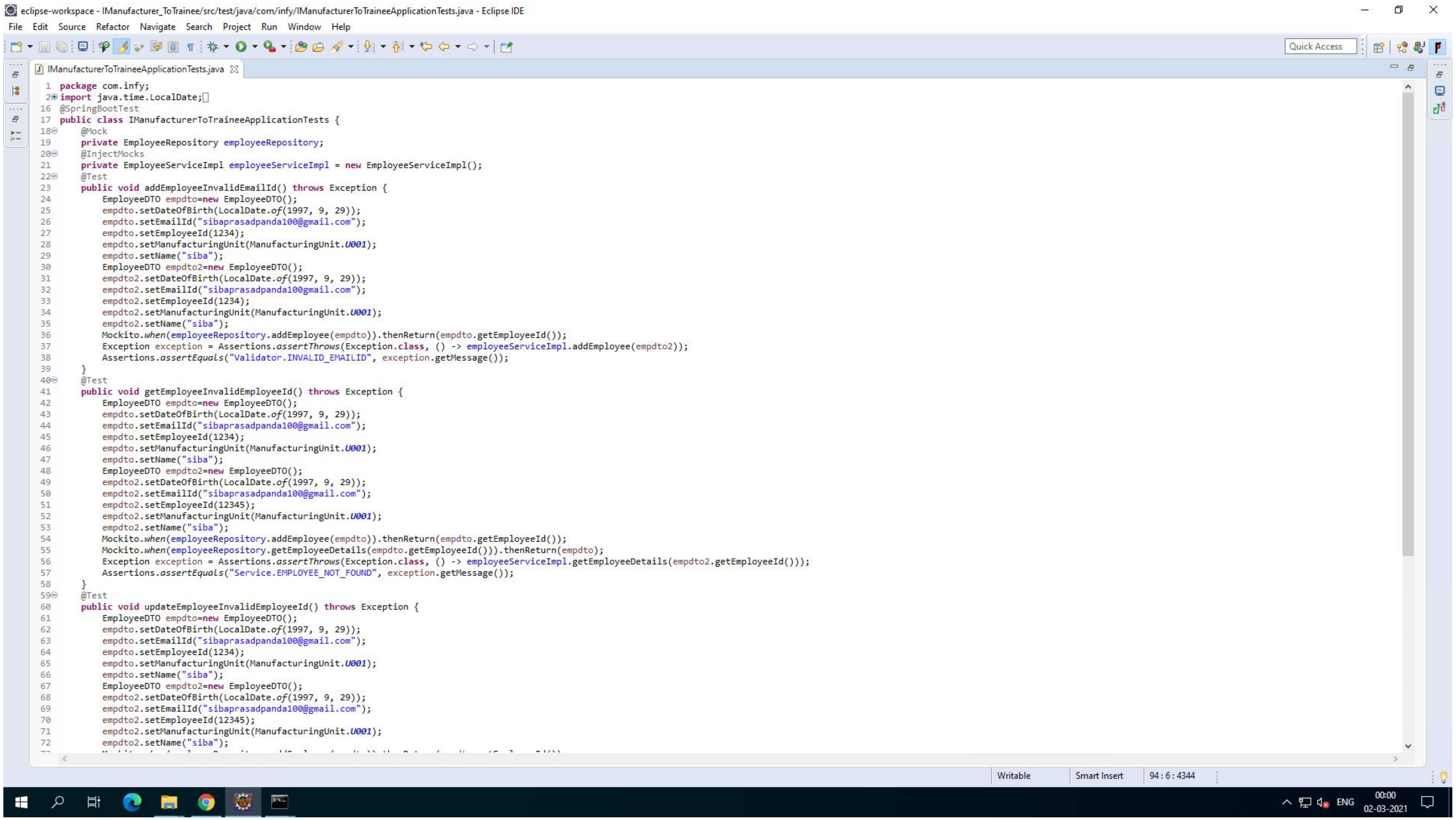
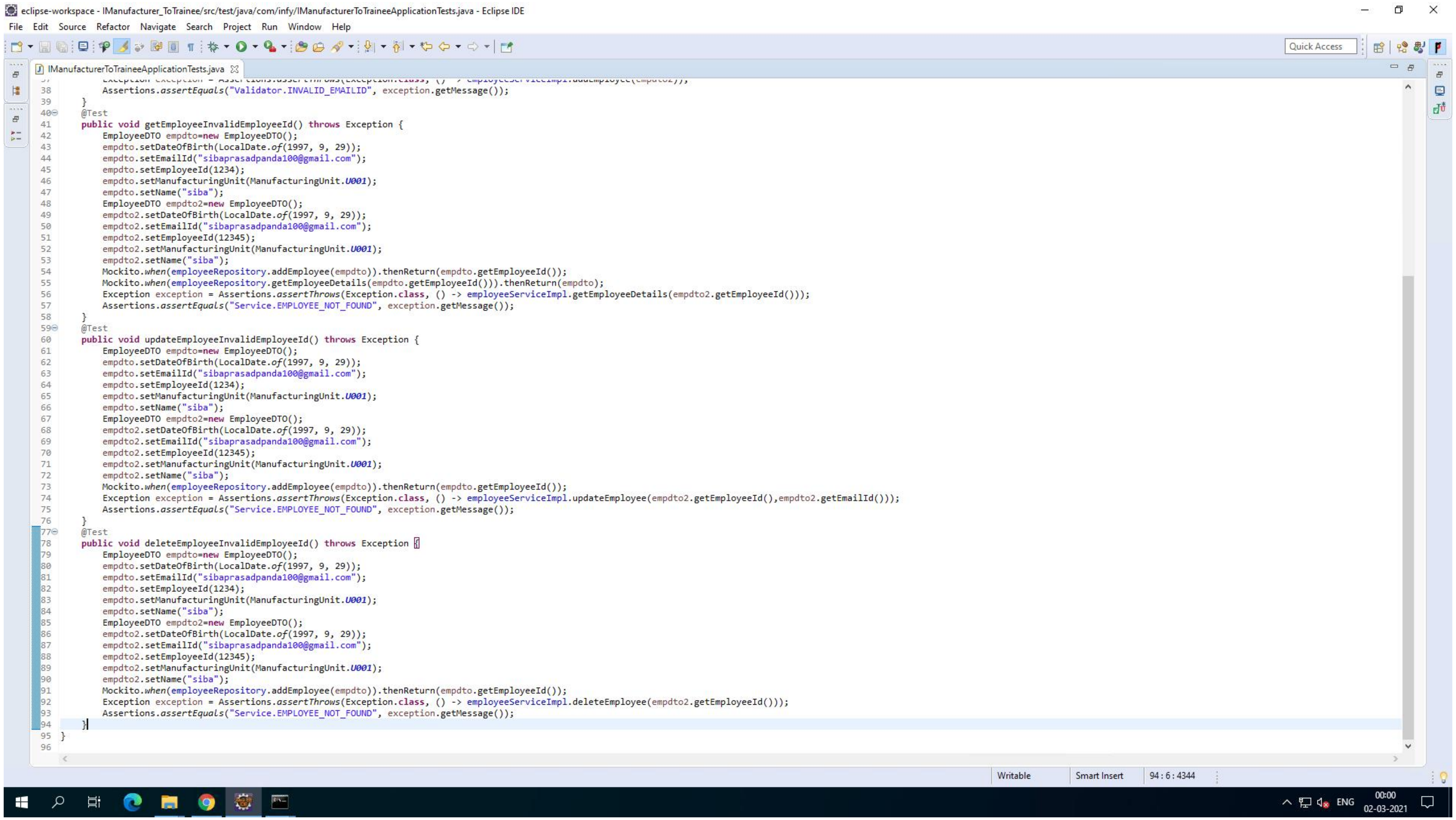


