr;
    private int m_Count;
    public GradeArr()
    {
        this.m_Arr = new Grade[INITIAL_SIZE];
        this.m_Count = 0;
    }
    public void AddGrade(Grade grade)
    {
        this.Realloc();
        this.m_Arr[m_Count] = grade;
        this.m_Count++;
    }
    public void AddGrade(Subject subject, int grade)
    {
        Grade tmp_grade = new Grade(subject, grade);
        this.AddGrade(tmp_grade);
    }
    private void Realloc()
    {
        if (this.m_Arr.Length >= this.m_Count)
            Array.Resize(ref this.m_Arr, this.m_Arr.Length + 10);
    }
    public override string ToString()
    {
        string str = string.Empty;
        for(int i=0;i<m_Count;i++)
            str += m_Arr[i].ToString() + "\n";
        return str;
    }
    public int CalcAverage()
    {
        int avg = 0;
        for(int i=0;i<this.m_Count;i++)// (Grade grade in this.m_Arr)
            avg += this.m_Arr[i].GetGrade();
        return avg / this.m_Count;
    }
    public int CountGradeBySubject(Subject subject)
    {
        int counter = 0;
        for(int i=0;i<m_Count;i++)
            if (m_Arr[i].GetSubjectName() == subject)
                counter++;
        return counter;
    }
}

```

```

public int CalcAverage(Subject subject)
{
    int avg = 0;
    int counter = 0;
    for (int i = 0; i < this.m_Count; i++)// (Grade grade in this.m_Arr)
    {
        if (this.m_Arr[i].GetSubjectName() == subject)
        {
            avg += this.m_Arr[i].GetGrade();
            counter++;
        }
    }
    if (counter > 0)
        return avg / counter;
    else
        return -1;
}
public int CountFails()
{
    int counter = 0;
    for(int i=0;i<m_Count;i++)
        if (m_Arr[i].IsFailed())
            counter++;
    return counter;
}
public int CountExcelence()
{
    int counter = 0;
    for (int i = 0; i < m_Count; i++)
        if (m_Arr[i].IsExcelence())
            counter++;
    return counter;
}
public Grade FindBestGrade()
{
    Grade grade = this.m_Arr[0];
    for (int i = 1; i < this.m_Count; i++)
    {
        if(this.m_Arr[i].GetGrade()>grade.GetGrade())
            grade = this.m_Arr[i];
    }
    return grade;
}
public int[] GetGradeBySubject(Subject subject)
{
    int[] arr = new int[this.CountGradeBySubject(subject)];
    int counter = 0;
    for (int i = 0; i < m_Count; i++)
    {
        if (m_Arr[i].GetSubjectName() == subject)
        {
            arr[counter] = m_Arr[i].GetGrade();
            counter++;
        }
    }
    return arr;
}
}

```

```
public class Student
{
    private string m_Name;
    private GradeArr m_GradeArr;
    public Student(string name)
    {
        this.m_Name = name;
        this.m_GradeArr = new GradeArr();
    }
    public void AddGrade(Grade grade)
    {
        this.m_GradeArr.AddGrade(grade);
    }
    public void AddGrade(Subject subject, int grade)
    {
        this.m_GradeArr.AddGrade(subject, grade);
    }
    public string GetName()
    {
        return this.m_Name;
    }
    public GradeArr GetGradeArr()
    {
        return this.m_GradeArr;
    }
    public int CalcAverage()
    {
        return this.m_GradeArr.CalcAverage();
    }
    public int CountFails()
    {
        return this.m_GradeArr.CountFails();
    }
    public int CountExcellence()
    {
        return this.m_GradeArr.CountExcellence();
    }
    public Grade FindBestGrade()
    {
        return this.m_GradeArr.FindBestGrade();
    }
    public int[] GetGradeBySubject(Subject subject)
    {
        return this.m_GradeArr.GetGradeBySubject(subject);
    }

    public bool LearnSubject(Subject subject)
    {
        return ( m_GradeArr.CountGradeBySubject(subject)>0)
    }
    public override string ToString()
    {
        return this.m_Name + "/n" + this.m_GradeArr.ToString();
    }
}
```

