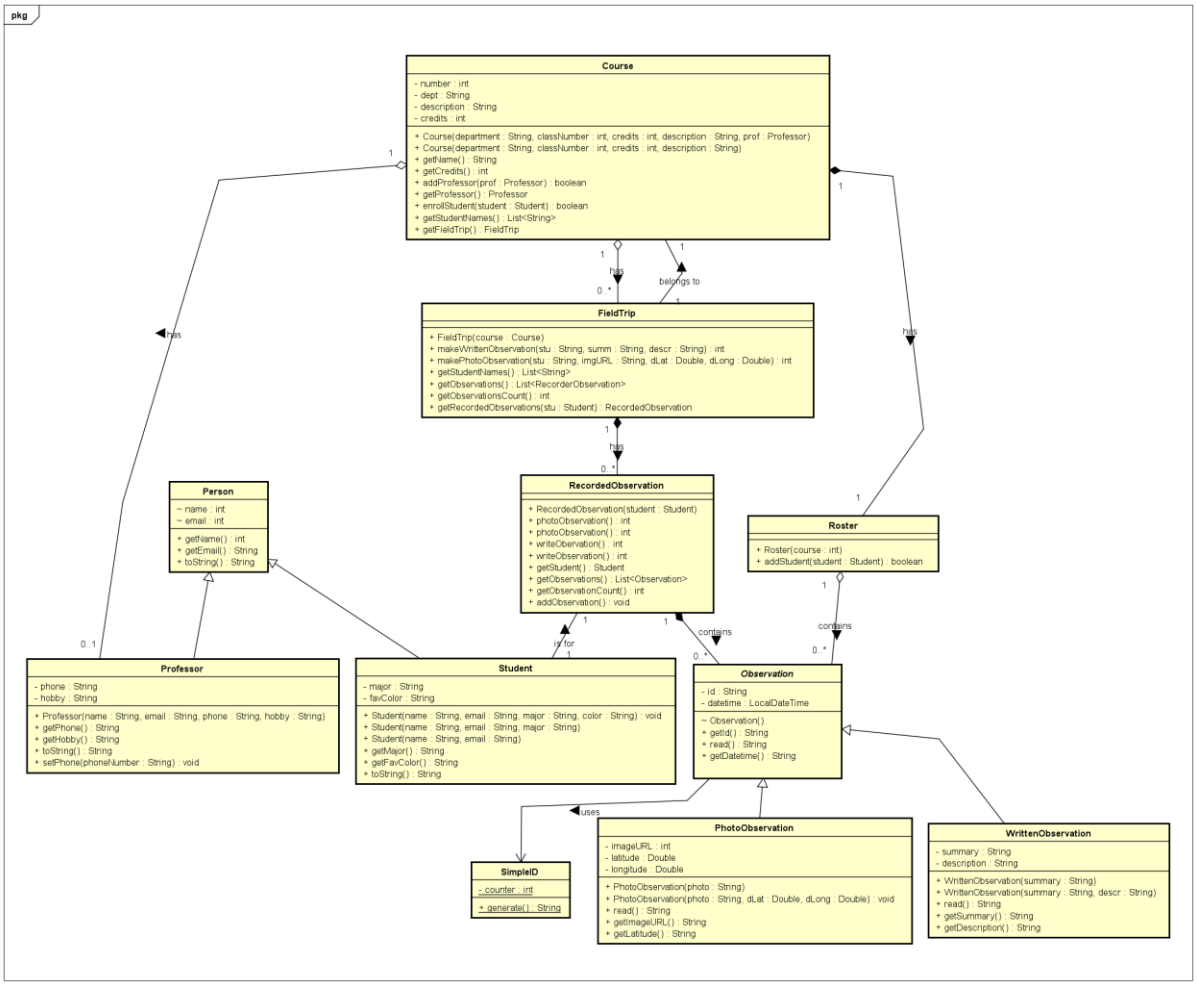
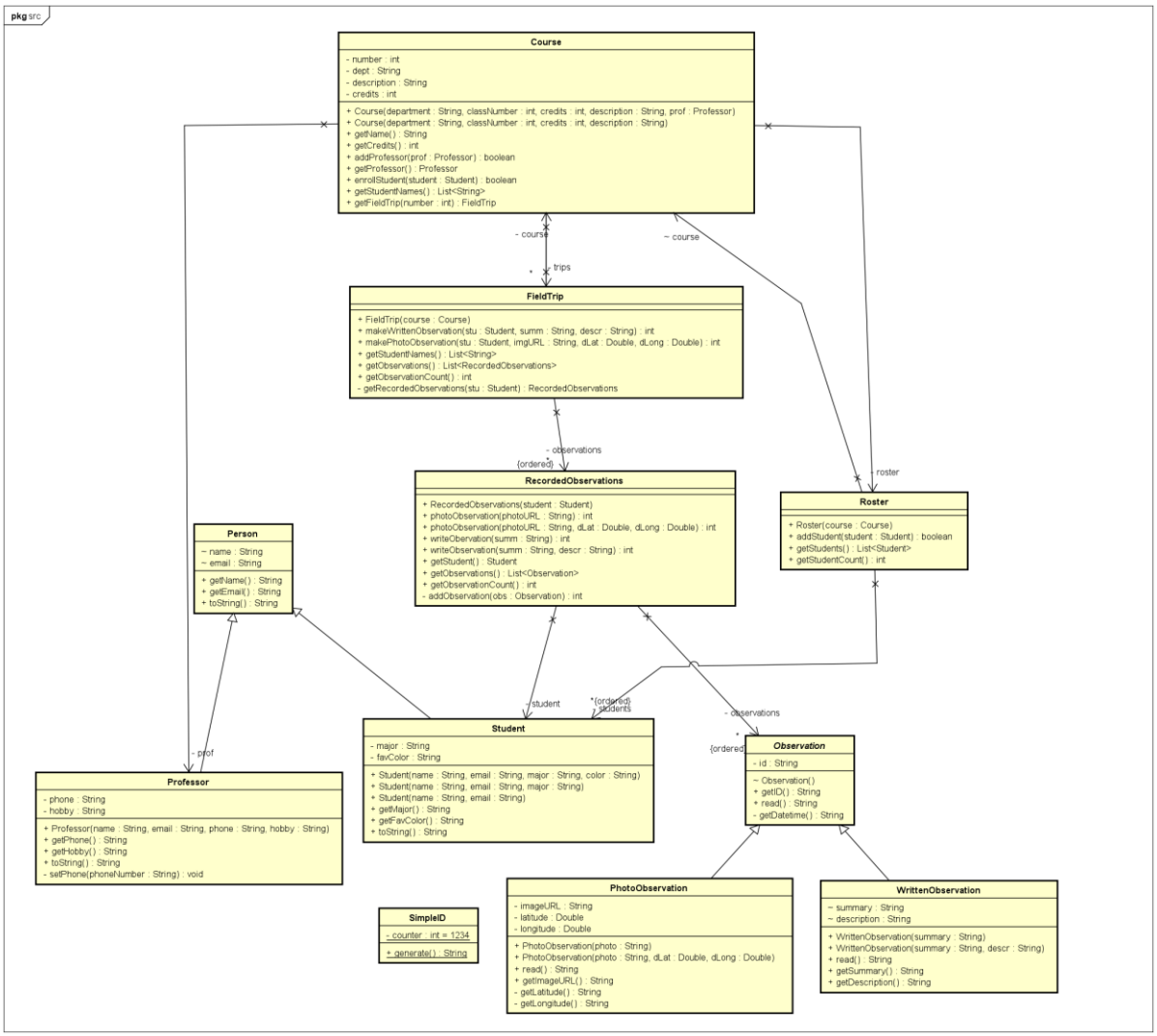