eclipse-workspace - IManufacturer_ToTrainee/src/test/java/com/infy/IManufacturerToTraineeApplicationTests.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
IManufacturerToTraineeApplicationTests.java
Writable
Smart Insert
94 : 6 : 4344
ENG
00:00
02-03-2021

```java
package com.infy;
import java.time.LocalDate;
@SpringBootTest
public class IManufacturerToTraineeApplicationTests {
    @Mock
    private EmployeeRepository employeeRepository;
    @InjectMocks
    private EmployeeServiceImpl employeeServiceImpl = new EmployeeServiceImpl();
    @Test
    public void addEmployeeInvalidEmailId() throws Exception {
        EmployeeDTO empdto=new EmployeeDTO();
        empdto.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto.setEmailId("sibaprasadpanda100@gmail.com");
        empdto.setEmployeeId(1234);
        empdto.setManufacturingUnit(ManufacturingUnit.U001);
        empdto.setName("siba");
        EmployeeDTO empdto2=new EmployeeDTO();
        empdto2.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto2.setEmailId("sibaprasadpanda100gmail.com");
        empdto2.setEmployeeId(1234);
        empdto2.setManufacturingUnit(ManufacturingUnit.U001);
        empdto2.setName("siba");
        Mockito.when(employeeRepository.addEmployee(empdto)).thenReturn(empdto.getEmployeeId());
        Exception exception = Assertions.assertThrows(Exception.class, () -> employeeServiceImpl.addEmployee(empdto2));
        Assertions.assertEquals("Validator.INVALID_EMAILID", exception.getMessage());
    }
    @Test
    public void getEmployeeInvalidEmployeeId() throws Exception {
        EmployeeDTO empdto=new EmployeeDTO();
        empdto.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto.setEmailId("sibaprasadpanda100@gmail.com");
        empdto.setEmployeeId(1234);
        empdto.setManufacturingUnit(ManufacturingUnit.U001);
        empdto.setName("siba");
        EmployeeDTO empdto2=new EmployeeDTO();
        empdto2.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto2.setEmailId("sibaprasadpanda100@gmail.com");
        empdto2.setEmployeeId(12345);
        empdto2.setManufacturingUnit(ManufacturingUnit.U001);
        empdto2.setName("siba");
        Mockito.when(employeeRepository.addEmployee(empdto)).thenReturn(empdto.getEmployeeId());
        Mockito.when(employeeRepository.getEmployeeDetails(empdto.getEmployeeId())).thenReturn(empdto);
        Exception exception = Assertions.assertThrows(Exception.class, () -> employeeServiceImpl.getEmployeeDetails(empdto2.getEmployeeId()));
        Assertions.assertEquals("Service.EMPLOYEE_NOT_FOUND", exception.getMessage());
    }
    @Test
    public void updateEmployeeInvalidEmployeeId() throws Exception {
        EmployeeDTO empdto=new EmployeeDTO();
        empdto.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto.setEmailId("sibaprasadpanda100@gmail.com");
        empdto.setEmployeeId(1234);
        empdto.setManufacturingUnit(ManufacturingUnit.U001);
        empdto.setName("siba");
        EmployeeDTO empdto2=new EmployeeDTO();
        empdto2.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto2.setEmailId("sibaprasadpanda100@gmail.com");
        empdto2.setEmployeeId(12345);
        empdto2.setManufacturingUnit(ManufacturingUnit.U001);
        empdto2.setName("siba");
```



