

Same diagram below, just flipped.

Umlple Online

Draw on the right, write (Umlple) model code on the left. Analyse models and generate code.

This tool stores your data in cookies and on a server. [I Understand](#). [Click to learn about privacy](#).

For help: [User manual](#) [Ask questions](#) [Report issue](#)

[Download](#) [Donate](#)

Umlple Online

Generate Java [Save as URL](#) [Show Tabs](#)

SAVE & LOAD

TOOLS

OPTIONS

DIAGRAM TYPE

- ☒ Editable Class
- ☐ JointJS Class
- ☐ GraphViz Class
- ☐ GraphViz State
- ☐ GraphViz Feature
- ☐ Composite Structure

SHOW VIEW

- ☒ Diagram (Canvas)
- ☒ Text Editor
- ☐ Layout Editor
- ☒ Attributes
- ☒ Methods
- ☐ Traits

PREFERENCES

- ☐ Photo Ready
- ☐ Manual Sync
- ☐ Pinch to Zoom

TASKS

```

Line= 18
1 String email;
2 }
3
4
5
6
7 class Student{
8     studentID;
9     phoneNum;
10    String[] selCourses;
11    int cGPA;
12    public Note makeANote(){
13
14    }
15    isA Person;
16    1--1 Grade;
17
18    4..* -- * Tournament;
19 }
20
21 class Professor{
22     public Course
23     createCourse(){
24
25 }
26
27 public Tournament
28 createTournament(){
29
30 }
31
32 public Question
33 createQuestion(){
34
35 }
36
37 isA Person;
38 1 -- 1..* Question;
39 1..* -- * Tournament;
40
41
42 class Post{ tag;
43
44 }
45
46 class Note{
47     summary/title;
48     details;
49     isA Post;
50
51     * -- * ReadingList;
52     * -- * Professor;
53     * -- * Student;
54 }
55
56 class Question{
57     createQuestion()
58     }
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Person

name : String

email : String

password : String

Account

email : String

password : String

Student

studentID : String

phoneNum : String

selCourses : String[]

cGPA : int

Professor

createCourse() : Course

createTournament() : Tournament

createQuestion() : Question

makeANote() : Note

Course

courseCode : String

title : String

semester : String

date : String

description : String

status : enum

Grade

createGrade() : void

ReadingList

listNotes : Note[]

Tournament

startDate : String

startTime : String

questions : Question[]

roundsNum : int

maxAttempterQ : int

roundLength : int

penalty : int

weightCorrectness : int

weightParticipation : int

Question

gradeTry : String

gradeCorrectness : String

status : enum

text : String

answer : String

transformAnswerToString() : String

Section

classList : String[]

Homework

deadlineDate : String

deadlineTime : String

Suggestion: Since there are so many classes, switch to automated layout (G).

UML Diagram:



Draw on the right, write (Uml) model code on the left. Analyse models and generate code.

This tool stores your data in cookies and on a server. [I understand](#). [Click to learn about privacy](#).

[Download](#) [Donate](#) For help: [User manual](#) [Ask questions](#) [Report issue](#)



Line= 18

E G S T D A M

Generate Java

Save as URL

Show Tabs

Suggestion: Since there are so many classes, [switch to automated layout](#) (G).