```

public class StudentArr
{
    private const int INITIAL_SIZE = 10;
    private Student[] m_Arr;
    private int m_Count;
    public StudentArr()
    {
        this.m_Arr = new Student[INITIAL_SIZE];
        this.m_Count = 0;
    }
    public void AddStudent(Student student)
    {
        this.Realloc();
        this.m_Arr[m_Count] = student;
        this.m_Count++;
    }
    public void AddStudent(string student_name )
    {
        Student tmp_student = new Student(student_name);
        this.AddStudent(tmp_student);
    }
    public Student FindStudent(string student_name)
    {
        for (int i = 0; i < m_Count; i++)
        {
            if (m_Arr[i].GetName() == student_name)
                return m_Arr[i];
        }
        return null;
    }
    private void Realloc()
    {
        if (this.m_Arr.Length >= this.m_Count)
            Array.Resize(ref this.m_Arr, this.m_Arr.Length + 10);
    }
    public override string ToString()
    {
        string str = string.Empty;
        for (int i = 0; i < m_Count; i++)
        {
            str += m_Arr[i].ToString();
            str += "/n";
        }
        return str;
    }
    public int CalcAverage()
    {
        int avg = 0;
        for (int i = 0; i < this.m_Count; i++)
        {
            avg += this.m_Arr[i].GetGradeArr().CalcAverage();
        }
        return avg / this.m_Count;
    }
    public int CountFails()
    {
        int counter = 0;
        for (int i = 0; i < this.m_Count; i++)
        {
            counter += m_Arr[i].GetGradeArr().CountFails();
        }
        return counter;
    }
}

```



```

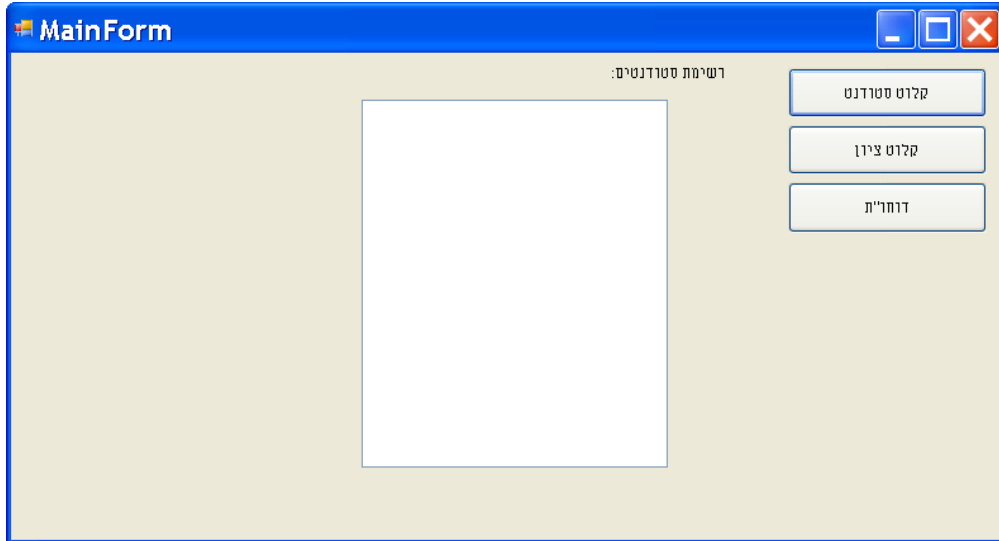
public int CalcAverage(Subject subject)
{
    int avg = 0;
    int counter = 0;
    for (int i = 0; i < this.m_Count; i++)
    {
        avg += this.m_Arr[i].GetGradeArr().CalcAverage(subject);
    }

    if (counter > 0)
        return avg / counter;
    else
        return -1;
}
public int CountExcelece()
{
    int counter = 0;
    for (int i = 0; i < this.m_Count; i++)
    {
        counter += m_Arr[i].GetGradeArr().CountExcelece();
    }
    return counter;
}
public Student FindBestStudent()
{
    Student student = this.m_Arr[0];
    for (int i = 1; i < this.m_Count; i++)
    {
        if (this.m_Arr[i].CalcAverage() > student.CalcAverage())
            student = this.m_Arr[i];
    }
    return student;
}
public int CountStudentBySubject(Subject subject)
{
    int counter = 0;
    for (int i = 1; i < this.m_Count; i++)
    {
        if (m_Arr[i].LearnSubject(subject))
        {
            counter++;
        }
    }
    return counter;
}
public int GetStudentNumber()
{
    return m_Count;
}
public Student GetStudentByIndex(int index)
{
    return m_Arr[index];
}
}

