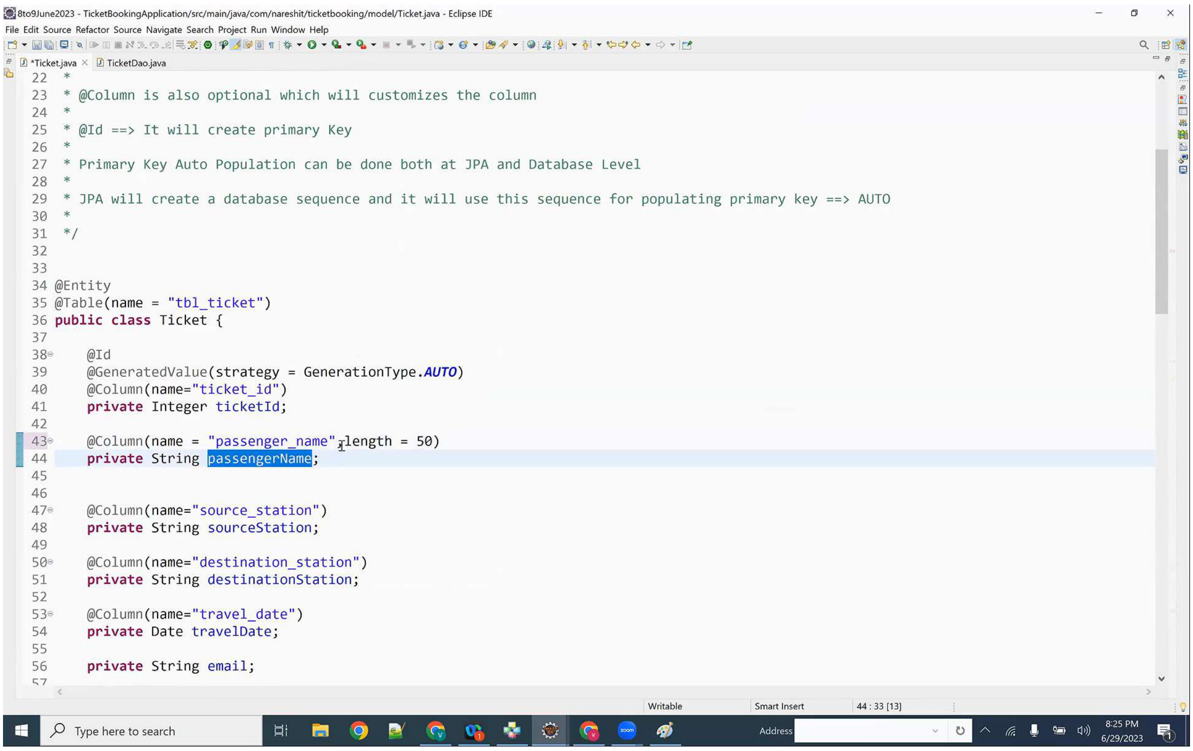
12

Wednesday, 02 July 2025 18:32


8to9June2023 - TicketBookingApplication/src/main/java/com/nareshit/ticketbooking/model/Ticket.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
*Ticket.java
TicketDao.java
Writable
Smart Insert
44 : 33 [13]
Type here to search
Address
8:25 PM
6/29/2023

```
 *
 * @Column is also optional which will customizes the column
 *
 * @Id ==> It will create primary Key
 *
 * Primary Key Auto Population can be done both at JPA and Database Level
 *
 * JPA will create a database sequence and it will use this sequence for populating primary key ==> AUTO
 *
 */


@Entity
@Table(name = "tbl_ticket")
public class Ticket {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="ticket_id")
    private Integer ticketId;

    @Column(name = "passenger_name", length = 50)
    private String passengerName;


    @Column(name="source_station")
    private String sourceStation;

    @Column(name="destination_station")
    private String destinationStation;

    @Column(name="travel_date")
    private Date travelDate;

    private String email;
```

the above example shows the mention the size of the data to be inserted in the column
@Column(name ="passenger_name", length=50)
    it accepts only 50 characters only