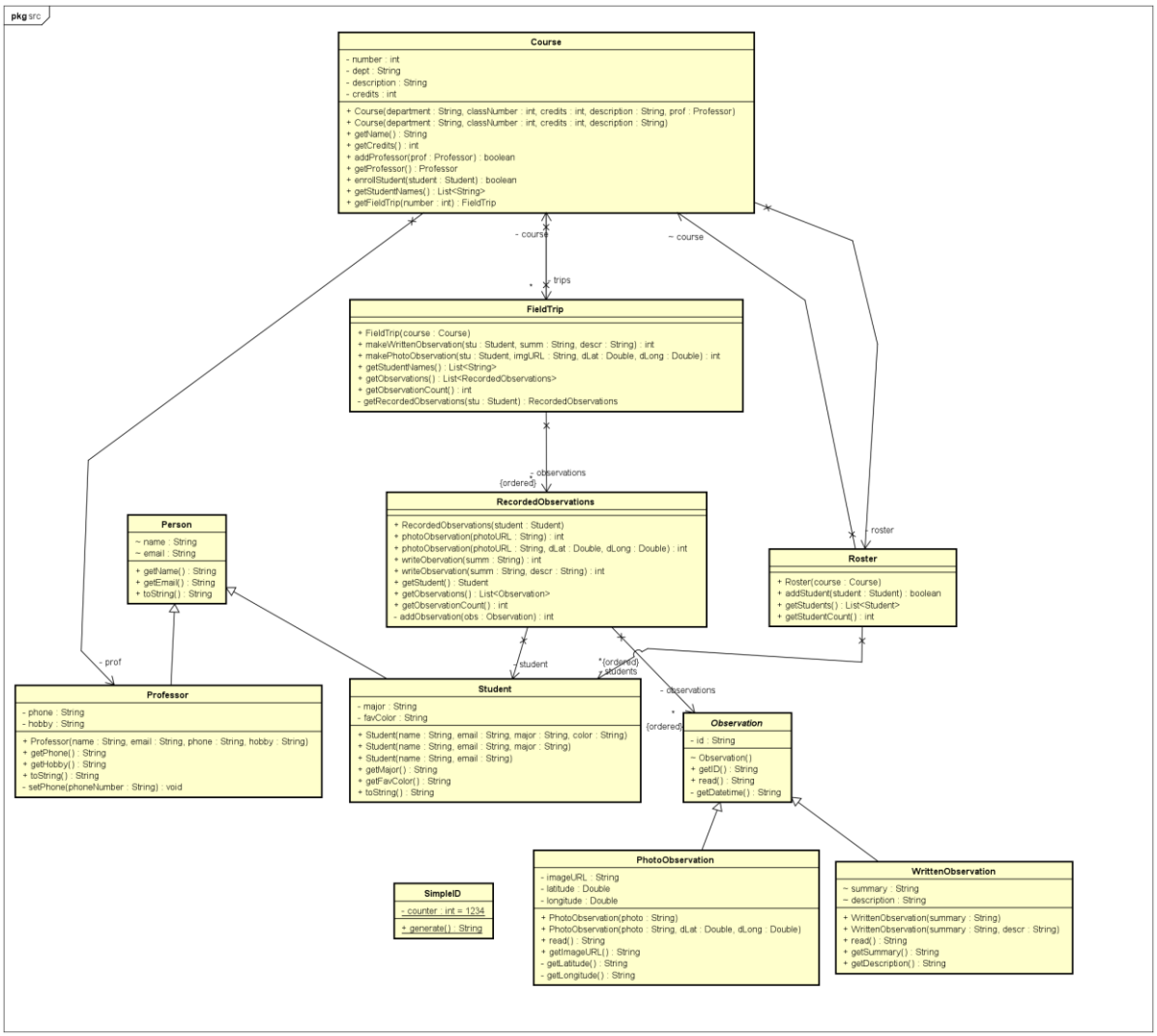
Task 1