File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access

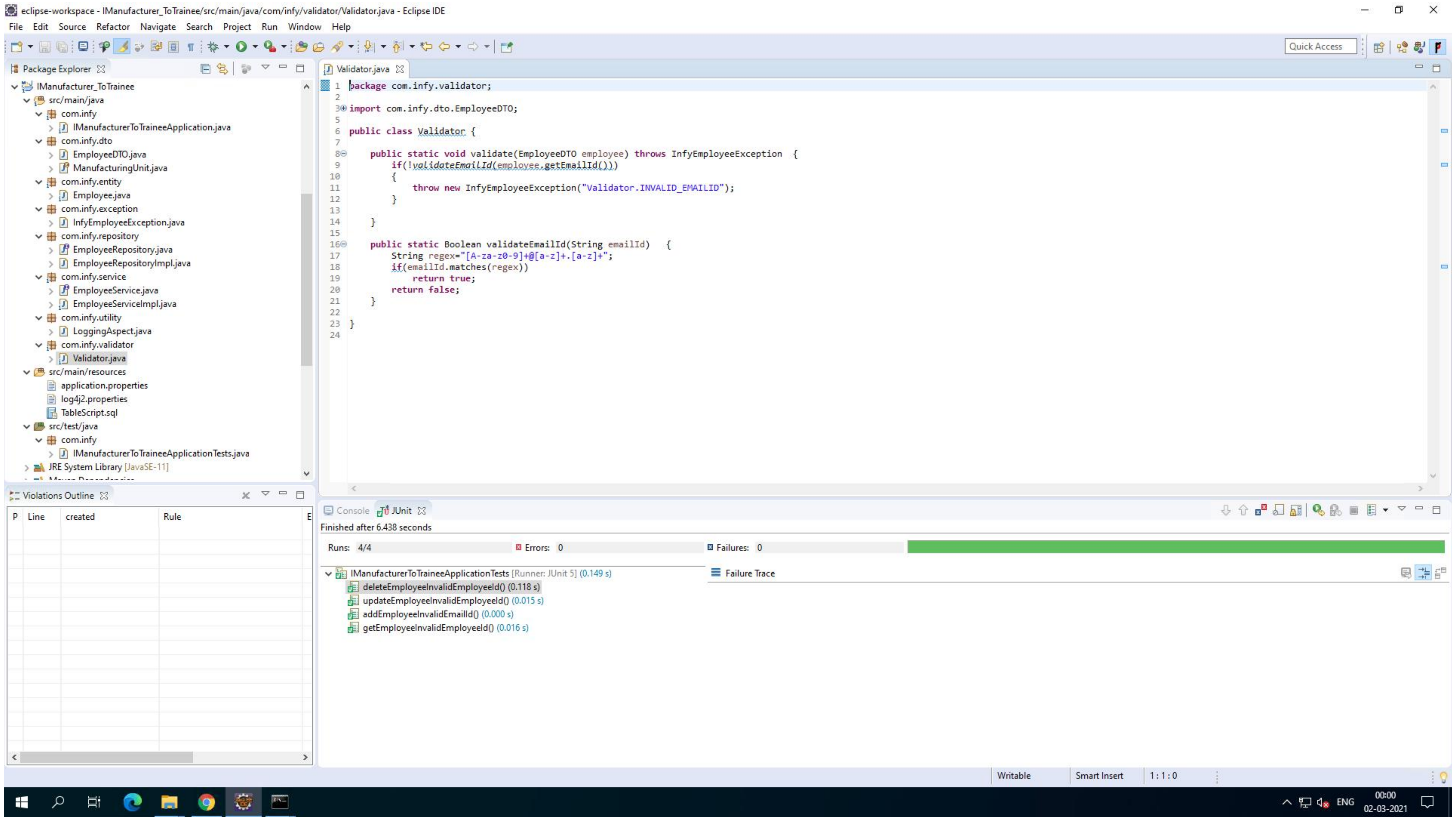
eclipse-workspace - IManufacturer_ToTrainee/src/test/java/com/infy/IManufacturerToTraineeApplicationTests.java - Eclipse IDE

IManufacturerToTraineeApplicationTests.java

```java
        Exception exception = Assertions.assertThrows(Exception.class, () -> employeeServiceImpl.addEmployee(empdto2));
        Assertions.assertEquals("Validator.INVALID_EMAILID", exception.getMessage());
    }
    @Test
    public void getEmployeeInvalidEmployeeId() throws Exception {
        EmployeeDTO empdto=new EmployeeDTO();
        empdto.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto.setEmailId("sibaprasadpanda100@gmail.com");
        empdto.setEmployeeId(1234);
        empdto.setManufacturingUnit(ManufacturingUnit.U001);
        empdto.setName("siba");
        EmployeeDTO empdto2=new EmployeeDTO();
        empdto2.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto2.setEmailId("sibaprasadpanda100@gmail.com");
        empdto2.setEmployeeId(12345);
        empdto2.setManufacturingUnit(ManufacturingUnit.U001);
        empdto2.setName("siba");
        Mockito.when(employeeRepository.addEmployee(empdto)).thenReturn(empdto.getEmployeeId());
        Mockito.when(employeeRepository.getEmployeeDetails(empdto.getEmployeeId())).thenReturn(empdto);
        Exception exception = Assertions.assertThrows(Exception.class, () -> employeeServiceImpl.getEmployeeDetails(empdto2.getEmployeeId()));
        Assertions.assertEquals("Service.EMPLOYEE_NOT_FOUND", exception.getMessage());
    }
    @Test
    public void updateEmployeeInvalidEmployeeId() throws Exception {
        EmployeeDTO empdto=new EmployeeDTO();
        empdto.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto.setEmailId("sibaprasadpanda100@gmail.com");
        empdto.setEmployeeId(1234);
        empdto.setManufacturingUnit(ManufacturingUnit.U001);
        empdto.setName("siba");
        EmployeeDTO empdto2=new EmployeeDTO();
        empdto2.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto2.setEmailId("sibaprasadpanda100@gmail.com");
        empdto2.setEmployeeId(12345);
        empdto2.setManufacturingUnit(ManufacturingUnit.U001);
        empdto2.setName("siba");
        Mockito.when(employeeRepository.addEmployee(empdto)).thenReturn(empdto.getEmployeeId());
        Exception exception = Assertions.assertThrows(Exception.class, () -> employeeServiceImpl.updateEmployee(empdto2.getEmployeeId(),empdto2.getEmailId()));
        Assertions.assertEquals("Service.EMPLOYEE_NOT_FOUND", exception.getMessage());
    }
    @Test
    public void deleteEmployeeInvalidEmployeeId() throws Exception {
        EmployeeDTO empdto=new EmployeeDTO();
        empdto.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto.setEmailId("sibaprasadpanda100@gmail.com");
        empdto.setEmployeeId(1234);
        empdto.setManufacturingUnit(ManufacturingUnit.U001);
        empdto.setName("siba");
        EmployeeDTO empdto2=new EmployeeDTO();
        empdto2.setDateOfBirth(LocalDate.of(1997, 9, 29));
        empdto2.setEmailId("sibaprasadpanda100@gmail.com");
        empdto2.setEmployeeId(12345);
        empdto2.setManufacturingUnit(ManufacturingUnit.U001);
        empdto2.setName("siba");
        Mockito.when(employeeRepository.addEmployee(empdto)).thenReturn(empdto.getEmployeeId());
        Exception exception = Assertions.assertThrows(Exception.class, () -> employeeServiceImpl.deleteEmployee(empdto2.getEmployeeId()));
        Assertions.assertEquals("Service.EMPLOYEE_NOT_FOUND", exception.getMessage());
    }
}
```

