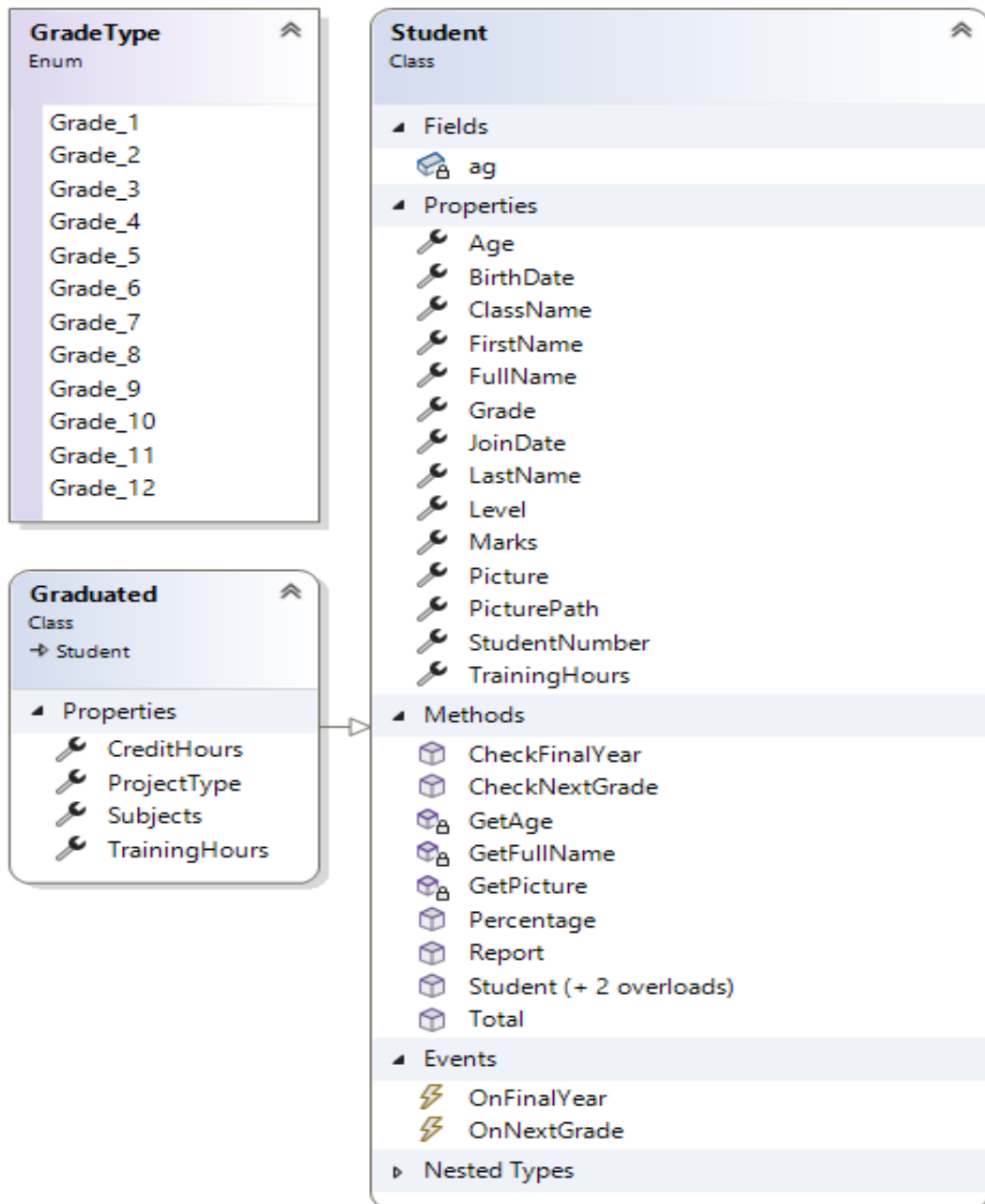


Task Description

Create Classes As Following (Please see all 6 Pages)



1- Create GradeType Enum As shown

2- Create Student Class With:

- **Class Constructors: 3 Overloads**
 - default
 - with first & Last name
 - with first & Last And BirthDate
- **Properties**
 - FullName & Age Properties are ReadONLY
 - Grade Property Based on GradeType Enum
 - Marks Property is a Double[]
 - Picture Property is a Byte[]
 - JoinDate Must be Over BirthDate by 6 years
- **Private Methods:**
 - GetPicture() → Render picture Bytes
 - GetFullName()--> Concatenate First & Last name
 - GetAge() --> calculate Age As Integer
- **Public Methods**
 - Total() ---> Sum Marks Array
 - Percentage()--> Divide Total()/Marks Length *100
 - Report() --> Show student Full name With Grade Level And Join Date And Age (See User Interface Pic)