==

@Id

- Primary key should be a unique value.
- If users give the primary key as input, then we cannot gaurentee uniqueness of the primary key
- we can generate primary key
  - at database level
  - at JPA level (framework level)

- Primary key auto population can be done both at JPA and Database level

- JPA will create a database sequence and it will use the sequence for populating the primary keys

==

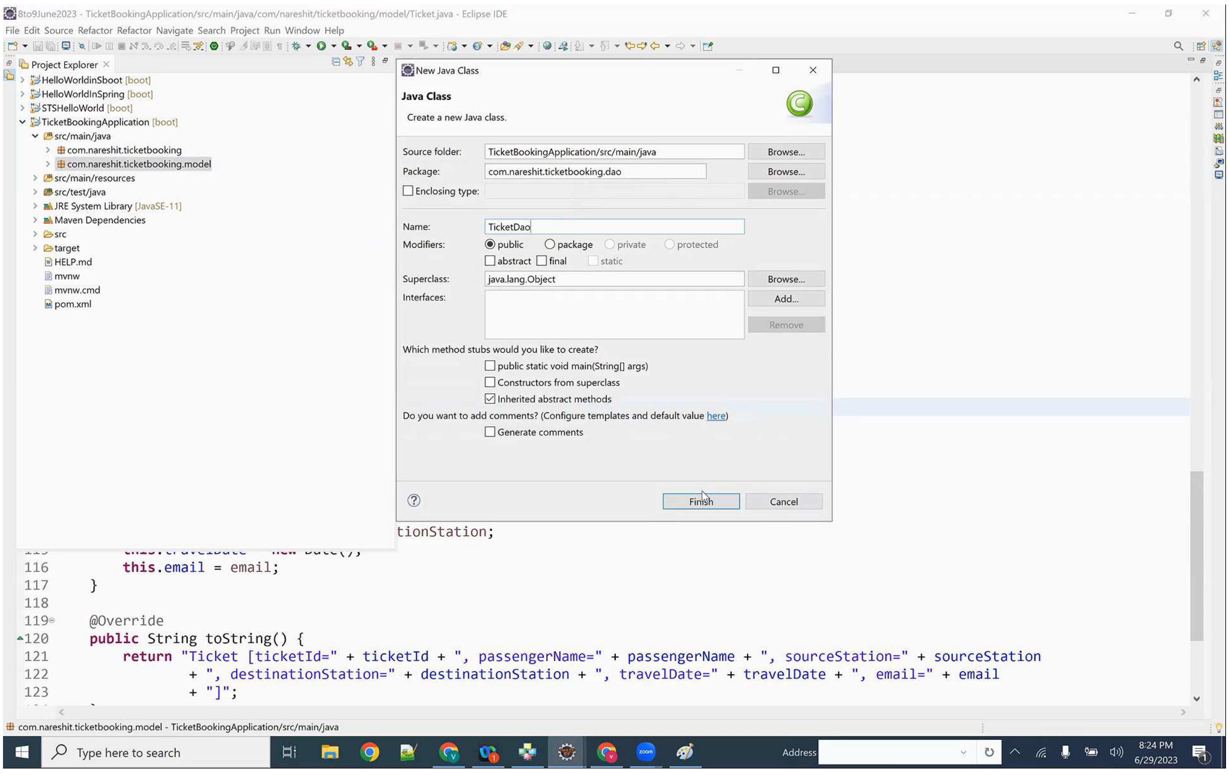
@GeneratedValue(startegy=GenerationType.Auto)

- This annotation is used to generate the primary keys.
- JPA will create a sequence, this sequence used to populate the primary key.

==

what is the purpose of the DAO class ?

- it is a wrapper up on the Database



8to9June2023 - TicketBookingApplication/src/main/java/com/nareshit/ticketbooking/model/Ticket.java - Eclipse IDE

File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Project Explorer

- HelloWorldinSboot [boot]
- HelloWorldInSpring [boot]
- STSHelloWorld [boot]
- TicketBookingApplication [boot]
  - src/main/java
    - com.nareshit.ticketbooking
    - com.nareshit.ticketbooking.model
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-11]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

New Java Class

Java Class

Create a new Java class.

