Student Registration System

1. **Create domain class diagram** (just lines and multiplicity, no arrows)
2. **Remove multiplicity**
3. **Consider a use case (ex. Register) and present it in a controller class**
4. **Add visibility -** visibility says from which class what other classes can be viewed) - use an arrow
5. **Add navigation** - connect the contr. class with the classes it should hanlde
6. **Elaborate the attributes -** add visibilty scope, declare types
7. **Add responsibilities - methods**

Attributes	Responsibilities	Collaboration
id: String name: String studentNumber: string grades: List<Grades> courses: List<Courses>	getGrades() getCourses() setCourses() getName() setName() getStudentNumber()	Course

StudentHandler

+enrollstudent: Student)
+assignToCourse(studentId: String, courseId: String)
+assignToActivity(studentId: String, activityId: String)
+remove(studentId: String)
+update(studentId: String, newStudent: Student)
gett(): Student
getAll(): List<Student>

CourseHandler

+create(course: Course)
+update(courseId: String, course: Course)
+delete(courseId: String)
+get(courseId: String)
+getAll(): List<Course>

TeacherHandler

+hire(teacher: Teacher)
+fire(teacherId: Teacher)
+assignToCourse(course: Course, teacher: Teacher)
+grade(studentId: String, courseId: String, grade: int)
+get()
+getAll()

Student

-name:String

getters, setters

Course

-name:String

getters, setters

Teacher

-name: String

ActivityHandler

+createActivity()
+updateActivity()
+deleteActivity()
+getActivity()
+getAllActivities()

Activity

-name:String

getters, setters

Grade

-grade: int

getters, setters