Many-to-One Deferred Acceptance Algorithm

Student-Project Matching Program

Model Assembly

CS3821: ManyToOne
Custom, many-to-one implementation of deferred acceptance algorithm matching students to projects.

Attributes
- matching: HashMap<Project, ArrayList<Student>>
- studentMatched: HashMap<Student, Boolean>
- requirements: HashMap<Prerequisites, Integer>
- rejected: ArrayList<Student>
- fullIteration: ArrayList<String>
- iterations: int

Responsibilities
- propose(s: Student, p: Project): boolean
- compute(): void
- tables(students: ArrayList<Student>, projects: ArrayList<Project>): void
<<constructor>> +ManyToOne(students: ArrayList<Student>, projects: ArrayList<Project>)
+ getMatchings(): HashMap<Project, ArrayList<Student>>
+ getRejected(): ArrayList<Student>
+ getDetails(): ArrayList<String>

CS3821: Student
Student class, for constructing new students with specific attributes.

Attributes
- studentID: int
- modulesAndGrades: HashMap<Prerequisites, Integer>
- preferenceList: Project[]

Responsibilities
- populate(): void
<<constructor>> + Student(id: int, pL: Project[])
<<constructor>> + Student(id: int, pL: Project[], g: int[])
+ setID(studentID: int): void
      {Exception == IllegalArgument}
+ getID(): int
+ setGrade(module: Prerequisite, grade: int): void
      {Exception == IllegalArgument}
+ getGrade(module: Prerequisite): int
+ setAllGrades(newGrades: int): void
      {Exception==IllegalArgument}
+ getAllGrades(): HashMap<Prerequisites, Integer>
+ getAverage: Int
+ setPreferences(preferenceList: Project[]): void
+ getPreferences: Project[]

CS3821: Project
Project class, for constructing new projects with specific attributes.

Attributes
- maxCapacity: int
- projectCode: String
- requirements: HashMap<Prerequisites, Integer>

Responsibilities
<<constructor>> + Project(mC: int, pC: String)
<<constructor>> + Project(mC: int, pC: String, requirements: HashMap<Prerequisites, Integer>)
+ setCapacity(maxCapacity: int): void
      {Exception == IllegalArgument}
+ getCapacity(): int
+ setCode(projectCode: String): void
+ getCode: String
+ setRequirement(r: HashMap<Preference, Integer>): void
      {Exception == IllegalArgument}
+ getRequirement(): HashMap<Prerequisites, Integer>

<<Enumeration>>
CS3821: Prerequisites
Enumerated class to maintain all prerequisite (core) modules taken by Student and weighted by Project.

SOFTWARE_DESIGN
SOFTWARE_DEVELOPMENT
INTERNET_SERVICES
NETWORK_SECURITY
ALGORITHMS
DATA_STRUCTURES
ARTIFICIAL_INTELLIGENCE
MACHINE_LEARNING

View Assembly

<<View>>
CS3821: StudentToProjectView
A graphical frame with multiple panels.

Heading Panel:
1. Information on topic
2. Information on functionality

Student Panel:
1. Input fields for student information and grades
2. Student table, showing existing students in any instance of program.
3. Add button, to add new student to front-end and back-end (with input information)
4. Delete button, to delete selected student in table from front-end and back-end.
5. Update button, to update selected student in table with new input information.

Project Panel:
1. Input fields for Project information and weightings
2. Project table, showing existing projects in any instance of program.
3. Add button, to add new project to front-end and back-end (with input information)
4. Delete button, to delete selected project in table from front-end and back-end.
5. Update button, to update selected project in table with new input information.

Information Panel:
1. General label, for general feedback while interacting with program
2. Specific label, for specific (detailed) feedback while interacting with program
3. Compute button, to run many-to-one deferred acceptance algorithm with existing students and projects.
4. Results table, showing matching results from algorithm

<<Interface>>
CS3821: StudentToProject
Program utilising many-to-one implementation of deferred acceptance to match students to projects.

Responsibilities:
- populateHashMap(): void
- compute(): void
- clearResultsTable(): void
- projectSelected(): void
- projectAdd(pName: String, pCapacity,: int, pCode: String, weightings: HashMap<Prerequisites, Integer>): String[][]
- projectDelete(arrayPos: int): String[][]
- projectUpdate(s: int, pN: String, pCap: int, pCode: String, weightings: HashMap<Prerequisites, Integer>): String[][]
- studentSelected(): void
- studentAdd(sName: String, sID: int, grade: int[], preferences: String): String[][]
- studentDelete(arrayPos: int): String[][]
- studentUpdate(s: int, sName: String, sID: int, grades: int[], preferences: String): String[][]
<<constructor>> + StudentToProject()
+ main(args: String[]): void

<<Output>>
CS3821: ManyToOneDetails
Generated text file of all details from instance of algorithm.

1. Student table, featuring all Student details.
2. Project Table, featuring all Project details
3. All computational steps of algorithm, alongside all iterations.
4. Final matching of algorithm.