Writable   Smart Insert   94 : 6 : 4344

ENG 00:00 02-03-2021

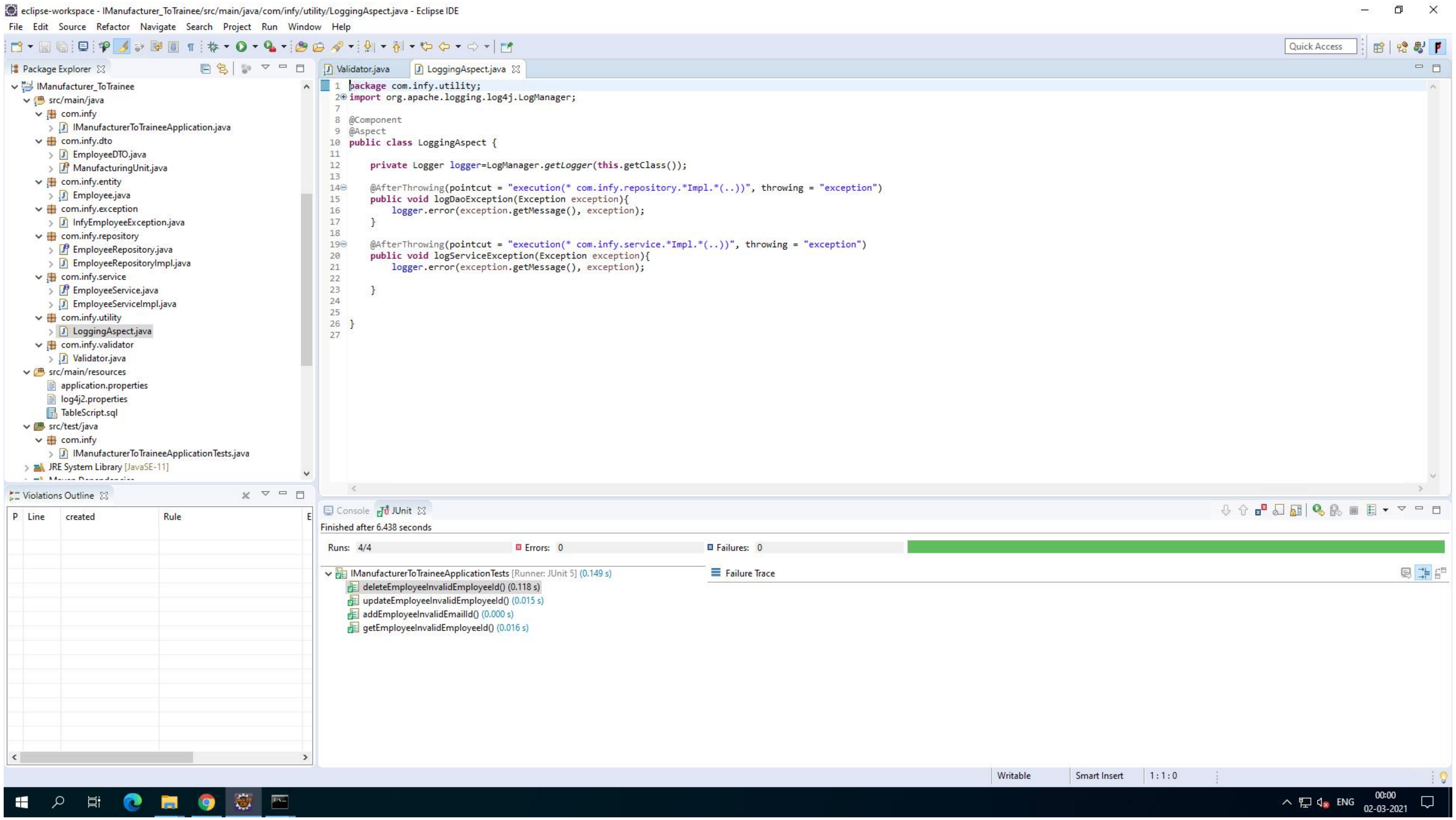


eclipse-workspace - IManufacturer_ToTrainee/src/main/java/com/infy/validator/Validator.java - Eclipse IDE

```java
package com.infy.validator;

import com.infy.dto.EmployeeDTO;

public class Validator {

    public static void validate(EmployeeDTO employee) throws InfyEmployeeException  {
        if(!validateEmailId(employee.getEmailId()))
        {
            throw new InfyEmployeeException("Validator.INVALID_EMAILID");
        }

    }

    public static Boolean validateEmailId(String emailId)   {
        String regex="[A-za-z0-9]+@[a-z]+.[a-z]+";
        if(emailId.matches(regex))
            return true;
        return false;
    }

}
```

```
IManufacturer_ToTrainee
  src/main/java
    com.infy
      IManufacturerToTraineeApplication.java
    com.infy.dto
      EmployeeDTO.java
      ManufacturingUnit.java
    com.infy.entity
      Employee.java
    com.infy.exception
      InfyEmployeeException.java
    com.infy.repository
      EmployeeRepository.java
      EmployeeRepositoryImpl.java
    com.infy.service
      EmployeeService.java
      EmployeeServiceImpl.java
    com.infy.utility
      LoggingAspect.java
    com.infy.validator
      Validator.java
  src/main/resources
    application.properties
    log4j2.properties
    TableScript.sql
  src/test/java
    com.infy
      IManufacturerToTraineeApplicationTests.java
  JRE System Library [JavaSE-11]
```

```
Finished after 6.438 seconds
Runs: 4/4        Errors: 0        Failures: 0

IManufacturerToTraineeApplicationTests [Runner: JUnit 5] (0.149 s)
    deleteEmployeeInvalidEmployeeId() (0.118 s)
    updateEmployeeInvalidEmployeeId() (0.015 s)
    addEmployeeInvalidEmailId() (0.000 s)
    getEmployeeInvalidEmployeeId() (0.016 s)
```

