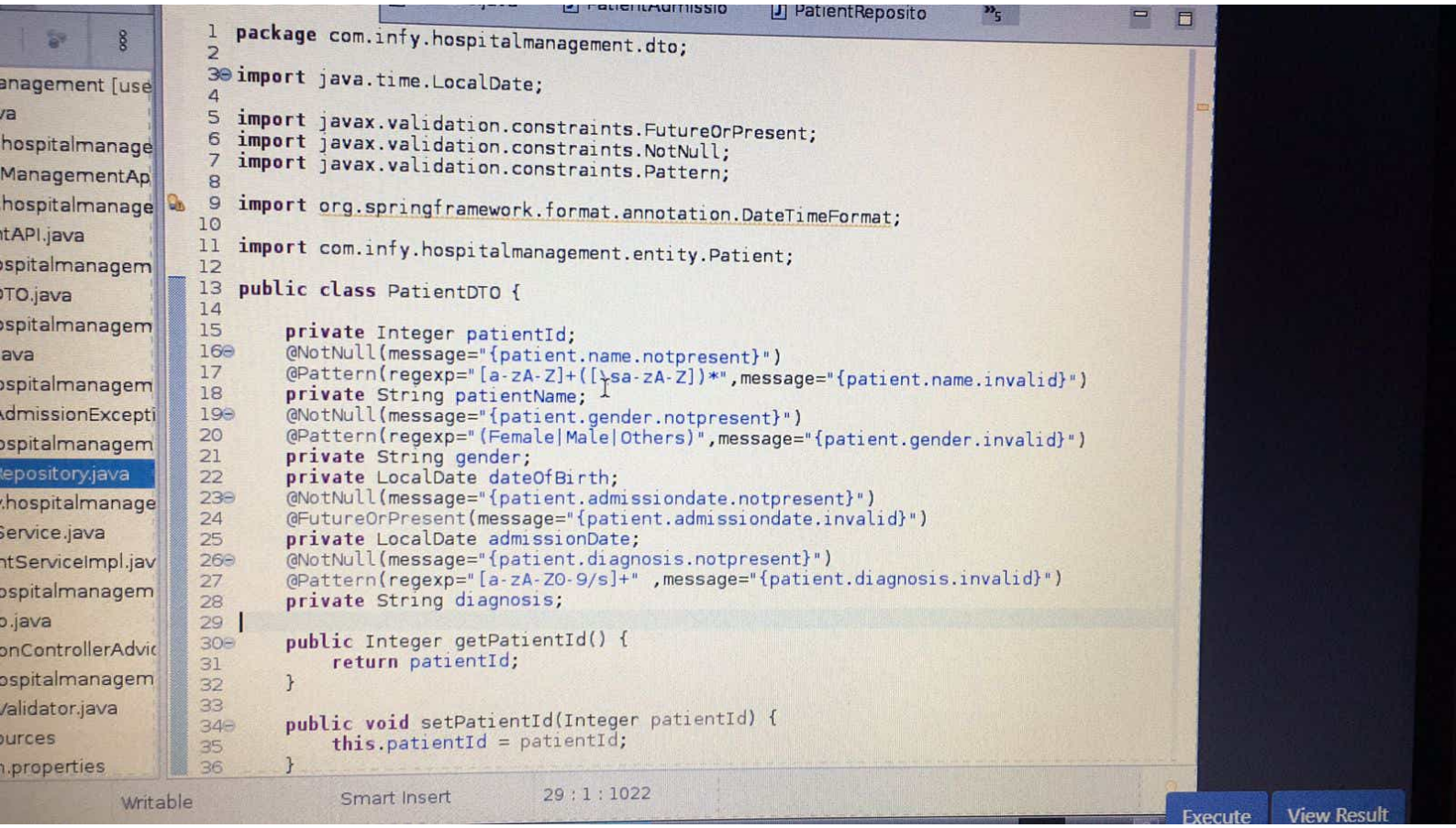
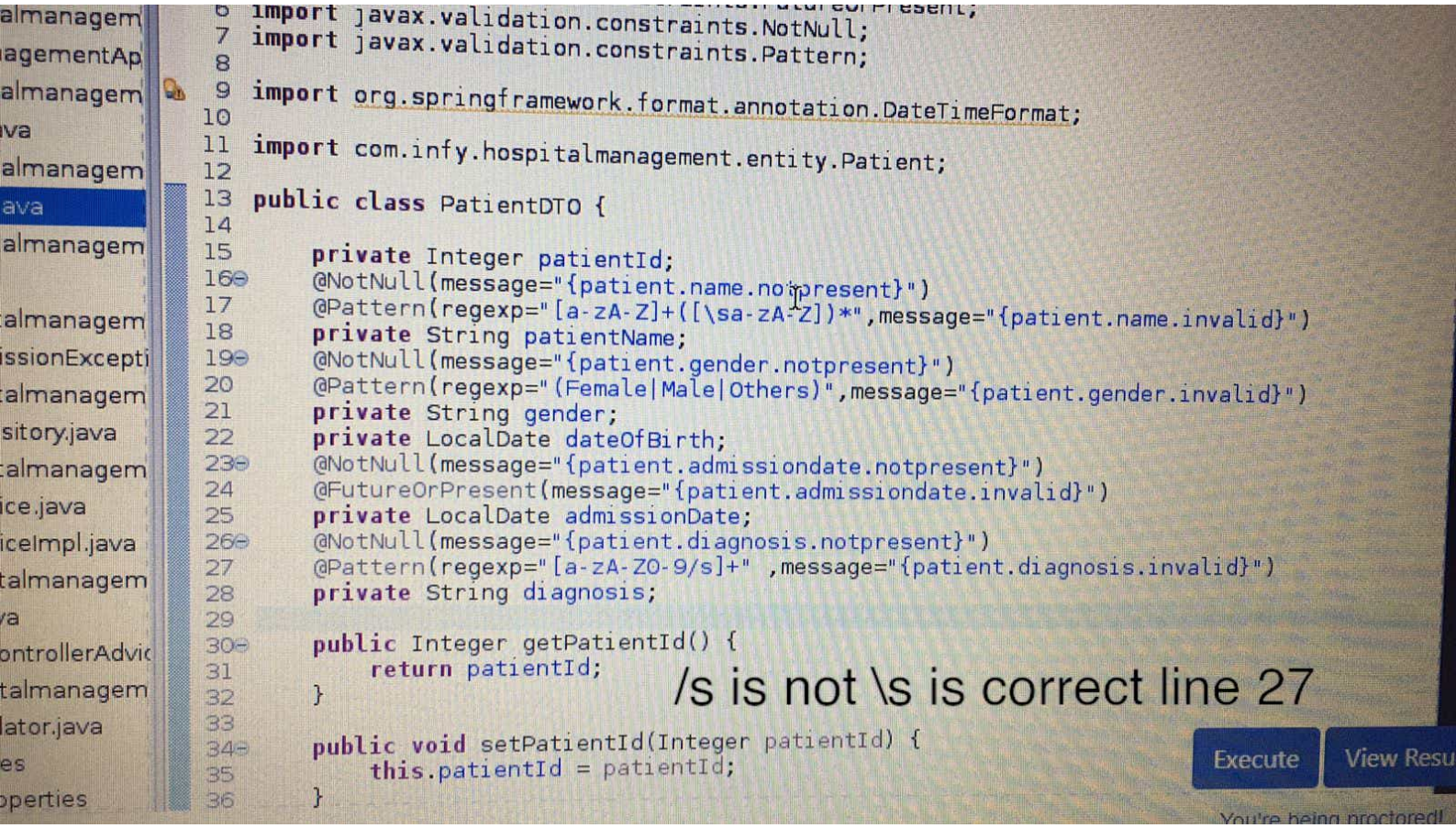
| | | | |
|---|---|---|---|
| PatientAPI-getPatientsByDiagnosis | 1 | 1 | 2 |
| PatientServiceImpl-Annotations | 3 | 0 | 3 |
| PatientServiceImpl-admitPatient | 3 | 0 | 3 |
| PatientServiceImpl-getListOfPatients | 2 | 0 | 2 |