Source folder: TicketBookingApplication/src/main/java

Package: com.nareshit.ticketbooking.dao

☐ Enclosing type:

Name: TicketDao

Modifiers: ● public ○ package ○ private ○ protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

☐ Generate comments

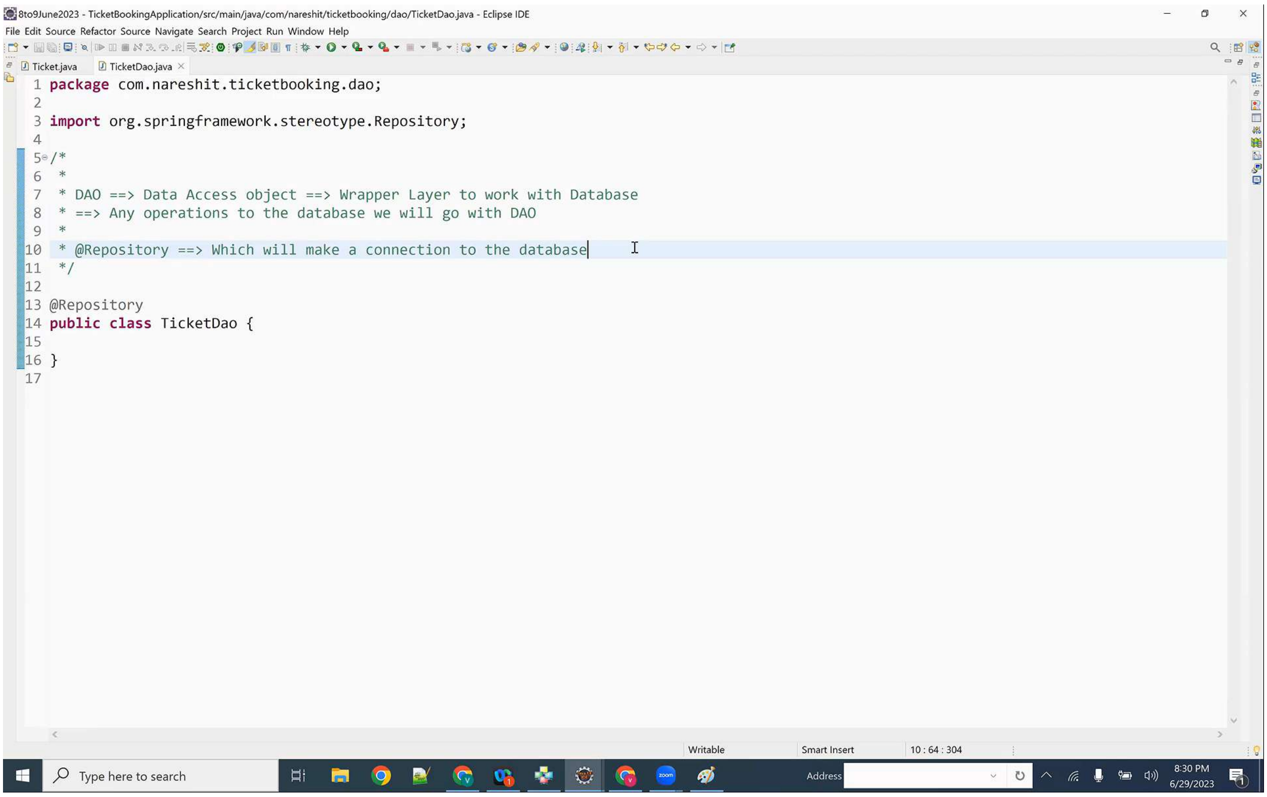
```
tionStation;
        this.email = email;
    }

    @Override
    public String toString() {
        return "Ticket [ticketId=" + ticketId + ", passengerName=" + passengerName + ", sourceStation=" + sourceStation
                + ", destinationStation=" + destinationStation + ", travelDate=" + travelDate + ", email=" + email
                + "]";
```

