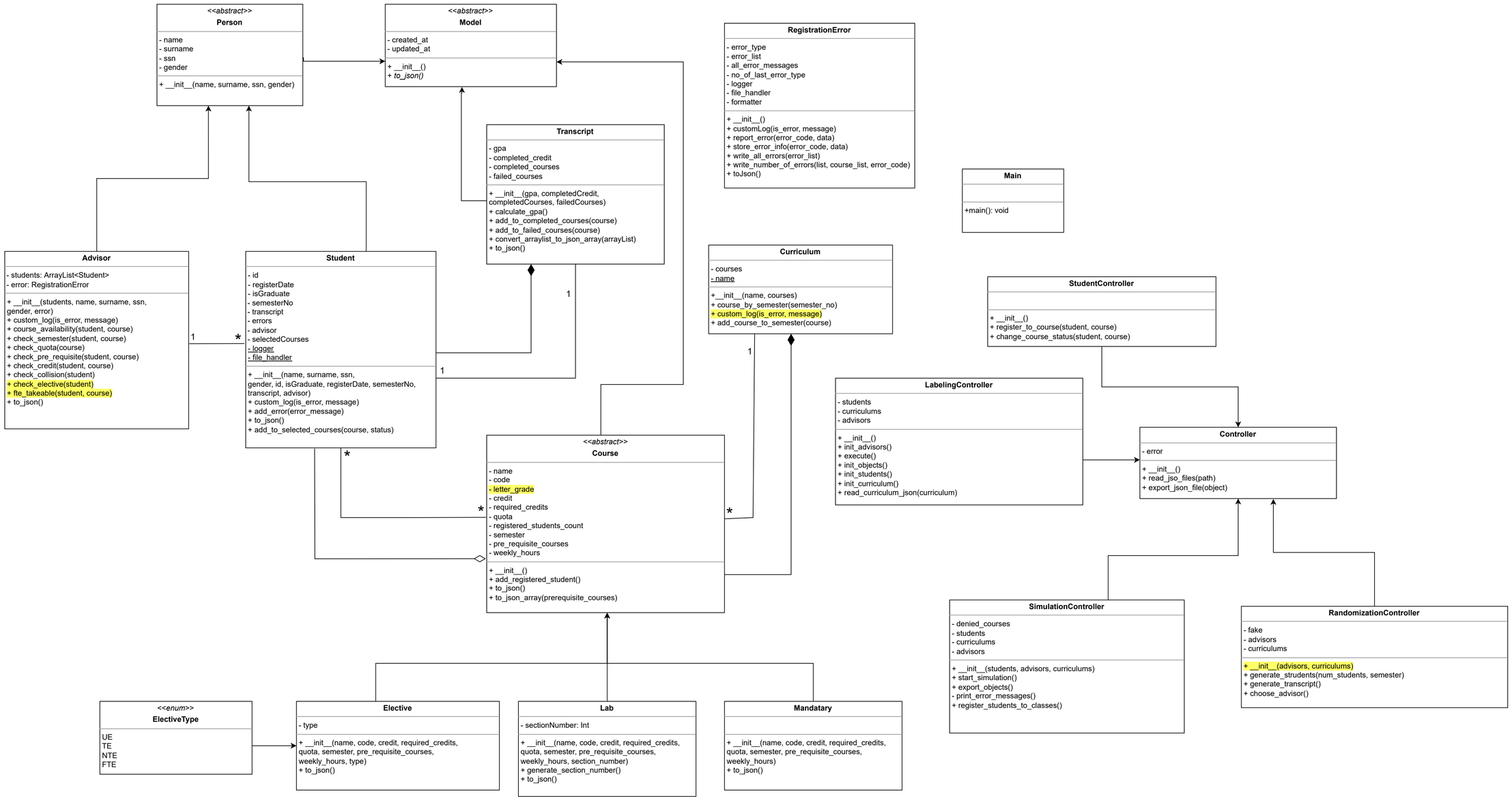
<<abstract>>
Person
- name
- surname
- ssn
- gender
+ __init__(name, surname, ssn, gender)
<<abstract>>
Model
- created_at
- updated_at
+ __init__()
+ to_json()
RegistrationError
- error_type
- error_list
- all_error_messages
- no_of_last_error_type
- logger
- file_handler
- formatter
+ __init__()
+ customLog(is_error, message)
+ report_error(error_code, data)
+ store_error_info(error_code, data)
+ write_all_errors(error_list)
+ write_number_of_errors(list, course_list, error_code)
+ toJson()
Transcript
- gpa
- completed_credit
- completed_courses
- failed_courses
+ __init__(gpa, completedCredit, completedCourses, failedCourses)
+ calculate_gpa()
+ add_to_completed_courses(course)
+ add_to_failed_courses(course)
+ convert_arraylist_to_json_array(arrayList)
+ to_json()
Main
+main(): void
Advisor
- students: ArrayList<Student>
- error: RegistrationError
+ __init__(students, name, surname, ssn, gender, error)
+ custom_log(is_error, message)
+ course_availability(student, course)
+ check_semester(student, course)
+ check_quota(course)
+ check_pre_requisite(student, course)
+ check_credit(student, course)
+ check_collision(student)
+ check_elective(student)
+ fte_takeable(student, course)
+ to_json()
Student
- id
- registerDate
- isGraduate
- semesterNo
- transcript
- errors
- advisor
- selectedCourses
- logger
- file_handler
+ __init__(name, surname, ssn, gender, id, isGraduate, registerDate, semesterNo, transcript, advisor)
+ custom_log(is_error, message)
+ add_error(error_message)
+ to_json()
+ add_to_selected_courses(course, status)
Curriculum
- courses
- name
+__init__(name, courses)
+ course_by_semester(semester_no)
+ custom_log(is_error, message)
+ add_course_to_semester(course)
StudentController
+ __init__()
+ register_to_course(student, course)
+ change_course_status(student, course)
LabelingController
- students
- curriculums
- advisors
+ __init__()
+ init_advisors()
+ execute()
+ init_objects()
+ init_students()
+ init_curriculum()
+ read_curriculum_json(curriculum)
Controller
- error
+ __init__()
+ read_jso_files(path)
+ export_json_file(object)
<<abstract>>
Course
- name
- code
- letter_grade
- credit
- required_credits
- quota
- registered_students_count
- semester
- pre_requisite_courses
- weekly_hours
+ __init__()
+ add_registered_student()
+ to_json()
+ to_json_array(prerequisite_courses)
SimulationController
- denied_courses
- students
- curriculums
- advisors
+ __init__(students, advisors, curriculums)
+ start_simulation()
+ export_objects()
- print_error_messages()
+ register_students_to_classes()
RandomizationController
- fake
- advisors
- curriculums
+ __init__(advisors, curriculums)
+ generate_students(num_students, semester)
+ generate_transcript()
+ choose_advisor()
<<enum>>
ElectiveType
UE
TE
NTE
FTE
Elective
- type
+ __init__(name, code, credit, required_credits, quota, semester, pre_requisite_courses, weekly_hours, type)
+ to_json()
Lab
- sectionNumber: Int
+ __init__(name, code, credit, required_credits, quota, semester, pre_requisite_courses, weekly_hours, section_number)
+ generate_section_number()
+ to_json()
Mandatary
+ __init__(name, code, credit, required_credits, quota, semester, pre_requisite_courses, weekly_hours)
+ to_json()
1
*
1
1
*
*
1
*
*