	CoureRegistrationSimulation							
FileManager1	-students: list							
	-courses: list							
Fread_student(self, path): list	-lecturers: list							
-read_course(self, path): list	-advisors: list							
+read_lecturers(self, path): list +read_params(self, path): list	-credit_limit: int							
+write_to_file(student, file_path): void	-enrollment_requests: list							
+write_to_file_with_array(student, file_path): void	-errors: list			Transcript				
	-prob_to_pass_class: float		-course_grades: ?					
	-prob_to: float	Schedule	-gpa: float					
	-names: list	-course_day_hour:	-taken_credit: int					
		+init(self, course_day_hour)						
	-surnames: list	+set_course_day_hour(self, course_day_hour): void	-taken_courses: list					
	-letterGrades: list	+is_collided(self, schedule, schedule1): boolean	-not_taken_courses: list					
	+init(self, students, courses, lecturers, advisors, credit_limit)	+append_schedule(self, schedule, schedule1): object	+init(self, course_grade:	s, student, gpa, taken_credit, taken_cour	ses, not_taken_courses):			
	+star_simulation(self): void	+_str_(self): void	+init(self)					
	+readParameters(): void		+add_taken_course(self, cou					
	+create_courses(self): void +create_students(self): void		+add_taken_course_one(self) +add_credit(self, added_credit					
	+create_lecturer(self): void		+calculate_gpa(self): float	,				
	+create_advisors(self): void		+random_course_grade(self,	courses, prob): ?				
	+match_student_advisor(self): void		+_str_(self): str		1			
	+get_random_course(list): list	1						
	+pass_or_fail(self): boolean							
	+setTranscriptAfter(EnrollmentRequest enrollmentRequest): void +create_random_student(self, size): list			*				
	+print_arraylist(arr): void							
	+check_system_requirements(self, enrollment_request): void							
	+apply_course(self, student): list					*	*	
			1	1	Elective Course		CompulsaryCourse	
			Student		-quota: int		-course_code: str	
	RegistrationSystem		-student_name: str		-schedule: Schedule		-course_name: str	
	+init(self)		-student_surname: str		-electiveType: str		-credit: int	Lecturer
	+check_course_is_taken_before(self, enrollment_request): void		-transcript_before: Transcript		+init(course_code: str, course_name: str,		-prerequisites:list	-lecturer_name: str
	+check_prerequisites(self, enrollment_request): void		-transcript_after: Transcript		credit: int, prerequisites: list, schedule: Schedule, semester: Semester, lecturer: Lecturer, quality	quota: int, elective_type: str)	-lecturer: Lecturer	-lecturer_surname: str
	+is_opened_this_term(self, enrollment_request, term):void		-schedule: Schedule				-course_section: CourseSection	-schedule: Schedule
	is_opened_unio_com(con, ornonmont_request, term).void		-advisor: Advisor	Г	*		-course_students: list	+init(self, lecturer_name: str, lecturer_surname: str, schedule: Schedule)
			-enrolled_year: str				-opened_term: str	+str(): str
			+init(self, student_name: str, student_sur	name: etr. transcript, before			+init(self, course_code: str, course_name: str, credit: int, prerequisites: list, lecturer: Lecturer, course_section: CourseSection	
			transcriptAfter: Transcript, schedule: Schedule	e, advisor: Advisor,			course_students: list, givenSemester: str)	
			enrolledYear: str) + init (self student name; str student sur	name: str. transcript, before			+get_random_section(self): Schedule +str(): str	* 1
			+init(self, student_name: str, student_sur schedule: Schedule, advisor: Advisor, enrolled	dYear: str)			1_5tt_(). 5tt	
			+init(self, student_id: str, student_name: enrolledYear: str)	str, student_surname: str,				
						<u>1</u>		
			+create_studentID(self): void	*				$egin{array}{cccccccccccccccccccccccccccccccccccc$
			+str(self): str	*[
								Evtends
								Exicites
								Extends
								Extends
					* EnrollmentRequest			Advisor
				-courses: list	* EnrollmentRequest			Advisor -students: Arraylist <student></student>
				-courses: list	* EnrollmentRequest *			-students: Arraylist <student></student>
				-student: Student	*,			-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list)</student>
				-student: Student -schedule: Schedule	*,		Course Section	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str	*,	-course_code: str		-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list)</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int	*,	-course_code: str -section_count: int		-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str	*.		CourseSection	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	*,	-section_count: int	CourseSection	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str	*.	-section_count: int -schedule: HashMap <str -quota:="" int<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	*.	-section_count: int -schedule: HashMap <str +init(self,="" -quota:="" course_co<="" int="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	*.	-section_count: int -schedule: HashMap <str -quota:="" int<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	*.	-section_count: int -schedule: HashMap <str +init(self,="" -quota:="" course_counts="" hashmap,="" int="" quote="" schedule:="" str():="" string<="" td="" the=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	CompulsoryCourse> courses, Student student)	-section_count: int -schedule: HashMap <str +init(self,="" +str():="" -quota:="" course_coschedule:="" hashmap,="" int="" quo="" string<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	CompulsoryCourse> courses, Student student)	-section_count: int -schedule: HashMap <str +init(self,="" +str():="" -quota:="" course_coschedule:="" hashmap,="" int="" quo="" string<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	CompulsoryCourse> courses, Student student)	-section_count: int -schedule: HashMap <str +init(self,="" +str():="" -quota:="" course_coschedule:="" hashmap,="" int="" quo="" string<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	CompulsoryCourse> courses, Student student)	-section_count: int -schedule: HashMap <str +init(self,="" +str():="" -quota:="" course_coschedule:="" hashmap,="" int="" quo="" string<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	CompulsoryCourse> courses, Student student)	-section_count: int -schedule: HashMap <str +init(self,="" +str():="" -quota:="" course_coschedule:="" hashmap,="" int="" quo="" string<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	CompulsoryCourse> courses, Student student)	-section_count: int -schedule: HashMap <str +init(self,="" +str():="" -quota:="" course_coschedule:="" hashmap,="" int="" quo="" string<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	CompulsoryCourse> courses, Student student)	-section_count: int -schedule: HashMap <str +init(self,="" +str():="" -quota:="" course_coschedule:="" hashmap,="" int="" quo="" string<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	CompulsoryCourse> courses, Student student)	-section_count: int -schedule: HashMap <str +init(self,="" +str():="" -quota:="" course_coschedule:="" hashmap,="" int="" quo="" string<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): voir</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): voir</student>
				-student: Student -schedule: Schedule -result: str -creditLimit: int -error: str +_init(ArrayList<	CompulsoryCourse> courses, Student student)	-section_count: int -schedule: HashMap <str +init(self,="" +str():="" -quota:="" course_coschedule:="" hashmap,="" int="" quo="" string<="" td=""><td>CourseSection ing, Schedule></td><td>-students: Arraylist<student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student></td></str>	CourseSection ing, Schedule>	-students: Arraylist <student> +init(lecturer_name: str, lecturer_surname: str, schedule: Schedule students: list) +check_schedule_collison(enrollmentRequest: EnrollmentRequest): void</student>