com.nareshit.ticketbooking.model - TicketBookingApplication/src/main/java

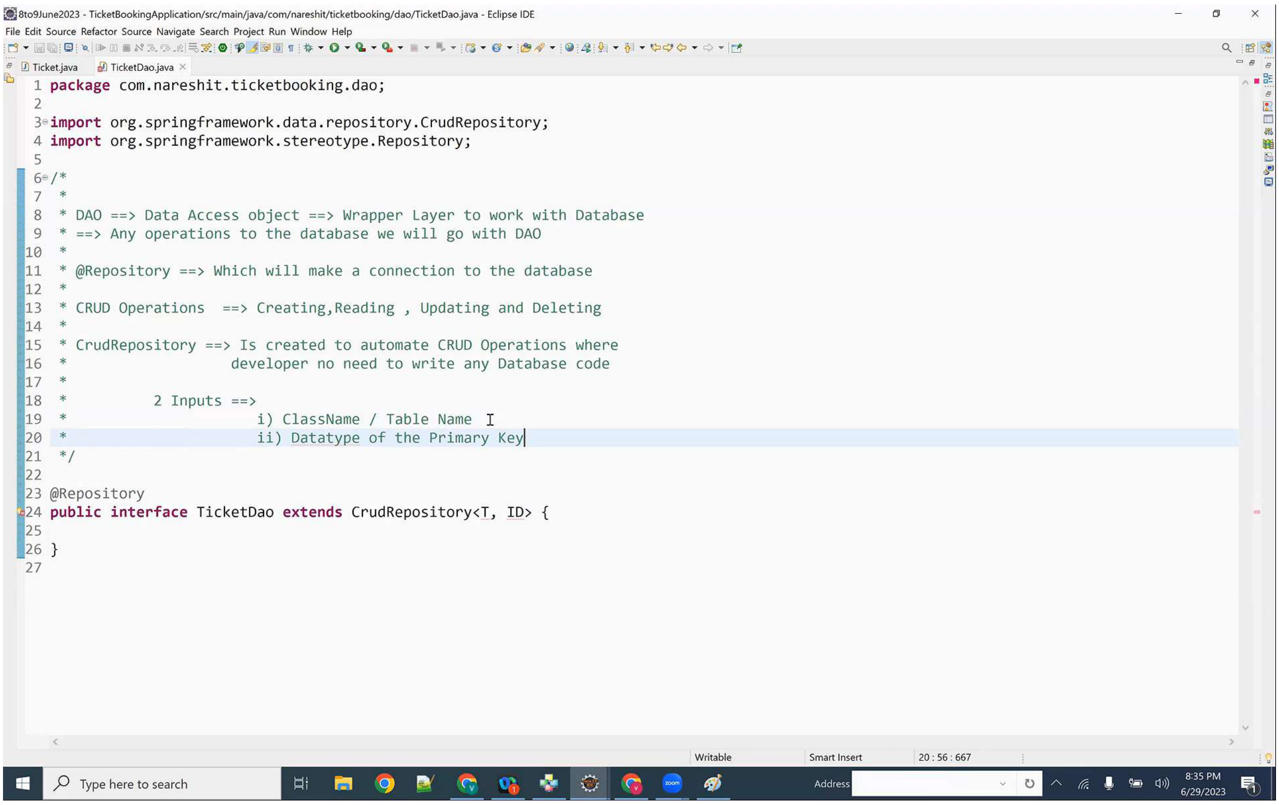
@Repository

- This annotation, which will make a connection to the database
- CRUD Operations
  - CrudRepository --> it is created to perform crud operations, where developer no need to write any sql code.
    - there are 2 inputs for the CrudRepository
      - ☐ ClassName
      - ☐ Datatype of the Primary Key

save --> insert the data or update the data
findById --> Retrieve the data
findAll --> Retrieve all the data
deleteById --> delete the record based on Id