pkg
Course
- number : int
- dept : String
- description : String
- credits : int
+ Course(department : String, classNumber : int, credits : int, description : String, prof : Professor)
+ Course(department : String, classNumber : int, credits : int, description : String)
+ getName() : String
+ getCredits() : int
+ addProfessor(prof : Professor) : boolean
+ getProfessor() : Professor
+ enrollStudent(student : Student) : boolean
+ getStudentNames() : List<String>
+ getFieldTrip() : FieldTrip
1
has
0..*
belongs to
1
has
has
FieldTrip
+ FieldTrip(course : Course)
+ makeWrittenObservation(stu : String, summ : String, descr : String) : int
+ makePhotoObservation(stu : String, imgURL : String, dLat : Double, dLong : Double) : int
+ getStudentNames() : List<String>
+ getObservations() : List<RecorderObservation>
+ getObservationsCount() : int
+ getRecordedObservations(stu : Student) : RecordedObservation
1
has
0..*
Person
~ name : int
~ email : int
+ getName() : int
+ getEmail() : String
+ toString() : String
RecordedObservation
+ RecordedObservation(student : Student)
+ photoObservation() : int
+ photoObservation() : int
+ writeObervation() : int
+ writeObservation() : int
+ getStudent() : Student
+ getObservations() : List<Observation>
+ getObservationCount() : int
+ addObservation() : void
Roster
+ Roster(course : int)
+ addStudent(student : Student) : boolean
1
contains
0..*
1
is for
1
1
contains
0..*
0..1
Professor
- phone : String
- hobby : String
+ Professor(name : String, email : String, phone : String, hobby : String)
+ getPhone() : String
+ getHobby() : String
+ toString() : String
+ setPhone(phoneNumber : String) : void
Student
- major : String
- favColor : String
+ Student(name : String, email : String, major : String, color : String) : void
+ Student(name : String, email : String, major : String)
+ Student(name : String, email : String)
+ getMajor() : String
+ getFavColor() : String
+ toString() : String
Observation
- id : String
- datetime : LocalDateTime
~ Observation()
+ getId() : String
+ read() : String
+ getDatetime() : String
uses
SimpleID
- counter : int
+ generate() : String
PhotoObservation
- imageURL : int
- latitude : Double
- longitude : Double
+ PhotoObservation(photo : String)
+ PhotoObservation(photo : String, dLat : Double, dLong : Double) : void
+ read() : String
+ getImageURL() : String
+ getLatitude() : String
WrittenObservation
- summary : String
- description : String
+ WrittenObservation(summary : String)
+ WrittenObservation(summary : String, descr : String)
+ read() : String
+ getSummary() : String
+ getDescription() : String