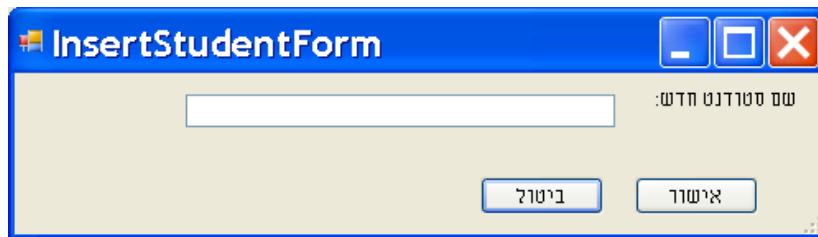
```

הרצת הפרויקט

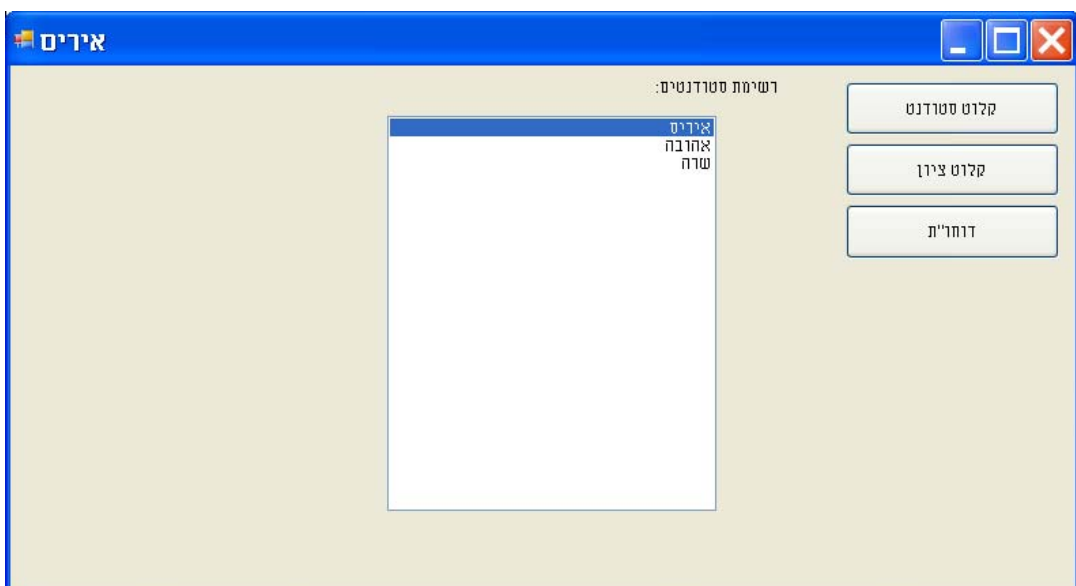
להלן התפריט הראשי של הפרויקט:



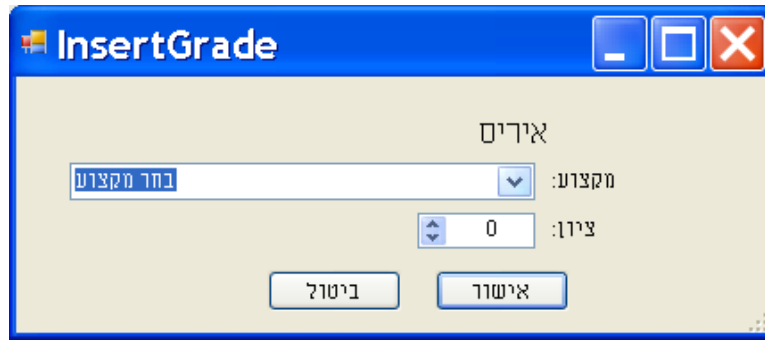
לחיצה על קלט סטודנט תציג את הטופס:



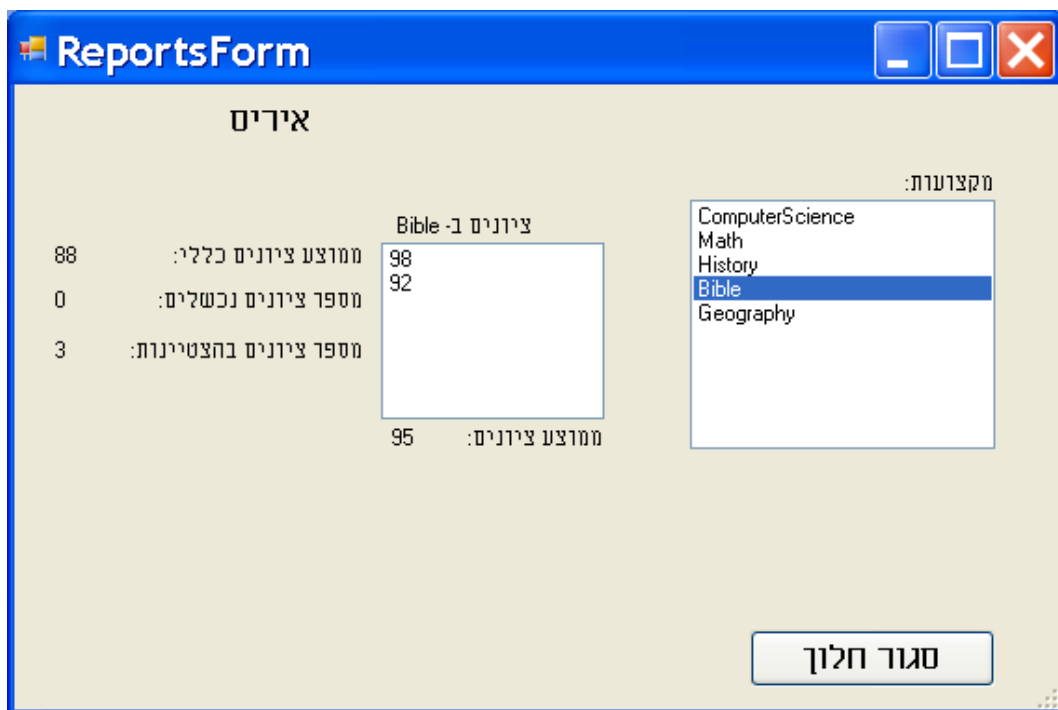
לאחר הוספת שם סטודנט יוצג שוב התפריט הראשי עם רשימת הסטודנטים:



בחירת אחד הסטודנטים ולחיצה על קלט ציון תציג את תפריט הוספת ציון:



תפריט זה מאפשר לבחור מקצוע ולתת לו ציון.
בחירה בכפתור דוח"ת תציג דו"ח ציוני התלמיד:



בדו"ח זה ניתן לצפות בציוני התלמיד על פי המקצועות השונים ולראות את ממוצע ציוניו במקצוע, ממוצע ציוניו הכללי, כמות הציונים הנכשלים וכמות הציונים המצטיינים של אותו תלמיד.

קוד התפריטים

partial class MainForm : Form