8to9June2023 - TicketBookingApplication/src/main/java/com/nareshit/ticketbooking/dao/TicketDao.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Ticket.java
TicketDao.java
Writable
Smart Insert
10 : 64 : 304
Type here to search
Address
8:30 PM
6/29/2023

```java
package com.nareshit.ticketbooking.dao;

import org.springframework.stereotype.Repository;

/*
 *
 * DAO ==> Data Access object ==> Wrapper Layer to work with Database
 * ==> Any operations to the database we will go with DAO
 *
 * @Repository ==> Which will make a connection to the database
 */

@Repository
public class TicketDao {

}
```


8to9June2023 - TicketBookingApplication/src/main/java/com/nareshit/ticketbooking/dao/TicketDao.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Ticket.java
TicketDao.java
Writable
Smart Insert
20 : 56 : 667
Type here to search
Address
8:35 PM
6/29/2023

```java
package com.nareshit.ticketbooking.dao;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

/*
 *
 * DAO ==> Data Access object ==> Wrapper Layer to work with Database
 * ==> Any operations to the database we will go with DAO
 *
 * @Repository ==> Which will make a connection to the database
 *
 * CRUD Operations  ==> Creating,Reading , Updating and Deleting
 *
 * CrudRepository ==> Is created to automate CRUD Operations where
 *                  developer no need to write any Database code
 *
 *          2 Inputs ==>
 *                      i) ClassName / Table Name
 *                      ii) Datatype of the Primary Key
 */

@Repository
public interface TicketDao extends CrudRepository<T, ID> {

}
```