```
4 String email;
5 }
6
7 class Student{
8     studentID;
9     phoneNum;
10    String[] selCourses;
11    int cGPA;
12    public Note makeANote(){
13
14    }
15    isA Person;
16    1--1 Grade;
17
18    4..* -- * Tournament;
19 }
20
21 class Professor{
22     public Course
23     createCourse(){
24
25 }
26 public Tournament
27 createTournament(){
28
29 }
30 public Question
31 createQuestion(){
32
33 }
34 isA Person;
35 1 -- 1..* Question;
36 1..* -- * Tournament;
37
38 }
39
40 class Post{ tag;
41
42 }
43
44 class Note{
45     summaryTitle;
46     details;
47     isA Post;
48
49     * -- * ReadingList;
50     * -- * Professor;
51     * -- * Student;
52 }
53
54 class Question{
55     gradeTry;
56     gradeCorrectness;
57     status;
58     text;
59     answer;
60     + transformAnswerToString();
61 }
```

SAVE & LOAD

TOOLS

OPTIONS

DIAGRAM TYPE

- ☒ Editable Class
- ☐ JointJS Class
- ☐ GraphViz Class
- ☐ GraphViz State
- ☐ GraphViz Feature
- ☐ Composite Structure

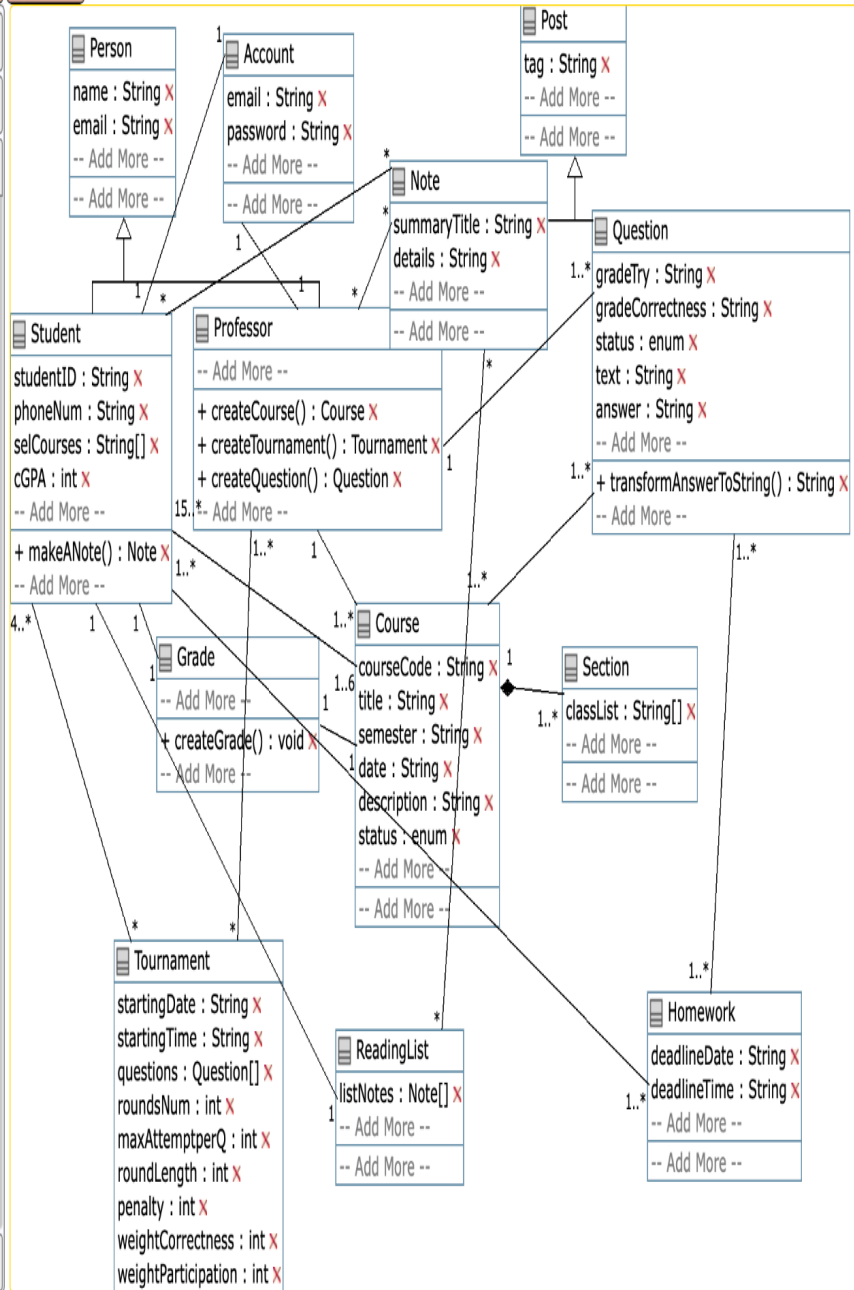
SHOW VIEW

- ☒ Diagram (Canvas)
- ☒ Text Editor
- ☐ Layout Editor
- ☒ Attributes
- ☒ Methods
- ☐ Traits

PREFERENCES

- ☐ Photo Ready
- ☐ Manual Sync
- ☐ Pinch to Zoom

TASKS



UML Diagram:

Assumptions:

We are assuming:

- Each student can only have one account and vice versa
- A professor can have 0 to many notes and 0 to many notes and 0 to many notes can only be made by one professor.
- A professor must have at least one question posted and these questions can only be made by one professor.
- A course can have no question at all to many questions if wanted and 0 to many question can be assigned to a course.
- A course will have at least one section and each section is only going to be for one course only.
- A professor will be teaching at least one course, and each course will only be taught by one professor.
- A student must have from 1 up until 6 courses selected to be enrolled, and these courses could only be taken if there is a min of 15 students or more.
- A course will have at least one question posted, and at least one question will be posted in a given course.
- Each course will have one passing grade and grade can be for 1 to many courses.
- Each student will have a grade for each course used towards his cGPA, and a grade will be assigned to a student.
- 0 to many notes can be added to a readingList, and a readingList can have 0 to many notes.
- A student can only have one reading list, and vice versa
- A homework will have at least one question and 0 to many questions can be assigned as a homework.
- 1 to many Homework can be assigned for a student, and a student must have at least one homework.
- A professor can create 0 to many tournaments, and 0 to many tournaments can be created by at least one professor.
- At least 4 students are requires to host a tournament, and a student can have 0 to many students.
- A section is a part that can't be present unless there is a course, that is why we used a decomposition association between these two classes.