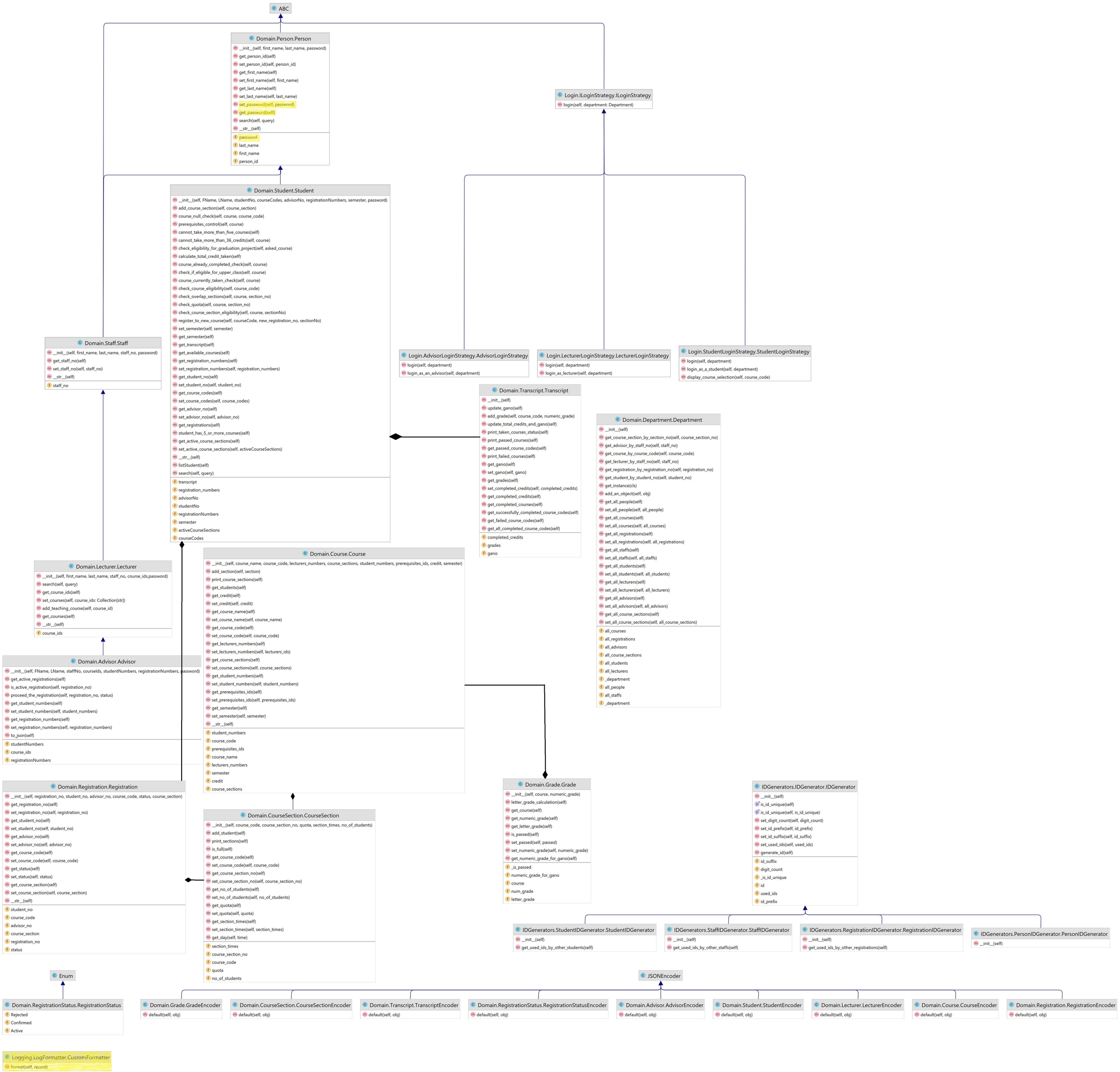
ABC
Domain.Person.Person
__init__(self, first_name, last_name, password)
get_person_id(self)
set_person_id(self, person_id)
get_first_name(self)
set_first_name(self, first_name)
get_last_name(self)
set_last_name(self, last_name)
set_password(self, password)
get_password(self)
search(self, query)
__str__(self)
password
last_name
first_name
person_id
Login.ILoginStrategy.ILoginStrategy
login(self, department: Department)
Domain.Student.Student
__init__(self, FName, LName, studentNo, courseCodes, advisorNo, registrationNumbers, semester, password)
add_course_section(self, course_section)
course_null_check(self, course, course_code)
prerequisites_control(self, course)
cannot_take_more_than_five_courses(self)
cannot_take_more_than_36_credits(self, course)
check_eligibility_for_graduation_project(self, asked_course)
calculate_total_credit_taken(self)
course_already_completed_check(self, course)
check_if_eligible_for_upper_class(self, course)
course_currently_taken_check(self, course)
check_course_eligibility(self, course_code)
check_overlap_sections(self, course, section_no)
check_quota(self, course, section_no)
check_course_section_eligibility(self, course, sectionNo)
register_to_new_course(self, courseCode, new_registration_no, sectionNo)
set_semester(self, semester)
get_semester(self)
get_transcript(self)
get_available_courses(self)
get_registration_numbers(self)
set_registration_numbers(self, registration_numbers)
get_student_no(self)
set_student_no(self, student_no)
get_course_codes(self)
set_course_codes(self, course_codes)
get_advisor_no(self)
set_advisor_no(self, advisor_no)
get_registrations(self)
student_has_5_or_more_courses(self)
get_active_course_sections(self)
set_active_course_sections(self, activeCourseSections)
__str__(self)
listStudent(self)
search(self, query)
transcript
registration_numbers
advisorNo
studentNo
registrationNumbers
semester
activeCourseSections
courseCodes
Domain.Staff.Staff
__init__(self, first_name, last_name, staff_no, password)
get_staff_no(self)
set_staff_no(self, staff_no)
__str__(self)
staff_no
Login.AdvisorLoginStrategy.AdvisorLoginStrategy
login(self, department)
login_as_an_advisor(self, department)
Login.LecturerLoginStrategy.LecturerLoginStrategy
login(self, department)
login_as_lecturer(self, department)
Login.StudentLoginStrategy.StudentLoginStrategy
login(self, department)
login_as_a_student(self, department)
display_course_selection(self, course_code)
Domain.Transcript.Transcript
__init__(self)
update_gano(self)
add_grade(self, course_code, numeric_grade)
update_total_credits_and_gano(self)
print_taken_courses_status(self)
print_passed_courses(self)
get_passed_course_codes(self)
print_failed_courses(self)
get_gano(self)
set_gano(self, gano)
get_grades(self)