```

{
    private StudentArr m_StudentArr = new StudentArr();
    public MainForm()
    {
        InitializeComponent();
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        //load students' name list
        InitStudentListBox();
    }
    private void InitStudentListBox()
    {
        //list box – נקחה את רשימת התלמידים שנמצאית ב-
        lstStudents.Items.Clear();
        // נעבור על כל התלמידים במערך
        for (int i = 0; i < m_StudentArr.GetStudentNumber(); i++)
        {
            // נוסיף את שם התלמיד לרשימת התלמידים המוצגת
            lstStudents.Items.Add(m_StudentArr.GetStudentByIndex(i).GetName());
        }
    }
    private void btnInsertStudent_Click(object sender, EventArgs e)
    {
        //ניצור טופס הוספת תלמיד
        InsertStudentForm frm = new InsertStudentForm();
        //אם נלחץ כפתור ה-OK
        if (frm.ShowDialog() == DialogResult.OK)
        {
            //אם הוכנס שם לא ריק
            if (frm.StudnetName() != string.Empty)
            {
                //ניצור תלמיד חדש ששמו ילקח מהטופס
                Student s = new Student(frm.StudnetName());
                //נוסיף תלמיד זה למערך התלמידים
                m_StudentArr.AddStudent(s);
            }
            //נציג מחדש את רשימת התלמידים בטופס הראשי
            InitStudentListBox();
        }
    }
    private void btnInsertGrade_Click(object sender, EventArgs e)
    {
        //יש להכניס לתלמיד קיים ציון
        //שם התלמיד הנבחר
        string student_name = string.Empty;

```

