User (abstract class)

variables

String username

#String password

String first Name

String lastName

Course Directory directory

methods

aetusername() get Username() getPassword () setPassword() getfirst Name () set First Name () get Last Name () setLastName() get Directory() set Directory L) viewCourse S() printmyInfo()

Student Interface

Defines what a should be able to do. Student must implement all methods.

> viewCourses() riewNotfullCourses() register() withdraw() view Registered (oursesL)

add Acourse()

Admin Interface

This interface defines that an Admin should be able to do. Admin shust implement all methods.

CreateCourse()

delete Course()

edi+Course() displayCourseInfo()

registerStudent()

view course SC) viewFullCourses()

write Full Courses ()
view Students In Specificourse()

viewAStudentsCourses() Sort(ourses()



Admin

Admin inherits from User & implements the Admin Interface.

Constructor : Admin()

methods:

CreateCourse()

delete Course()

edi+Course()

display(ourseInfo()

registerStudent() viewCourses() viewFullCourses()

write Full Courses ()

viewStudents In SpecificCourse()

viewAStudents Courses()

Sort (ourses ()

Student

Student inherits from User & implements the StudentInterface.

Constructors:

Student()

Student (String first Name, String last Name)

Student (String first Name, String both ame, String wering String possword)

Additional variable:

ArrayList< Course> my courses

viewCourses()

riewNotfull Courses()

view Registered (ourses L)





register()

withdraw()

add Acourse()

Course Here we define what a course CourseDirectory This class is where we store an Arrayhist of courses as well as an Arrayhist of Students. Both an Admin and a Student have their own CourseDirect object is. Variables: int sectionNum variables: String name # Array List < Courses > courses List String id int max Students Num String course Location # ArrayList < Student > students List int num Registered Students Arraylist (Student) Students String instructor Constructor: public Course Directory () Constructors: methods: viewCourses() get Students List() set Students List() Public (our sel) public Course(name, id, maxstudents Num viewNotfullCourses() num Registered Students, students Registered, getCoursesLiSt() register() kreate NewCourse () instructor, section Num, course Location) withdraw() Methods: delete Course() view Registered (oursesL) edi+Course() getName() display(ourseInfol)
registerStudent() set Courses () get Instructor() setName() setInstructor() viewAIICourses() get Section Num() setId() viewFullCourses() write Full Courses ()
view Students In Specific Course() sotSection Num() SO+Id() get (ourseLocation() getmax Students Num() rienAstudents Courses() setCourseLocation() Sort (ourses() 50+MaxStudentsWum() display(ourseInfol) getNum Registered Students () display All Courses () setNum Resistered Students () displayCourses For Student() get Students Registered () setStudents Registered()

Course Registration Application

This class has the main() method.

main()

- 1) create an Admin & store their courseDirectory w/ info from the CSV file (here we read from the csv file)
- 2) tell the user to select if they are an Admin or Student (1st time=no students).
- 3) If the user is an Admin, ask for username & password & display the 2 menus where they decide what they want to do. Keep getting Admin's input until they exit.

 4) If the user is a student, ask for username & password & display the menu of what a student can choose to do. Continue until Student exits.
- 5) go back to the menu asking if the user is a Student or Admin & if they select "exit", then Stop the program entirely
- · whenever an Admin or a Student is done, serialize their course Directory whenever an Admin or Student logs in, deservalize the file containing course Directory info

Methods:

serialize Course Directory () deserialize Course Directory () (courses Management Admin Menu () reports AdminMenu() StudentMenu() read Course Data() treads from csv file