set_completed_credits(self, completed_credits)
get_completed_credits(self)
get_completed_courses(self)
get_successfully_completed_course_codes(self)
get_failed_course_codes(self)
get_all_completed_course_codes(self)
completed_credits
grades
gano
Domain.Department.Department
__init__(self)
get_course_section_by_section_no(self, course_section_no)
get_advisor_by_staff_no(self, staff_no)
get_course_by_course_code(self, course_code)
get_lecturer_by_staff_no(self, staff_no)
get_registration_by_registration_no(self, registration_no)
get_student_by_student_no(self, student_no)
get_instance(cls)
add_an_object(self, obj)
get_all_people(self)
set_all_people(self, all_people)
get_all_courses(self)
set_all_courses(self, all_courses)
get_all_registrations(self)
set_all_registrations(self, all_registrations)
get_all_staffs(self)
set_all_staffs(self, all_staffs)
get_all_students(self)
set_all_students(self, all_students)
get_all_lecturers(self)
set_all_lecturers(self, all_lecturers)
get_all_advisors(self)
set_all_advisors(self, all_advisors)
get_all_course_sections(self)
set_all_course_sections(self, all_course_sections)
all_courses
all_registrations
all_advisors
all_course_sections
all_students
all_lecturers
_department
all_people
all_staffs
_department
Domain.Lecturer.Lecturer
__init__(self, first_name, last_name, staff_no, course_ids,password)
search(self, query)
get_course_ids(self)
set_courses(self, course_ids: Collection[str])
add_teaching_course(self, course_id)
get_courses(self)
__str__(self)
course_ids
Domain.Course.Course
__init__(self, course_name, course_code, lecturers_numbers, course_sections, student_numbers, prerequisites_ids, credit, semester)
add_section(self, section)
print_course_sections(self)
get_students(self)
get_credit(self)
set_credit(self, credit)
get_course_name(self)
set_course_name(self, course_name)
get_course_code(self)
set_course_code(self, course_code)
get_lecturers_numbers(self)
set_lecturers_numbers(self, lecturers_ids)
get_course_sections(self)
set_course_sections(self, course_sections)
get_student_numbers(self)
set_student_numbers(self, student_numbers)
get_prerequisites_ids(self)
set_prerequisites_ids(self, prerequisites_ids)
get_semester(self)
set_semester(self, semester)
__str__(self)
student_numbers
course_code
prerequisites_ids
course_name
lecturers_numbers
semester
credit
course_sections
Domain.Advisor.Advisor
__init__(self, FName, LName, staffNo, courseIds, studentNumbers, registrationNumbers, password)
get_active_registrations(self)
is_active_registration(self, registration_no)
proceed_the_registration(self, registration_no, status)
get_student_numbers(self)
set_student_numbers(self, student_numbers)
get_registration_numbers(self)
set_registration_numbers(self, registration_numbers)
to_json(self)
studentNumbers