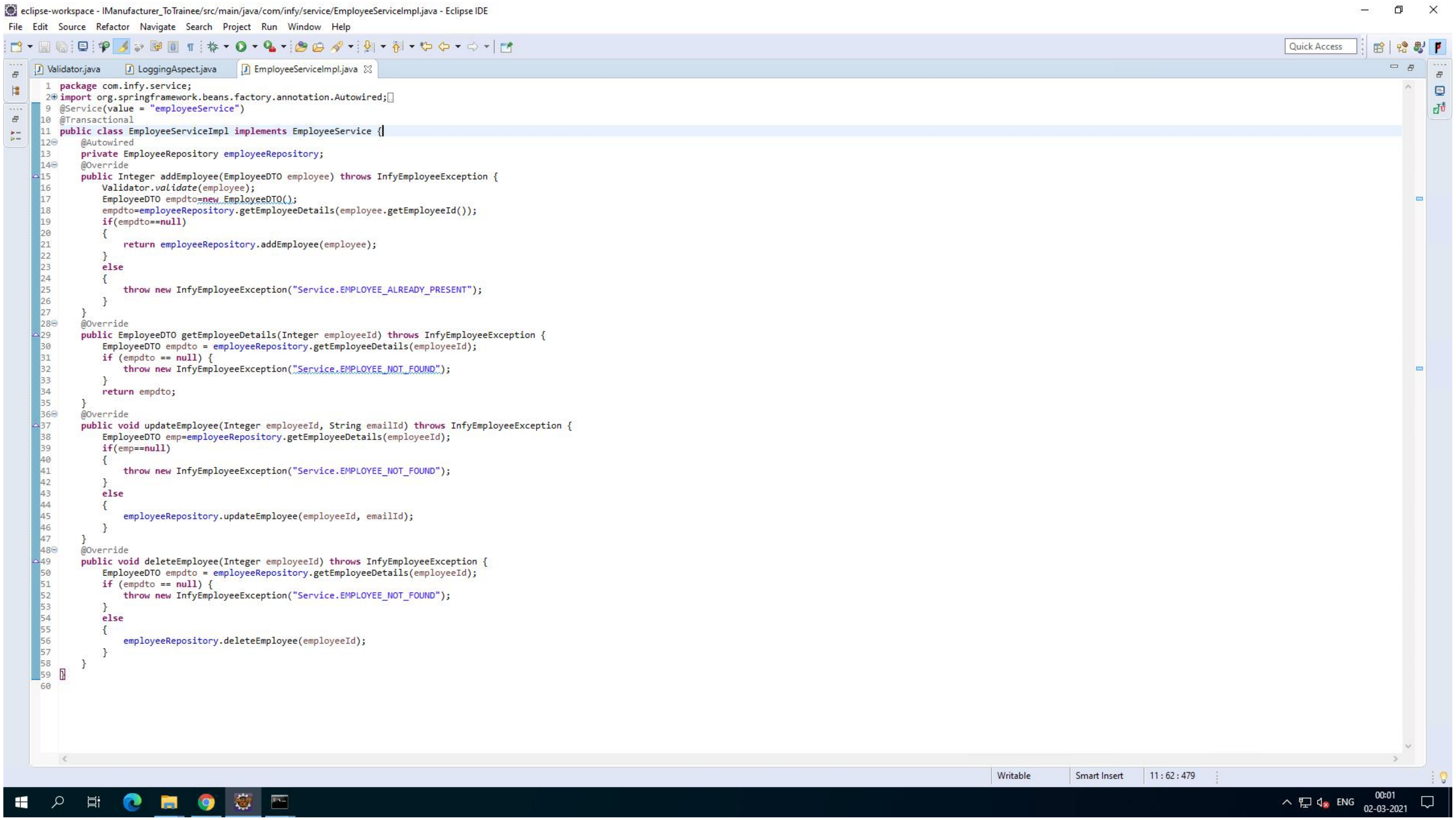


eclipse-workspace - IManufacturer_ToTrainee/src/main/java/com/infy/utility/LoggingAspect.java - Eclipse IDE

