Devon James May 8, 2019

CSI 405 Prof. Umang

# ***Final Project – UML Diagrams***

*Topic:* University Model

My project aims to model a standard university. Each person within the university is of one type of person: Student or Employee. An employee can be one of two things (at least in my code): Professor or Advisor. Each employee has a salary that can be calculated and raised if necessary. Courses were also made and has a Professor assigned to every course. I've also represented standard buildings: Housing Complex, Library and Lecture Building.

The University contains a list of all employees that work within the University. Each Student contains a list of the courses they are currently taking. Each Course contains a list of students who are enrolled in the course. Each Building contains a list of the people who are currently in the building.

The Salary interface was used to hold functions that would be used by every type of Employee. The PrintArray interface was used to hold a function that prints an array of any type (generic). The Subject and ClassType enums were used to store all possible values for their respective types (to avoid error). The Building class was made abstract since you wouldn't need to create a general Building.

UML Diagrams are listed below:

| <<interface>> PrintArray<T> |
|---|
| |
| + printArray(): void |

```
<<interface>>
Salary
-----------------------------
-----------------------------
+ calculateSalary(): double
+ raiseSalary(): void
```

```
<<enumeration>>
Subject
-----------------------------
MATH
ENGLISH
SCIENCE
-----------------------------
```

```
<<enumeration>>
ClassType
-----------------------------
LECTURE
HYBRID
FULLY_ONLINE
-----------------------------
```

```
University
-----------------------------
- university: University
- universityName: String
- yearFounded: int
- maxStudents: int
- universityAddress: String
- employeeList: ArrayList<Employee>
-----------------------------
- University()
```

| |
|---|
| + getUniversity(): University<br>+ getUniversityName(): String<br>+ setUniversityName(newName: String): void<br>+ getYearFounded(): int<br>+ setYearFounded(newYear: int): void<br>+ getMaxStudents(): int<br>+ setMaxStudents(newMax: int): void<br>+ getUniversityAddress(): String<br>+ setUniversityAddress(newAddress: String): void<br>+ getEmployeeList(): ArrayList<Employee><br>+ toString(): String<br>+ addEmployee(e: Employee): void<br>+ deleteEmployee(e: Employee): void<br>+ printArray(): void |

| Person |
|---|
| - firstName: String<br>- lastName: String<br>- age: int |
| + Person()<br>+ Person(fName: String, lName: String, age: int)<br>+ getFirstName(): String<br>+ setFirstName(newFName: String): void<br>+ getLastName(): String<br>+ setLastName(newLName: String): void<br>+ getAge(): int<br>+ setAge(newAge: int): void<br>+ toString(): String<br>+ goToBuilding(b: Building): void<br>+ leaveBuilding(b: Building): void |

| Student |
|---|
| - studentID: int |
| - creditsTaken: double |
| - maxCoursesPerSemester: int |
| - currentCourses: ArrayList<Course> |
| + Student() |
| + Student(fName: String, lName: String, age: int, sID: int, credits: double) |
| + getStudentID(): int |
| + setStudentID(newID: int): void |
| + getCreditsTaken(): double |
| + setCreditsTaken(newCreds: double): void |
| + getCurrentCourses(): ArrayList<Course> |
| + toString(): String |
| + addCourse(c: Course): void |
| + deleteCourse(c: Course): void |
| + printArray(): void |

| Employee |
|---|
| - employeeID: int |
| - hoursPerWeek: double |
| # salaryPerYear: double |
| - baseRate: double |
| + Employee() |
| + Employee(fName: String, lName: String, age: int, eID: int, hours: double) |
| + getEmployeeID(): int |
| + setEmployeeID(newID: int): void |
| + getHoursPerWeek(): double |
| + setHoursPerWeek(newHours: double): void |
| + getSalaryPerYear(): double |
| + setSalaryPerYear(newSalary: double): void |

| |
|---|
| + calculateSalary(): double |
| + raiseSalary(): void |
| + toString(): String |

| Professor |
|---|
| - professorRate: double |
| - raisePercentage: double |
| + Professor() |
| + Professor(fName: String, lName: String, age: int, eID: int, hours: double) |
| + calculateSalary(): double |
| + raiseSalary(): void |
| + toString(): String |

| Advisor |
|---|
| - advisorRate: double |
| - raisePercentage: double |
| + Advisor() |
| + Advisor(fName: String, lName: String, age: int, eID: int, hours: double) |
| + calculateSalary(): double |
| + raiseSalary(): void |
| + toString(): String |

| Course |
|---|
| - courseName: String |
| - professor: Professor |
| - courseSubject: Subject |
| - classType: ClassType |
| - maxStudents: int |
| - studentsEnrolled: ArrayList<Student> |

+ Course(name: String, p: Professor, s: Subject, c: ClassType, max: int)

+ getCourseName(): String

+ setCourseName(newName: String): void

+ getProfessor(): Professor

+ setProfessor(newProfesor: Professor): void

+ getSubject(): Subject

+ setSubject(newSubject: Subject): void

+ getMaxStudents(): int

+ setMaxStudents(newMax: int): void

+ getStudentsEnrolled(): ArrayList<Student>

+ toString(): String

+ addStudent(s: Student): void

+ deleteStudent(s: Student): void

+ printArray(): void

*Building*

- buildingName: String
- yearBuilt: int
- maxOccupancy: int
- peopleInBuilding: ArrayList<Person>

+ Building()

+ Building(name: String, year: int, max: int)

+ getBuildingName(): String

+ setBuildingName(newName: String): void

+ getYearBuilt(): int

+ setYearBuilt(newYear: int): void

+ getMaxOccupancy(): int

+ setMaxOccupancy(newMax: int): void

+ getMaxStudents(): int

+ setMaxStudents(newMax: int): void

+ getPeopleInBuilding(): ArrayList<Person>

+ toString(): String

+ addPerson(p: Person): void

+ deletePerson(p: Person): void

+ printArray(): void

| HousingComplex |
|---|
| |
| + HousingComplex()<br>+ HousingComplex(name: String, year: int, max: int)<br>+ toString(): String |

| LectureBuilding |
|---|
| |
| + LectureBuilding()<br>+ LectureBuilding(name: String, year: int, max: int)<br>+ toString(): String |

| Library |
|---|
| |
| + Library()<br>+ Library(name: String, year: int, max: int)<br>+ toString(): String |

| UniversityDriver |
|---|
| |
| + main(args: String[] ): void |