course_ids
registrationNumbers
Domain.Registration.Registration
__init__(self, registration_no, student_no, advisor_no, course_code, status, course_section)
get_registration_no(self)
set_registration_no(self, registration_no)
get_student_no(self)
set_student_no(self, student_no)
get_advisor_no(self)
set_advisor_no(self, advisor_no)
get_course_code(self)
set_course_code(self, course_code)
get_status(self)
set_status(self, status)
get_course_section(self)
set_course_section(self, course_section)
__str__(self)
student_no
course_code
advisor_no
course_section
registration_no
status
Domain.CourseSection.CourseSection
__init__(self, course_code, course_section_no, quota, section_times, no_of_students)
add_student(self)
print_sections(self)
is_full(self)
get_course_code(self)
set_course_code(self, course_code)
get_course_section_no(self)
set_course_section_no(self, course_section_no)
get_no_of_students(self)
set_no_of_students(self, no_of_students)
get_quota(self)
set_quota(self, quota)
get_section_times(self)
set_section_times(self, section_times)
get_day(self, time)
section_times
course_section_no
course_code
quota
no_of_students
Domain.Grade.Grade
__init__(self, course, numeric_grade)
letter_grade_calculation(self)
get_course(self)
get_numeric_grade(self)
get_letter_grade(self)
is_passed(self)
set_passed(self, passed)
set_numeric_grade(self, numeric_grade)
get_numeric_grade_for_gano(self)
_is_passed
numeric_grade_for_gano
course
num_grade
letter_grade
IDGenerators.IDGenerator.IDGenerator
__init__(self)
is_id_unique(self)
is_id_unique(self, is_id_unique)
set_digit_count(self, digit_count)
set_id_prefix(self, id_prefix)
set_id_suffix(self, id_suffix)
set_used_ids(self, used_ids)
generate_id(self)
id_suffix
digit_count
_is_id_unique
id
used_ids
id_prefix
IDGenerators.StudentIDGenerator.StudentIDGenerator
__init__(self)
get_used_ids_by_other_students(self)
IDGenerators.StaffIDGenerator.StaffIDGenerator
__init__(self)
get_used_ids_by_other_staffs(self)
IDGenerators.RegistrationIDGenerator.RegistrationIDGenerator
__init__(self)
get_used_ids_by_other_registrations(self)
IDGenerators.PersonIDGenerator.PersonIDGenerator
__init__(self)
Enum
Domain.RegistrationStatus.RegistrationStatus
Rejected
Confirmed
Active
JSONEncoder
Domain.Grade.GradeEncoder
default(self, obj)
Domain.CourseSection.CourseSectionEncoder
default(self, obj)
Domain.Transcript.TranscriptEncoder
default(self, obj)
Domain.RegistrationStatus.RegistrationStatusEncoder
default(self, obj)
Domain.Advisor.AdvisorEncoder
default(self, obj)
Domain.Student.StudentEncoder
default(self, obj)
Domain.Lecturer.LecturerEncoder
default(self, obj)
Domain.Course.CourseEncoder
default(self, obj)
Domain.Registration.RegistrationEncoder
default(self, obj)
Logging.LogFormatter.CustomFormatter
format(self, record)