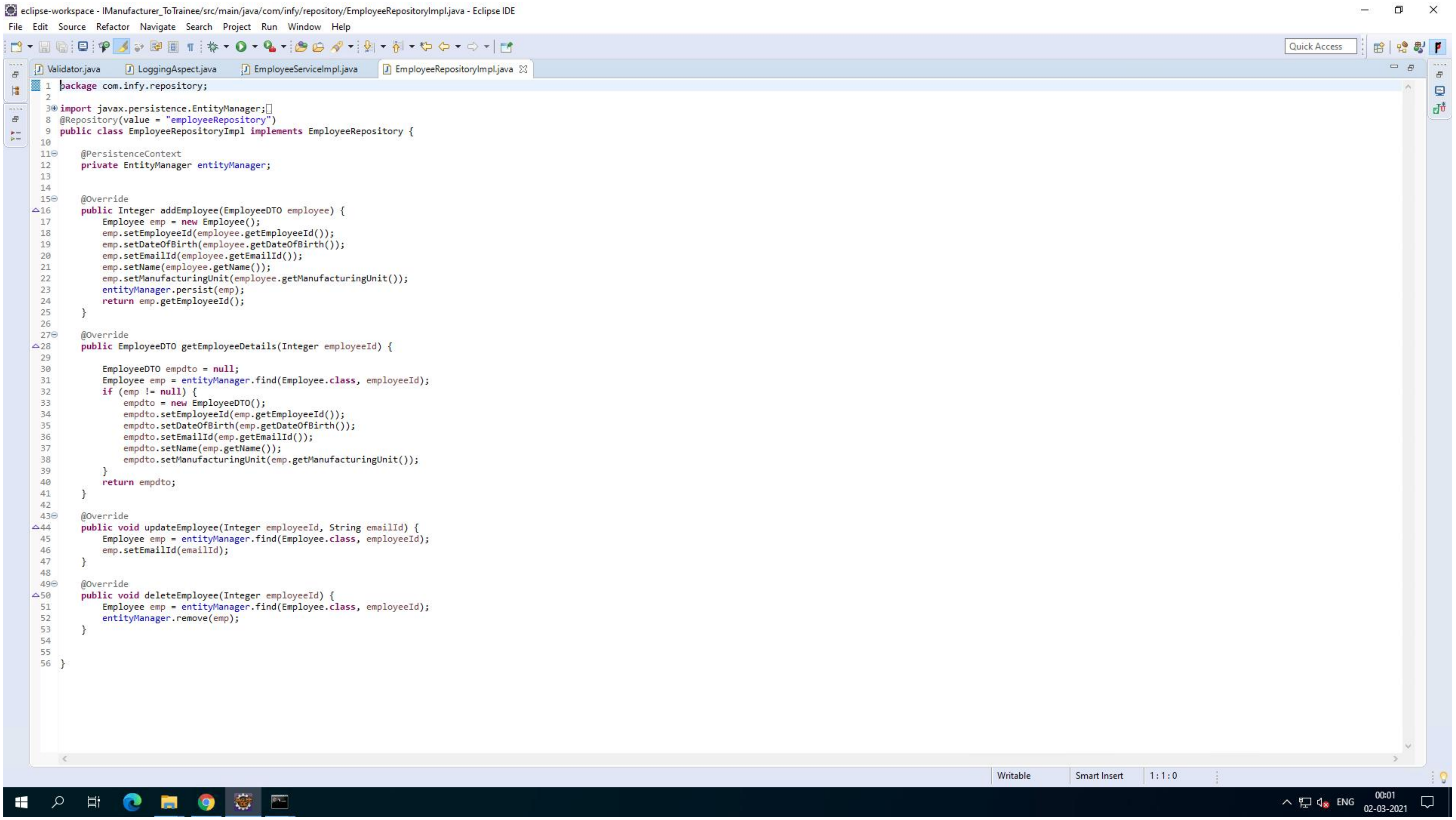
```
IManufacturer_ToTrainee
  src/main/java
    com.infy
      IManufacturerToTraineeApplication.java
    com.infy.dto
      EmployeeDTO.java
      ManufacturingUnit.java
    com.infy.entity
      Employee.java
    com.infy.exception
      InfyEmployeeException.java
    com.infy.repository
      EmployeeRepository.java
      EmployeeRepositoryImpl.java
    com.infy.service
      EmployeeService.java
      EmployeeServiceImpl.java
    com.infy.utility
      LoggingAspect.java
    com.infy.validator
      Validator.java
  src/main/resources
    application.properties
    log4j2.properties
    TableScript.sql
  src/test/java
    com.infy
      IManufacturerToTraineeApplicationTests.java
  JRE System Library [JavaSE-11]
```

```java
package com.infy.utility;
import org.apache.logging.log4j.LogManager;

@Component
@Aspect
public class LoggingAspect {

    private Logger logger=LogManager.getLogger(this.getClass());

    @AfterThrowing(pointcut = "execution(* com.infy.repository.*Impl.*(..))", throwing = "exception")
    public void logDaoException(Exception exception){
        logger.error(exception.getMessage(), exception);
    }

    @AfterThrowing(pointcut = "execution(* com.infy.service.*Impl.*(..))", throwing = "exception")
    public void logServiceException(Exception exception){
        logger.error(exception.getMessage(), exception);

    }


}
```

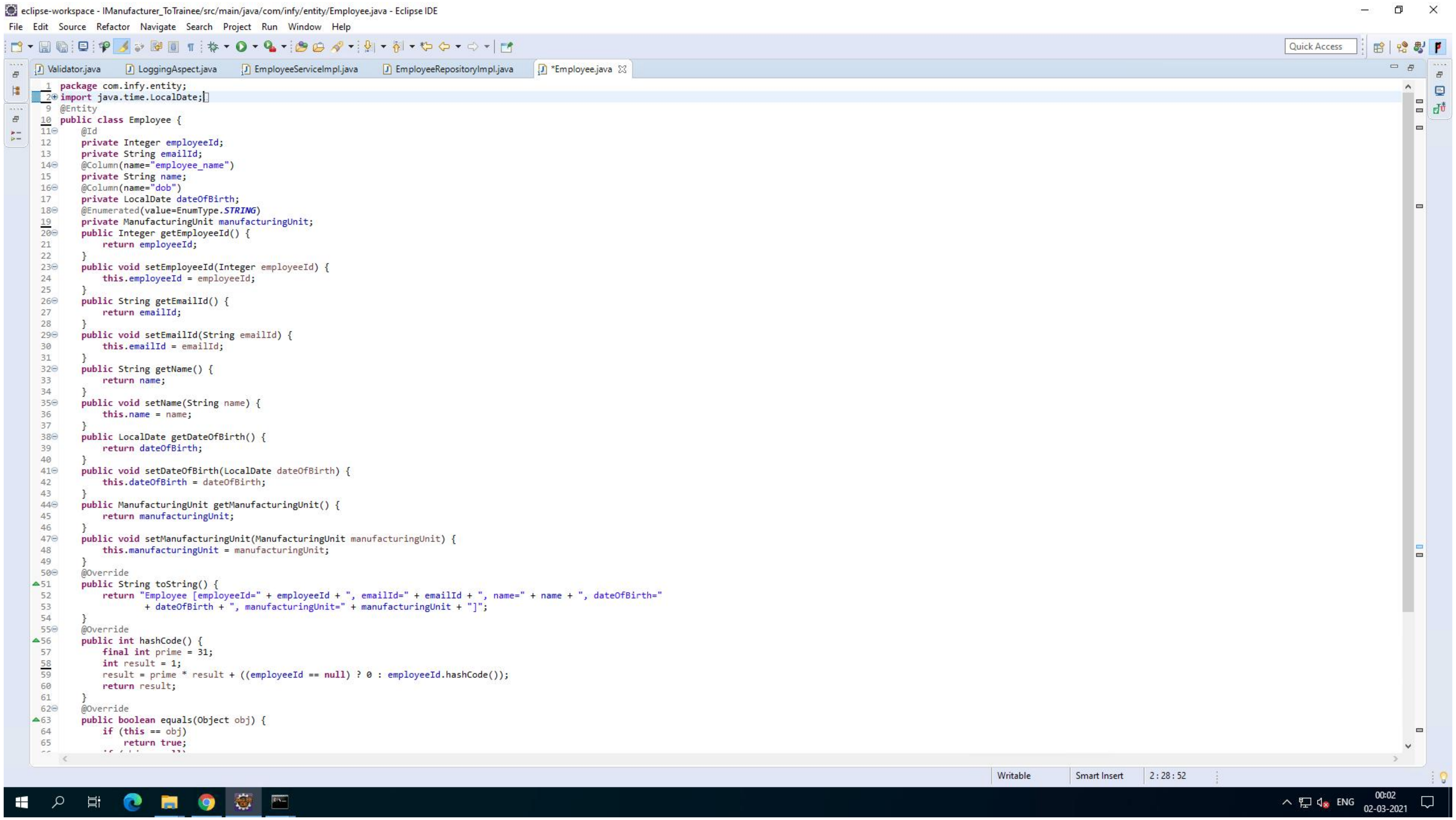
```
Finished after 6.438 seconds

Runs: 4/4        Errors: 0        Failures: 0

IManufacturerToTraineeApplicationTests [Runner: JUnit 5] (0.149 s)
    deleteEmployeeInvalidEmployeeId() (0.118 s)
    updateEmployeeInvalidEmployeeId() (0.015 s)
    addEmployeeInvalidEmailId() (0.000 s)
    getEmployeeInvalidEmployeeId() (0.016 s)
```