Task 2.1


pkg src
Course
- number : int
- dept : String
- description : String
- credits : int
+ Course(department : String, classNumber : int, credits : int, description : String, prof : Professor)
+ Course(department : String, classNumber : int, credits : int, description : String)
+ getName() : String
+ getCredits() : int
+ addProfessor(prof : Professor) : boolean
+ getProfessor() : Professor
+ enrollStudent(student : Student) : boolean
+ getStudentNames() : List<String>
+ getFieldTrip(number : int) : FieldTrip
- course
trips
~ course
FieldTrip
+ FieldTrip(course : Course)
+ makeWrittenObservation(stu : Student, summ : String, descr : String) : int
+ makePhotoObservation(stu : Student, imgURL : String, dLat : Double, dLong : Double) : int
+ getStudentNames() : List<String>
+ getObservations() : List<RecordedObservations>
+ getObservationCount() : int
- getRecordedObservations(stu : Student) : RecordedObservations
- observations
{ordered}
roster
RecordedObservations
+ RecordedObservations(student : Student)
+ photoObservation(photoURL : String) : int
+ photoObservation(photoURL : String, dLat : Double, dLong : Double) : int
+ writeObservation(summ : String) : int
+ writeObservation(summ : String, descr : String) : int
+ getStudent() : Student
+ getObservations() : List<Observation>
+ getObservationCount() : int
- addObservation(obs : Observation) : int
Roster
+ Roster(course : Course)
+ addStudent(student : Student) : boolean
+ getStudents() : List<Student>
+ getStudentCount() : int
Person
~ name : String
~ email : String
+ getName() : String
+ getEmail() : String
+ toString() : String
- student
*{ordered}
students
- observations
{ordered}
- prof
Student
- major : String
- favColor : String
+ Student(name : String, email : String, major : String, color : String)
+ Student(name : String, email : String, major : String)
+ Student(name : String, email : String)
+ getMajor() : String
+ getFavColor() : String
+ toString() : String
Observation
- id : String
~ Observation()
+ getID() : String
+ read() : String
- getDatetime() : String
Professor
- phone : String
- hobby : String
+ Professor(name : String, email : String, phone : String, hobby : String)
+ getPhone() : String
+ getHobby() : String
+ toString() : String
- setPhone(phoneNumber : String) : void
SimpleID
- counter : int = 1234
+ generate() : String
PhotoObservation
- imageURL : String
- latitude : Double
- longitude : Double
+ PhotoObservation(photo : String)
+ PhotoObservation(photo : String, dLat : Double, dLong : Double)
+ read() : String
+ getImageURL() : String
- getLatitude() : String
- getLongitude() : String
WrittenObservation
~ summary : String
~ description : String
+ WrittenObservation(summary : String)
+ WrittenObservation(summary : String, descr : String)
+ read() : String
+ getSummary() : String
+ getDescription() : String

