Contents

Description

Purpose

This project is designed to teach you how to interpret and implement a simple UML specification involving multiple classes. It involves inheritance, polymorphism, properties, exception handling and `String` methods.

Challenge

In short

Develop multiple classes to organize the evaluations for courses. Evaluations can be of different natures (project, in-class exam, finale exam), have different weights and be marked as optional. However, evaluations of different natures also have different attributes and methods: for example, only projects have a submission link, and only exams have a room.

In more details

We want to implement the classes pictured in the UML diagram (you can zoom in or, in your browser, right click and select "Open Image in New Tab", or access its text version[1]).

Remembering that:

- Static methods are u n d e r l i n e d,
- Abstract methods are *in italics*.

Your code should be such that the following:

```
using System;
using System.Collections.Generic;
```
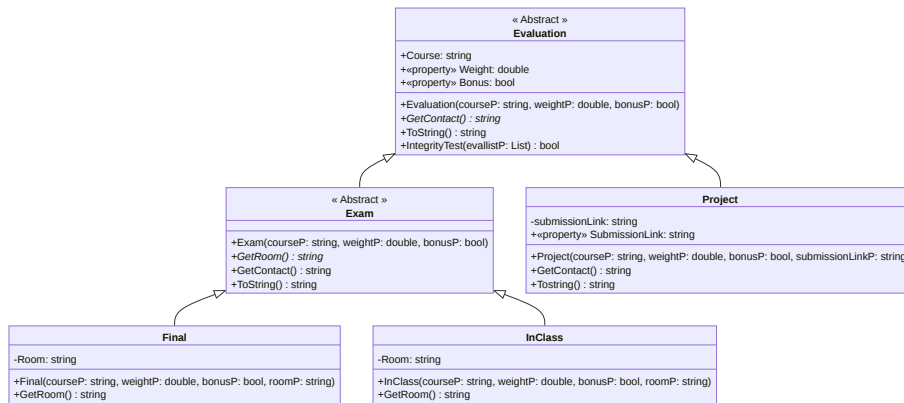
---

[1]https:/princomp.github.io/uml/cla/Evaluation.txt

```
« Abstract »
Evaluation

+Course: string
+«property» Weight: double
+«property» Bonus: bool

+Evaluation(courseP: string, weightP: double, bonusP: bool)
+GetContact() : string
+ToString() : string
+IntegrityTest(evallistP: List) : bool
```

```
« Abstract »
Exam

+Exam(courseP: string, weightP: double, bonusP: bool)
+GetRoom() : string
+GetContact() : string
+ToString() : string
```

```
Project

-submissionLink: string
+«property» SubmissionLink: string

+Project(courseP: string, weightP: double, bonusP: bool, submissionLinkP: string)
+GetContact() : string
+Tostring() : string
```

```
Final

-Room: string

+Final(courseP: string, weightP: double, bonusP: bool, roomP: string)
+GetRoom() : string
```

```
InClass

-Room: string

+InClass(courseP: string, weightP: double, bonusP: bool, roomP: string)
+GetRoom() : string
```