==

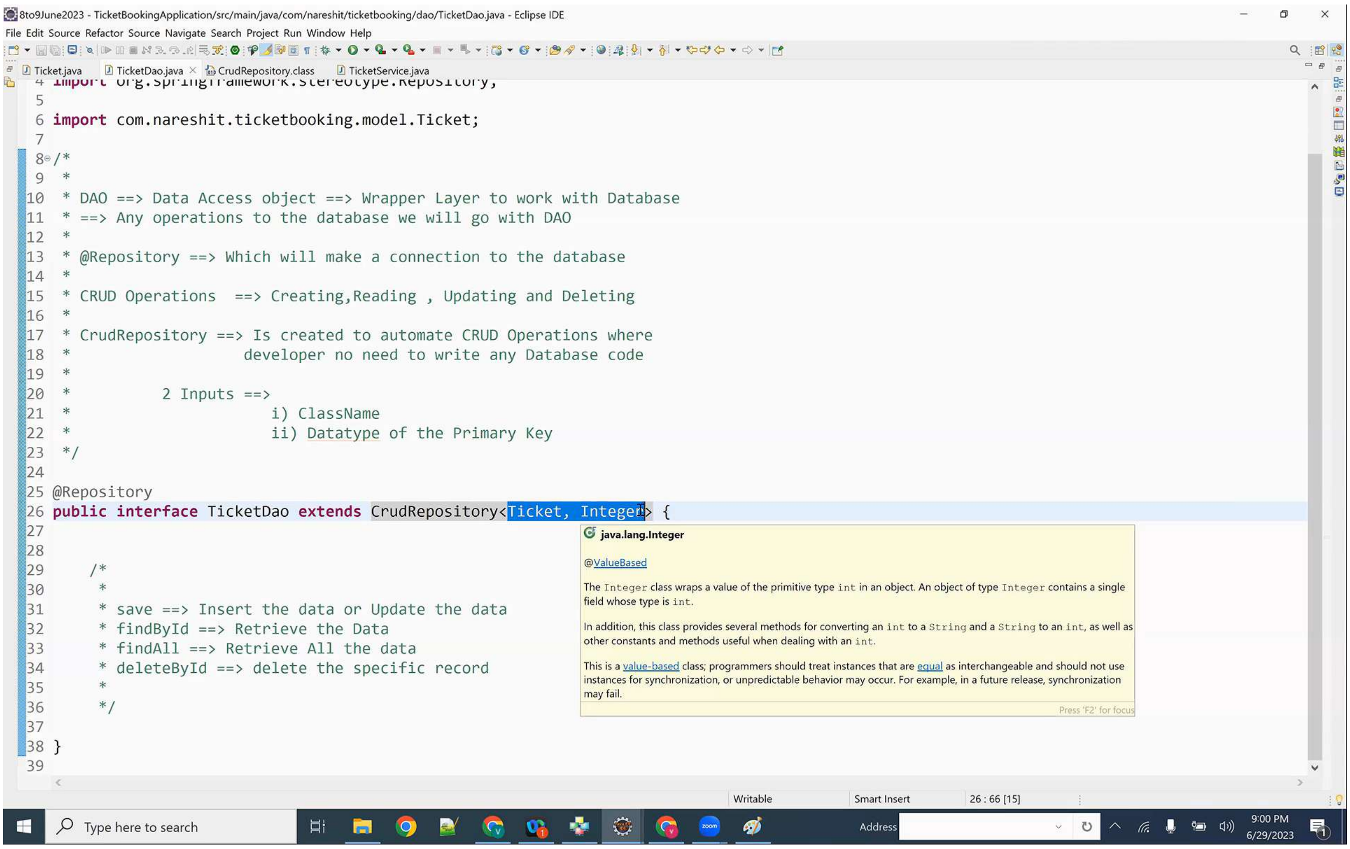
@Service

- This annotation is used, to write our business logic
- it has implemented transaction management capability
- this service layer interact with dao layer

@Autowired

Iterable

A generic datastructure which work with list, set, ..


8to9June2023 - TicketBookingApplication/src/main/java/com/nareshit/ticketbooking/dao/TicketDao.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Ticket.java
TicketDao.java
CrudRepository.class
TicketService.java

```java
import org.springframework.stereotype.Repository;

import com.nareshit.ticketbooking.model.Ticket;

/*
 *
 * DAO ==> Data Access object ==> Wrapper Layer to work with Database
 * ==> Any operations to the database we will go with DAO
 *
 * @Repository ==> Which will make a connection to the database
 *
 * CRUD Operations  ==> Creating,Reading , Updating and Deleting
 *
 * CrudRepository ==> Is created to automate CRUD Operations where
 *                  developer no need to write any Database code
 *
 *          2 Inputs ==>
 *                      i) ClassName
 *                      ii) Datatype of the Primary Key
 */

@Repository
public interface TicketDao extends CrudRepository<Ticket, Integer> {


    /*
     *
     * save ==> Insert the data or Update the data
     * findById ==> Retrieve the Data
     * findAll ==> Retrieve All the data
     * deleteById ==> delete the specific record
     *
     */

}
```