NOTE: Please Mark All Public Methods as virtual

- **Event Raise Methods:**
 - CheckFinalYear() --> when Grade Property value =Grade_12
 - CheckGrade() -->When Percentage is over .5
- **Events With Delegates:**
 - OnFinalYear
 - OnNextGrad

3- Create Graduated Class Inherits from Students

Add the following new Properties:

CreditHours

ProjectType

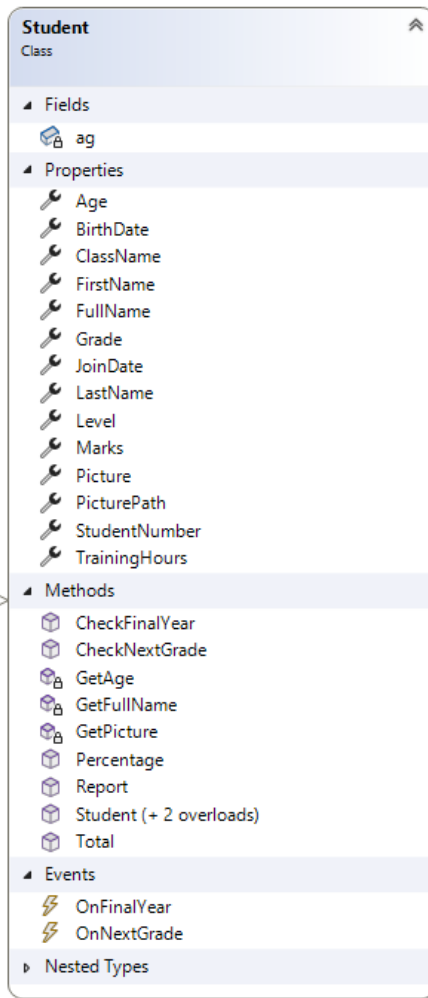
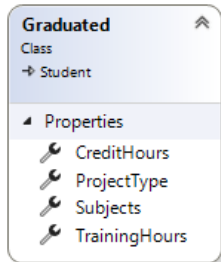
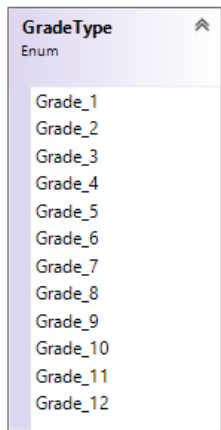
override:

JoinDate Property --> Must be Over 18 years

Training Hours Property --> Must Start from 100

Report() Method

Classes Diagram



1- Create Grade Type Enum As shown
 2- Create Student Class With:
 Class Constructors: 3 Overloads
 default
 with first & Last name
 with first & Last And BirthDate
 -FullName & Age Properties are ReadONLY
 -Grade Property Based on GradeType Enum
 -Marks Property is a Double[]
 -Picture Property is a Byte[]
 -JoinDate Must be Over BirthDate by 6 years

*Private Methods:
 GetPicture()
 GetFullName() --> Concat First & Last name
 GetAge() --> calculate Age As Integer

*Public Methods
 Total() ---> Sum Marks Array
 Percentage() --> Divide Total()/Marks Length *100
 Report() --> Show student Full name With Grade Level And Join Date And Age (See Interface Pic)

NOTE: Please Mark All Public Methods as virtual

Event Raise Methods:
 CheckFinalYear() --> when Grade Property value =Grade_12
 CheckGrade() -->When Percentage is over .5

Events:
 OnFinalYear
 OnNextGrade

3- Create Graduated Class Inherits from Students
 override:
 JoinDate Property --> Must be Over Bith by 18 year
 Training Hours Propert --> Must Start from 100
 Report() Method

4- Create User Interface to Implement Student Object As show in Picture

User Interface

4- Create User Interface to Implement Student Object As show in Picture

Form1


First Name: Mostafa

First Name: Sayed Mohamed

Birth Date: Wednesday, October 1, 2014

Join Date: Tuesday, October 1, 2024

Grade Level: Grade_6

Student Picture:  Browse

Save Student

Student Name: Mostafa Sayed Mohamed
Grade: Grade_6
Join at: 10/1/2024 1:53:18 PM
With Age 9

button1

5- Using EntityFrameWork Package Create a database using previous 2 Classes (Students & Graduated)

- Use Program Interface to Add Student to database table
- Use Program Interface to Add Graduated to Database table

Best Wishes & Good Luck