eclipse-workspace - IManufacturer_ToTrainee/src/main/java/com/infy/service/EmployeeServiceImpl.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Validator.java
LoggingAspect.java
EmployeeServiceImpl.java
Writable
Smart Insert
11 : 62 : 479
ENG
00:01
02-03-2021

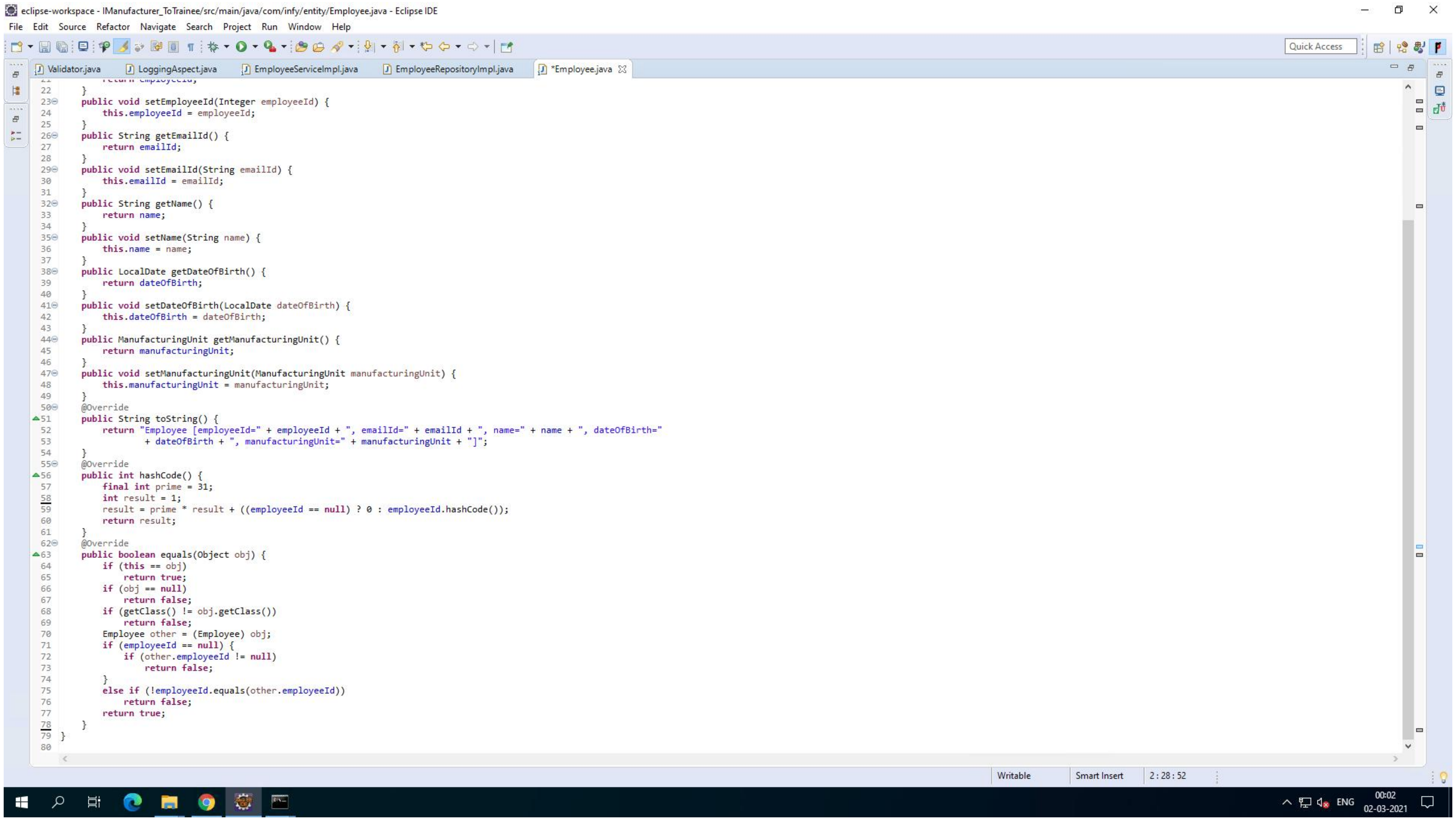
```java
package com.infy.service;
import org.springframework.beans.factory.annotation.Autowired;
@Service(value = "employeeService")
@Transactional
public class EmployeeServiceImpl implements EmployeeService {
    @Autowired
    private EmployeeRepository employeeRepository;
    @Override
    public Integer addEmployee(EmployeeDTO employee) throws InfyEmployeeException {
        Validator.validate(employee);
        EmployeeDTO empdto=new EmployeeDTO();
        empdto=employeeRepository.getEmployeeDetails(employee.getEmployeeId());
        if(empdto==null)
        {
            return employeeRepository.addEmployee(employee);
        }
        else
        {
            throw new InfyEmployeeException("Service.EMPLOYEE_ALREADY_PRESENT");
        }
    }
    @Override
    public EmployeeDTO getEmployeeDetails(Integer employeeId) throws InfyEmployeeException {
        EmployeeDTO empdto = employeeRepository.getEmployeeDetails(employeeId);
        if (empdto == null) {
            throw new InfyEmployeeException("Service.EMPLOYEE_NOT_FOUND");
        }
        return empdto;
    }
    @Override
    public void updateEmployee(Integer employeeId, String emailId) throws InfyEmployeeException {
        EmployeeDTO emp=employeeRepository.getEmployeeDetails(employeeId);
        if(emp==null)
        {
            throw new InfyEmployeeException("Service.EMPLOYEE_NOT_FOUND");
        }
        else
        {
            employeeRepository.updateEmployee(employeeId, emailId);
        }
    }
    @Override
    public void deleteEmployee(Integer employeeId) throws InfyEmployeeException {
        EmployeeDTO empdto = employeeRepository.getEmployeeDetails(employeeId);
        if (empdto == null) {
            throw new InfyEmployeeException("Service.EMPLOYEE_NOT_FOUND");
        }
        else
        {
            employeeRepository.deleteEmployee(employeeId);
        }
    }
}
```