Task 2.2


pkg src
Course
- number : int
- dept : String
- description : String
- credits : int
+ Course(department : String, classNumber : int, credits : int, description : String, prof : Professor)
+ Course(department : String, classNumber : int, credits : int, description : String)
+ getName() : String
+ getCredits() : int
+ addProfessor(prof : Professor) : boolean
+ getProfessor() : Professor
+ enrollStudent(student : Student) : boolean
+ getStudentNames() : List<String>
+ getFieldTrip(number : int) : FieldTrip
- course
~ course
trips
FieldTrip
+ FieldTrip(course : Course)
+ makeWrittenObservation(stu : Student, summ : String, descr : String) : int
+ makePhotoObservation(stu : Student, imgURL : String, dLat : Double, dLong : Double) : int
+ getStudentNames() : List<String>
+ getObservations() : List<RecordedObservations>
+ getObservationCount() : int
- getRecordedObservations(stu : Student) : RecordedObservations
observations
{ordered}
RecordedObservations
+ RecordedObservations(student : Student)
+ photoObservation(photoURL : String) : int
+ photoObservation(photoURL : String, dLat : Double, dLong : Double) : int
+ writeObservation(summ : String) : int
+ writeObservation(summ : String, descr : String) : int
+ getStudent() : Student
+ getObservations() : List<Observation>
+ getObservationCount() : int
- addObservation(obs : Observation) : int
Person
~ name : String
~ email : String
+ getName() : String
+ getEmail() : String
+ toString() : String
roster
Roster
+ Roster(course : Course)
+ addStudent(student : Student) : boolean
+ getStudents() : List<Student>
+ getStudentCount() : int
- prof
Professor
- phone : String
- hobby : String
+ Professor(name : String, email : String, phone : String, hobby : String)
+ getPhone() : String
+ getHobby() : String
+ toString() : String
- setPhone(phoneNumber : String) : void
- student
*{ordered}
students
Student
- major : String
- favColor : String
+ Student(name : String, email : String, major : String, color : String)
+ Student(name : String, email : String, major : String)
+ Student(name : String, email : String)
+ getMajor() : String
+ getFavColor() : String
+ toString() : String
- observations
*{ordered}
Observation
- id : String
~ Observation()
+ getID() : String
+ read() : String
- getDatetime() : String
SimpleID
- counter : int = 1234
+ generate() : String
PhotoObservation
- imageURL : String
- latitude : Double
- longitude : Double
+ PhotoObservation(photo : String)
+ PhotoObservation(photo : String, dLat : Double, dLong : Double)
+ read() : String
+ getImageURL() : String
- getLatitude() : String
- getLongitude() : String
WrittenObservation
~ summary : String
~ description : String
+ WrittenObservation(summary : String)
+ WrittenObservation(summary : String, descr : String)
+ read() : String
+ getSummary() : String
+ getDescription() : String

Task 2 Questions & Answers

1. All three diagrams have a different level of abstraction. Explain the main differences.

The manual diagram uses associations for relationships, focusing on conceptual structure. The first automatic diagram shows relationships as attributes, reflecting implementation details. The second automatic diagram sits between, converting some attributes to associations while retaining code-level elements.

2. Why do all three diagrams still represent the same system even though they look so different?

All preserve the essential classes, inheritance relationships, and key connections. The differences are in visualization style and detail level, not in the core system structure.

3. Why is an association basically the same as having the attribute in the class?

Associations and attributes represent the same relationship at different abstraction levels. Associations show conceptual relationships while attributes show implementation. When generating code, associations become attributes anyway.

Task 3 Question & Answer

What are the main differences, and which diagram type would be the best to export to Java right away as a template?

The main differences are in completeness and accuracy. Task 1 code lacks proper package declarations, has type inconsistencies, and incomplete relationships. Task 2 code includes proper package structure, correct types, and complete relationships.

The Task 2.2 diagram would be best to export to Java as a template because it includes complete package declarations, proper data types, consistent naming, and requires minimal modification to be functional code, striking a good balance between UML modeling and implementation readiness.