Person

Staff

Professor

Student

0..1

Room

Section

Course

Exam

1

*

*

*

1

*

1

*

1

*

1

1

1

1

1

1

*

1

composition: tail is apart of head,
where head owns its own tail
in memory

Department

Controller

Window

Window

Window

Window

Window
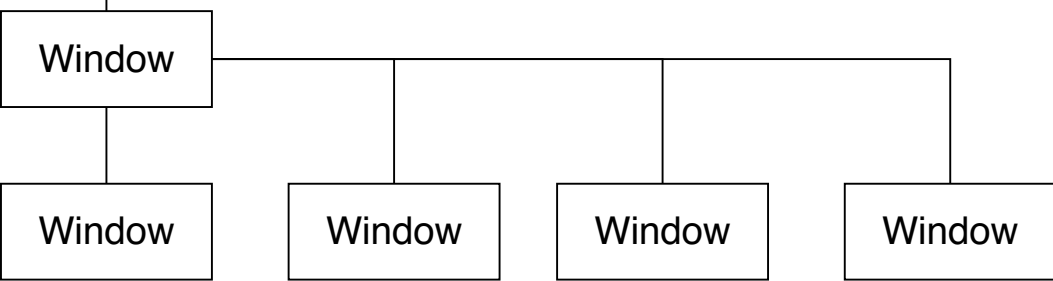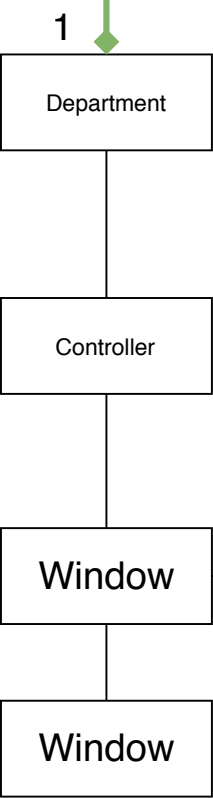
Person

name: String

email: String

id: String

+ CMP: Comparator<Person>

+ Person(List<String>)

+ getName(): String

+ getEmail(): String

+ getID(): String

+ compareTo(Person): int

Person is an
abstract class

Student

- courses: List<String>

+ Student(List<String>)

+ getCourses(): List<String>

courses: List<String> should
have the CAT#s of courses
the student is taking

Staff

- role: String

+ Staff(List<String>)

+ getRole(): String

Professor

- courses: List<String>

+ Professor(List<String>)

+ getCourses(): List<String>

courses: List<String> should
have the CAT#s of courses
the professor teaches

Exam

- date: Date

- room: Room

+ Exam(Date, Room)

+ getDate(): Date

+ getRoom(): Room

Room

- roomNum: int

+ Room(int)

+ getRoomNum(): int

Section

- letter: String

- prof: Professor

- catNum: String

+ Section(String, Professor, String)

+ getLetter(): String

+ getProfessor(): Professor

+ getCatNum(): String

Course

+ name: String

+ desc: String

+ section: Section

+ exam: Exam

+ Course(String, Section, String, Exam)

+ getName(): String

+ getDesc(): String

+ getSection(): Section

+ getExam(): Exam

Department

+ field: type

+ method(type): type