java.lang.Integer

@ValueBased

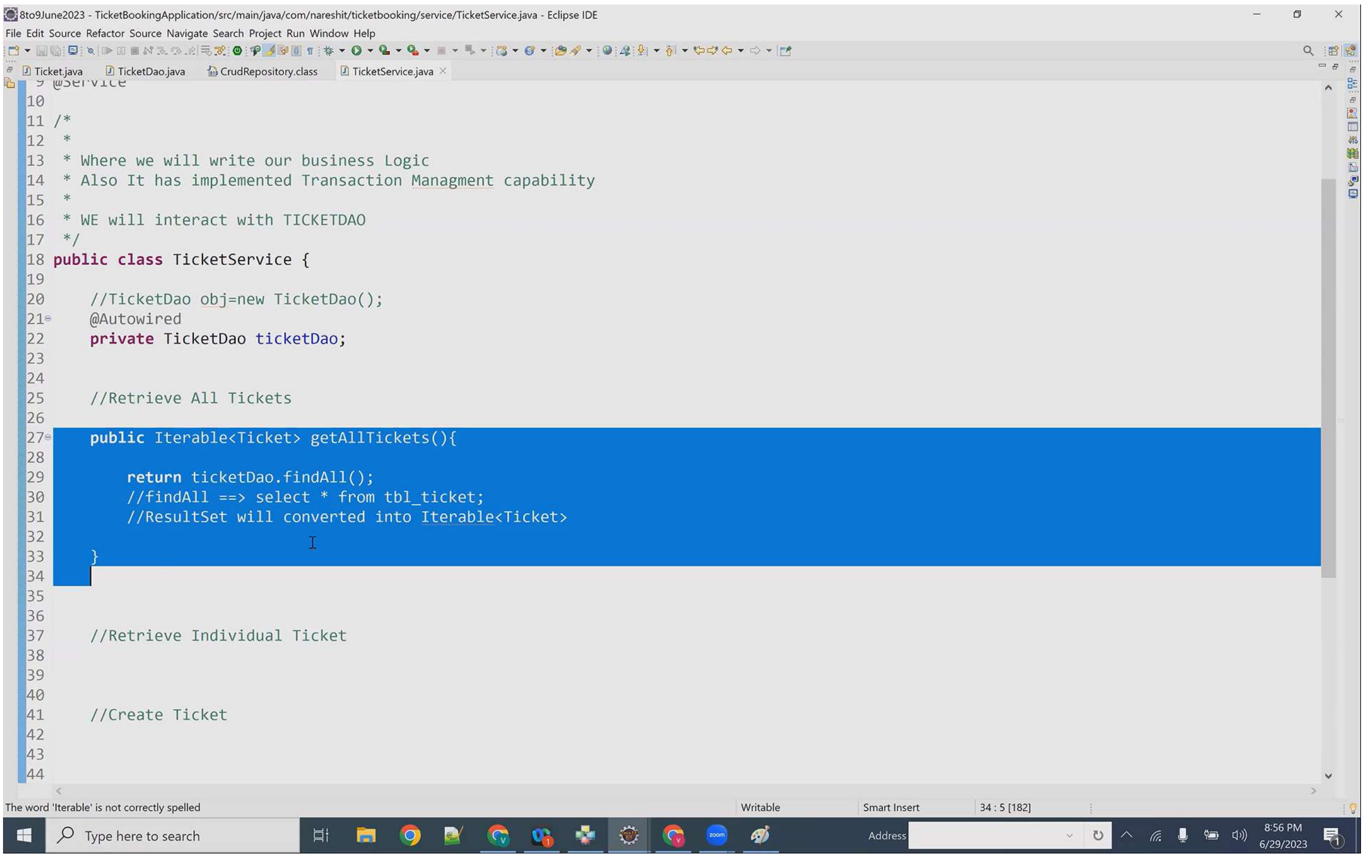
The Integer class wraps a value of the primitive type int in an object. An object of type Integer contains a single field whose type is int.

In addition, this class provides several methods for converting an int to a String and a String to an int, as well as other constants and methods useful when dealing with an int.

This is a value-based class; programmers should treat instances that are equal as interchangeable and should not use instances for synchronization, or unpredictable behavior may occur. For example, in a future release, synchronization may fail.

Press 'F2' for focus

Writable
Smart Insert
26 : 66 [15]
Type here to search
Address
9:00 PM
6/29/2023


8to9June2023 - TicketBookingApplication/src/main/java/com/nareshit/ticketbooking/service/TicketService.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Ticket.java
TicketDao.java
CrudRepository.class
TicketService.java

```java
@Service

/*
 *
 * Where we will write our business Logic
 * Also It has implemented Transaction Managment capability
 *
 * WE will interact with TICKETDAO
 */
public class TicketService {

    //TicketDao obj=new TicketDao();
    @Autowired
    private TicketDao ticketDao;


    //Retrieve All Tickets

    public Iterable<Ticket> getAllTickets(){

        return ticketDao.findAll();
        //findAll ==> select * from tbl_ticket;
        //ResultSet will converted into Iterable<Ticket>

    }



    //Retrieve Individual Ticket



    //Create Ticket



```

The word 'Iterable' is not correctly spelled