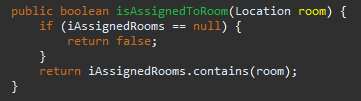
**BaseExam.java**

**•** It is used as the base class for the Exam entity and provides a set of properties and methods that are common to all exams in the application. The class defines various properties for an exam, including its name, note, length, exam size, print offset, seating type, assigned preference, average period, and unique ID rolled forward from.

The class provides several methods to access and modify these properties, such as getName(), setName(String name), getAssignedPeriod(), setAssignedPeriod(ExamPeriod assignedPeriod), getOwners(), setOwners(Set<ExamOwner> owners).

**• Updating the class:**



The above method was added. This method takes a Location object representing a room and returns a boolean indicating whether the exam is assigned to that room. It does this by checking whether the exam's iAssignedRooms set of assigned rooms is non-null and contains the given room.

**BaseDegree.java**

**•** It is used as the base class for the Degree entity and provides a set of properties and methods that are common to all degrees in the application. The class defines a String property iExternalUniqueId and an association with the Session entity object. The class provides several methods for accessing and modifying these properties, such as getExternalUniqueId(), setExternalUniqueId(String externalUniqueId), getSession(), setSession(Session session).

**• Updating the class:**



**•** This method returns a boolean indicating whether the degree has a non-null and non-empty externalUniqueId property.

**BaseUser.java**

**•** It is an abstract class and serves as a base for other classes that represent different types of users in the system. The class has three private instance variables: iUsername, iPassword, and iExternalUniqueId. The class has four public methods and three protected methods. The public methods are: getUsername(), setUsername(String username), getPassword(), setPassword(String password). The protected methods are: initialize(), equals(Object o), hashCode(). The class also has two constants (PROP\_USERNAME and PROP\_PASSWORD) that are used as keys for accessing the username and password properties of the User object.

**• Updating the class:**



This method returns a boolean indicating whether the user has an external unique identifier or not. It checks if the iExternalUniqueId instance variable is not null and not empty, and returns true if that's the case, and false otherwise.

**BaseWaitList.java**

• It is an abstract class and serves as a base for other classes that represent different types of waitlists in the system.

The class has eleven private instance variables that store information about the waitlist, including iUniqueId, iType, iTimestamp, iWaitListed, iChangedBy, iRequest, iEnrollment, iWaitListedTimeStamp, iStudent, iCourseOffering, iEnrolledCourse, iCourseDemand, and iSwapCourseOffering.

The class has fourteen public methods and one protected method. The public methods are: getUniqueId(), setUniqueId(Long uniqueId), getType(), setType(Integer type), getTimestamp(), setTimestamp(Date timestamp), isWaitListed(), getWaitListed(), setWaitListed(Boolean waitListed), getChangedBy(), setChangedBy(String changedBy), getRequest(), setRequest(String request), getEnrollment(), setEnrollment(String enrollment), getWaitListedTimeStamp(), setWaitListedTimeStamp(Date waitListedTimeStamp).

The protected method is: initialize(): This method is called when the object is created and can be used to initialize any instance variables.

**• Updating the class:**



This method returns a boolean indicating whether the waitlist has an enrollment or not. It checks if the iEnrollment instance variable is not null and not empty, and returns true if that's the case, and false otherwise.