eclipse-workspace - IManufacturer_ToTrainee/src/main/java/com/infy/repository/EmployeeRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Validator.java
LoggingAspect.java
EmployeeServiceImpl.java
EmployeeRepositoryImpl.java
Writable
Smart Insert
1 : 1 : 0
ENG
00:01
02-03-2021

```java
package com.infy.repository;

import javax.persistence.EntityManager;
@Repository(value = "employeeRepository")
public class EmployeeRepositoryImpl implements EmployeeRepository {

    @PersistenceContext
    private EntityManager entityManager;


    @Override
    public Integer addEmployee(EmployeeDTO employee) {
        Employee emp = new Employee();
        emp.setEmployeeId(employee.getEmployeeId());
        emp.setDateOfBirth(employee.getDateOfBirth());
        emp.setEmailId(employee.getEmailId());
        emp.setName(employee.getName());
        emp.setManufacturingUnit(employee.getManufacturingUnit());
        entityManager.persist(emp);
        return emp.getEmployeeId();
    }

    @Override
    public EmployeeDTO getEmployeeDetails(Integer employeeId) {

        EmployeeDTO empdto = null;
        Employee emp = entityManager.find(Employee.class, employeeId);
        if (emp != null) {
            empdto = new EmployeeDTO();
            empdto.setEmployeeId(emp.getEmployeeId());
            empdto.setDateOfBirth(emp.getDateOfBirth());
            empdto.setEmailId(emp.getEmailId());
            empdto.setName(emp.getName());
            empdto.setManufacturingUnit(emp.getManufacturingUnit());
        }
        return empdto;
    }

    @Override
    public void updateEmployee(Integer employeeId, String emailId) {
        Employee emp = entityManager.find(Employee.class, employeeId);
        emp.setEmailId(emailId);
    }

    @Override
    public void deleteEmployee(Integer employeeId) {
        Employee emp = entityManager.find(Employee.class, employeeId);
        entityManager.remove(emp);
    }


}
```


eclipse-workspace - IManufacturer_ToTrainee/src/main/java/com/infy/entity/Employee.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Validator.java
LoggingAspect.java
EmployeeServiceImpl.java
EmployeeRepositoryImpl.java
*Employee.java
Writable
Smart Insert
2 : 28 : 52
ENG
00:02
02-03-2021

```java
package com.infy.entity;
import java.time.LocalDate;
@Entity
public class Employee {
    @Id
    private Integer employeeId;
    private String emailId;
    @Column(name="employee_name")
    private String name;
    @Column(name="dob")
    private LocalDate dateOfBirth;
    @Enumerated(value=EnumType.STRING)
    private ManufacturingUnit manufacturingUnit;
    public Integer getEmployeeId() {
        return employeeId;
    }
    public void setEmployeeId(Integer employeeId) {
        this.employeeId = employeeId;
    }
    public String getEmailId() {
        return emailId;
    }
    public void setEmailId(String emailId) {
        this.emailId = emailId;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public LocalDate getDateOfBirth() {
        return dateOfBirth;
    }
    public void setDateOfBirth(LocalDate dateOfBirth) {
        this.dateOfBirth = dateOfBirth;
    }
    public ManufacturingUnit getManufacturingUnit() {
        return manufacturingUnit;
    }
    public void setManufacturingUnit(ManufacturingUnit manufacturingUnit) {
        this.manufacturingUnit = manufacturingUnit;
    }
    @Override
    public String toString() {
        return "Employee [employeeId=" + employeeId + ", emailId=" + emailId + ", name=" + name + ", dateOfBirth="
                + dateOfBirth + ", manufacturingUnit=" + manufacturingUnit + "]";
    }
    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((employeeId == null) ? 0 : employeeId.hashCode());
        return result;
    }
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
```


eclipse-workspace - IManufacturer_ToTrainee/src/main/java/com/infy/entity/Employee.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Validator.java
LoggingAspect.java
EmployeeServiceImpl.java
EmployeeRepositoryImpl.java
*Employee.java

```java
    }
    public void setEmployeeId(Integer employeeId) {
        this.employeeId = employeeId;
    }
    public String getEmailId() {
        return emailId;
    }
    public void setEmailId(String emailId) {
        this.emailId = emailId;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public LocalDate getDateOfBirth() {
        return dateOfBirth;
    }
    public void setDateOfBirth(LocalDate dateOfBirth) {
        this.dateOfBirth = dateOfBirth;
    }
    public ManufacturingUnit getManufacturingUnit() {
        return manufacturingUnit;
    }
    public void setManufacturingUnit(ManufacturingUnit manufacturingUnit) {
        this.manufacturingUnit = manufacturingUnit;
    }
    @Override
    public String toString() {
        return "Employee [employeeId=" + employeeId + ", emailId=" + emailId + ", name=" + name + ", dateOfBirth="
                + dateOfBirth + ", manufacturingUnit=" + manufacturingUnit + "]";
    }
    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((employeeId == null) ? 0 : employeeId.hashCode());
        return result;
    }
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Employee other = (Employee) obj;
        if (employeeId == null) {
            if (other.employeeId != null)
                return false;
        }
        else if (!employeeId.equals(other.employeeId))
            return false;
        return true;
    }
}
```