1 Problem

A student management system stores data about students. There are two kinds of university students: resident students and non-resident students. Both kinds of students have a name and a list of registered courses. Both kinds of students are restricted to register for no more than 10 courses. When calculating the tuition for a student, a base amount is first determined from the list of courses they are currently registered (each course has an associated fee). For a non-resident student, there is a discount rate applied to the base amount to waive the fee for on-campus accommodation. For a resident student, there is a premium rate applied to the base amount to account for the fee for on-campus accommodation and meals.

Task: Design classes that satisfy the above problem statement.

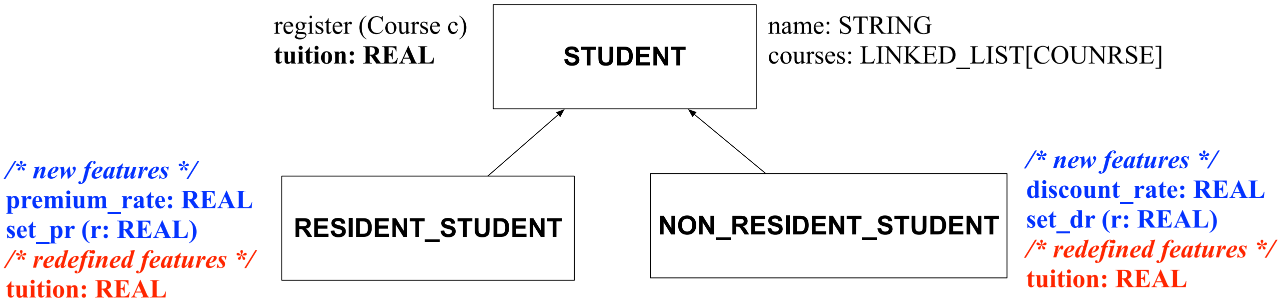
Requirements: At runtime:

- A student management system stores a collection of (all kinds of) students.
- Each type of student is able to register a course and calculate their tuition fee.

Design Principles to Consider:

- Cohesion
- Single Choice Principle

2 Static Type Declares Expectation



```
s1,s2,s3: STUDENT ; rs: RESIDENT_STUDENT ; nrs : NON_RESIDENT_STUDENT
create {STUDENT} s1.make ("S1")
create {RESIDENT_STUDENT} s2.make ("S2")
create {NON_RESIDENT_STUDENT} s3.make ("S3")
create {RESIDENT_STUDENT} rs.make ("RS")
create {NON_RESIDENT_STUDENT} nrs.make ("NRS")
```

	name	courses	register	tuition	premium_rate	set_pr	discount_rate	set_dr
s.								
rs.								
nrs.								