```

if (lstStudents.SelectedItem != null)
    student_name = lstStudents.SelectedItem.ToString();
// אם נבחר שם
if (student_name != string.Empty)
{
    // צור טופס הכנסת ציון עבור התלמיד הנבחר
    InsertGradeForm frm = new InsertGradeForm(student_name);
    if (frm.ShowDialog() == DialogResult.OK)
    {
        // מצא את התלמיד במערך התלמידים על פי שמו
        Student student = m_StudentArr.FindStudent(student_name);
        // צור ציון חדש במקצוע
        Grade grade = new Grade(frm.TheSubject(), frm.Grade());
        // הוסף את הציון למערך ציוני התלמיד
        student.GetGradeArr().AddGrade(grade);
    }
}
else // אם לא נבחר שם
{
    MessageBox.Show("נדרש לקלוט שם תלמיד לפני הוספת ציון");
}
}

private void btnReport_Click(object sender, EventArgs e)
{
    if (lstStudents.SelectedItem != null)
    {
        // שם התלמיד הנבחר
        string student_name = lstStudents.SelectedItem.ToString();
        // מצא את התלמיד במערך התלמידים על פי שמו
        Student student = m_StudentArr.FindStudent(student_name);
        // פתח טופס דוח עבור התלמיד
        ReportsForm frm = new ReportsForm(student);
        // הצג את הטופס
        frm.ShowDialog();
    }
    else
        MessageBox.Show("נדרש לבחור שם תלמיד לצורך הפקת הדוח");
}

private void lstStudents_SelectedIndexChanged(object sender, EventArgs e)
{
    // אם נבחר תלמיד מרשימת התלמידים
    if (lstStudents.SelectedItem != null)
    {
        // שנה את שם הטופס להיות שם התלמיד שנבחר
        this.Text = lstStudents.SelectedItem.ToString();
    }
}
}
}

```

```

public partial class InsertStudentForm : Form
{
    public InsertStudentForm()
    {
        InitializeComponent();
    }

    private void btnCancel_Click(object sender, EventArgs e)
    {
        // ביטול הוספת התלמיד
        this.DialogResult = DialogResult.Abort;
        // סגירת חלון הוספת התלמיד
        this.Close();
    }

    private void btnOK_Click(object sender, EventArgs e)
    {
        // הוספת התלמיד
        this.DialogResult = DialogResult.OK;
        // סגירת חלון הוספת תלמיד
        this.Close();
    }

    public string StudnetName()
    {
        // החזרת שם התלמיד שהוסף
        return txtStudentName.Text;
    }
}

public partial class InsertGradeForm : Form
{
    public InsertGradeForm(string student_name)
    {
        InitializeComponent();
        // הצגת שם התלמיד אליו יוכנס הציון
        lblStudentName.Text = student_name;
    }

    private void InsertGradeForm_Load(object sender, EventArgs e)
    {
        // בעת טעינת הדף יש לטעון את שמות המקצועות
        Type enumType = typeof(Subject);
        // יצירת מערך שמות המקצועות
        // הפעולה GetValues יוצרת מערך של ערכי enum
        Array enum_val_array = Enum.GetValues(enumType);
        // יצירת קומבו-בוקס שבשורתה הראשונה המחזרות בחר מקצוע
        cboSubjects.Items.Add("מקצוע בחר");
        // מעבר על מערך המקצועות
        /*
        * enum is a Value Type therefor not an Object but an object
        */
        //foreach מעבר כלולאת
    }
}

```

```

foreach (object obj in enum_val_array)
{
    // הוסף את הנושא לקומבו-בוקס
    cboSubjects.Items.Add(obj.ToString());
}
// בחירת הערך הראשון בקומבו-בוקס
cboSubjects.SelectedIndex = 0;
}
private void btnCancel_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.Abort;
    this.Close();
}
private void btnOK_Click(object sender, EventArgs e)
{
    // אם נבחר מקצוע
    if (cboSubjects.Text != "מקצוע")
    {
        this.DialogResult = DialogResult.OK;
        this.Close();
    }
    else
    {
        MessageBox.Show("נדרש לבחור מקצוע לפני קביעת ציון");
    }
}
}
public int Grade()
{
    // החזרת ערך הציון
    // שימו לב להמרה מדצימלי לשלם
    return (int)numGrade.Value;
}
}
public Subject TheSubject()
{
    //המרת טקסט לערך של enum
    switch (cboSubjects.Text)
    {
        case "ComputerScience":
            return Subject.ComputerScience;
        case "Math":
            return Subject.Math;
        case "History":
            return Subject.History;
        case "Bible":
            return Subject.Bible;
        case "Geography":
            return Subject.Geography;
    }
    return Subject.ComputerScience;
}
}
}

```