Figure 1: A UML diagram for the Evaluation Exam class (text version[2])

```csharp
using System.IO;

class Program
{
  static void Main()
  {
    // Helper string, to display more nicely
    string sep = "\n\n" + new String('*', 20) + "\n\n";

    // Evaluation test = new Evaluation(); // Should
    ↪  return an error if uncommented.

    Project project1 = new Project(
      "CSCI 1302",
      .05,
      false,

      ↪  "https://lms.augusta.edu/d2l/lms/dropbox/user/folder_submit_files.d2l?
    );
    Project bonusproject1 = new Project(
      "CSCI 1302",
      .05,
      true,

      ↪  "https://lms.augusta.edu/d2l/lms/dropbox/user/folder_submit_files.d2l?
    );
    Project project2 = new Project(
      "CSCI 1302",
      .05,
```

```csharp
    false,
    ↪    "https://lms.augusta.edu/d2l/lms/dropbox/user/folder_submit_files.d2l?
);

Console.Write(sep);
try
{
  Project badUrl = new Project(
    "CSCI 1302",
    .1,
    false,
    "https://sketchy-url.com/totallyLegitProject"
  );
}
catch (Exception e)
{
  Console.Write(e.Message);
}
Console.Write(sep);

// Exam test = new Exam(); // Should return an error
↪  if uncommented.
InClass inclassexam1 = new InClass(
  "CSCI 1302",
  .25,
  false,
  "University Hall 243"
);
InClass inclassexam2 = new InClass(
  "CSCI 1302",
  .25,
  false,
  null
);
Final finaleexam = new Final(
  "CSCI 1302",
  .4,
  false,
  ""
);

Console.Write(
  project1
    + sep
    + project2
```

```csharp
        + sep
        + inclassexam1
        + sep
        + inclassexam2
        + sep
        + finaleexam
);

List<Evaluation> listEval1 = new List<Evaluation>()
{
  project1,
  project2,
  inclassexam1,
  inclassexam2,
  finaleexam,
  bonusproject1,
};

Console.Write(sep);

Console.Write(
  "Your list of non-bonus evaluations sums up to 100%:
    ↳  "
    + Evaluation.IntegrityTest(listEval1)
);

List<Evaluation> listEval2 = new List<Evaluation>()
{
  project1,
  inclassexam2,
  finaleexam,
  bonusproject1,
};

Console.Write(sep);

Console.Write(
  "Your list of non-bonus evaluations sums up to 100%:
    ↳  "
    + Evaluation.IntegrityTest(listEval2)
);

Console.Write(sep);

InClass inclassexam3 = new InClass(
  "CSCI 1301",
```

```csharp
          .50,
          false,
          null
      );
      InClass inclassexam4 = new InClass(
          "CSCI 1302",
          .50,
          false,
          null
      );
      List<Evaluation> listEval3 = new List<Evaluation>()
      {
          inclassexam3,
          inclassexam4,
      };
      try
      {
          Evaluation.IntegrityTest(listEval3);
      }
      catch (Exception e)
      {
          Console.Write(e.Message);
      }
      Console.Write(sep);

      /* (Bonus)
       * Testing the export into file /
       * import from file methods.
       */
      string directoryPath = AppDomain
          .CurrentDomain
          .BaseDirectory;
      string filePath = Path.Combine(
          directoryPath,
          "finaleexam.txt"
      );
      finaleexam.ExportToFile(filePath);
      // Make sure that the file was created at
      // bin/Debug/finaleexam.txt
      Final copy = Final.ImportFromFile(filePath);
      Console.WriteLine(copy);
  }
}
```

should display (something along the lines of)

****

```
Submission link
↪ (https://sketchy-url.com/totallyLegitProject) does
↪ not start with "https://lms.augusta.edu/d2l/lms/".
Parameter name:
↪ https://sketchy-url.com/totallyLegitProject

********************

Course:         CSCI 1302
Weight:         5,00 %
Contact:        Check the LMS submission folder.
Link:
↪ https://lms.augusta.edu/d2l/lms/dropbox/user/folder_submit_files.d2l?db=2302

********************

Course:         CSCI 1302
Weight:         5,00 %
Contact:        Check the LMS submission folder.
Link:
↪ https://lms.augusta.edu/d2l/lms/dropbox/user/folder_submit_files.d2l?db=2325

********************

Course:         CSCI 1302
Weight:         25,00 %
Contact:        Contact the CSCI 1302 instructor.
Room:           University Hall 243

********************

Course:         CSCI 1302
Weight:         25,00 %
Contact:        Contact the CSCI 1302 instructor.
Room:           Same as class meeting room.

********************

Course:         CSCI 1302
Weight:         40,00 %
Contact:        Contact the CSCI 1302 instructor.
Room:           Check the Registrar's exam schedules for
↪ CSCI 1302.

********************
```

```
Your list of non-bonus evaluations sums up to 100%: True

********************

Evaluations are for different classes.

********************
```

Pay attention to details in the code above. For example, your code should be such that:

- An exception will be raised if the url for the project's submission link does not start with "https://lms.augusta.edu/d2l/lms/",
- For exams (in-class or finale exam), the room should be displayed if it is provided, otherwise, "Same as class meeting room." should be displayed if the exam in in-class, and "Check the Registrar's exam schedules for" (followed by the course property) should be displayed if the exam is the final exam,
- For all string attributes and properties, **null** or empty strings should *not* make the program crash,
- The *static* `IntegrityTest` method should take as input a list of evaluations, and returns **true** if all the evaluations belong to the same course and the weights of the evaluations that are not bonuses adds to 1, **false** if the evaluations all belong to the same course but the weights of the evaluations that are not bonuses do not add to 1, and raise an exception otherwise,
- `Final` and `InClass` do *not* have `ToString()` methods, and yet displaying information about them still return information about the room.

Submission

Please, follow our guideline on project submission[3]. In particular, make sure you write your name and the date in a delimited comment at the beginning of your file.

Bonuses

This project is already challenging *as is*, but bonus will be given if:

- (easy) Except in the `Evaluation` class, make it so that the `GetContact()` methods cannot be overridden,
- (medium) Your `Main` method exhibits many test cases and illustrates your classes' features nicely,

---

[3]https://princomp.github.io/projects/submission

- (hard) A mechanism is implemented to either export the attributes of an `Evaluation` object into a file or to load it from a file.