| PatientValidator-isValidDateOfBirth | 4 | 0 | 4 |
| PatientValidator-validatePatient | 0 | 1 | 1 |
| **Total** | **41** | **3** | **44** |

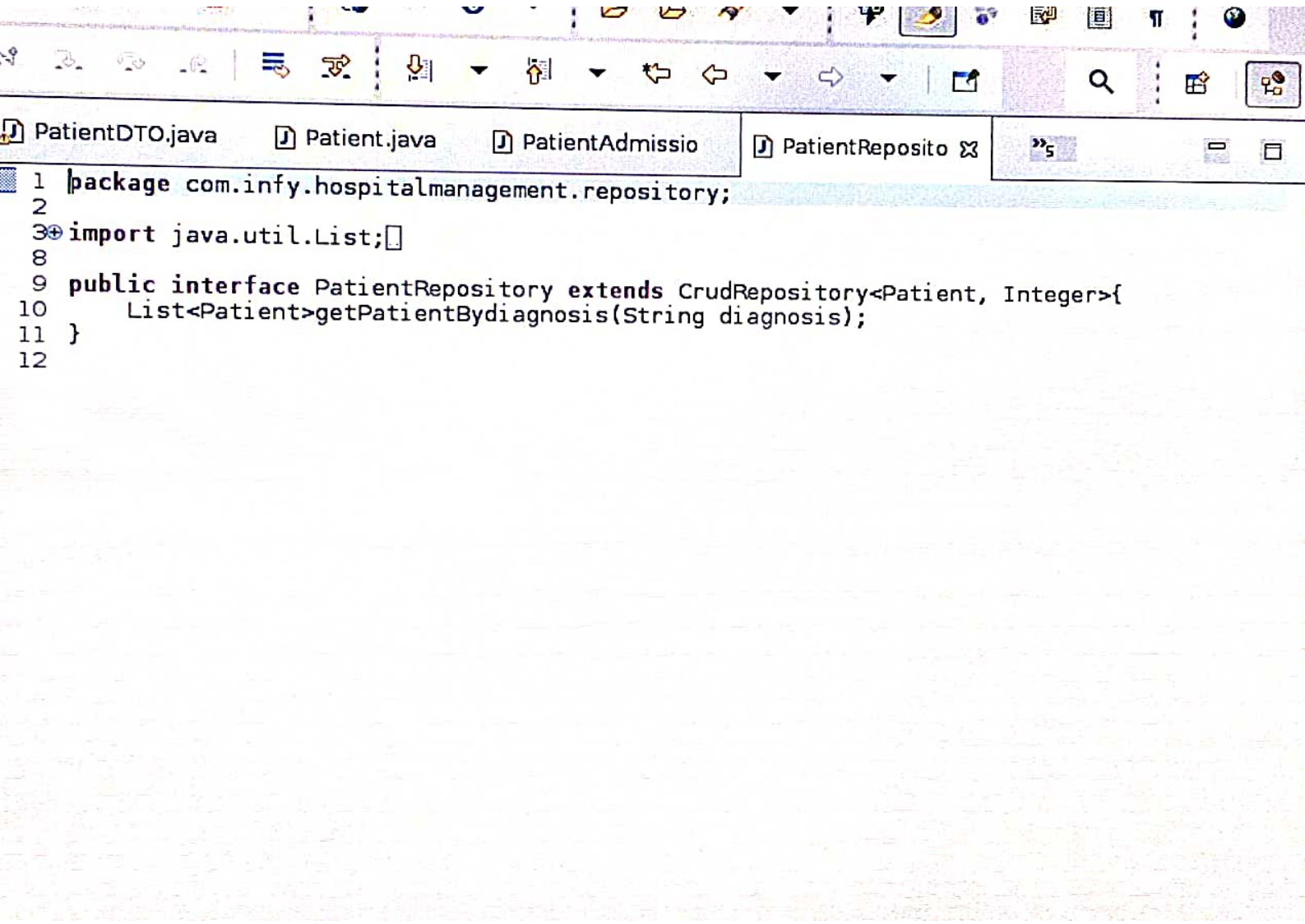
PatientDTO.java | HospitalManagem | PatientAPI.java | Patient.java | »4

```
 1 package com.infy.hospitalmanagement.entity;
 2
 3⊕ import java.time.LocalDate;
10 @Entity
11 @Table(name="patient")
12 public class Patient {
13⊖     @Id
14      @GeneratedValue(strategy=GenerationType.IDENTITY)
15      private Integer patientId;
16      private String patientName;
17      private String gender;
18      private LocalDate dateOfBirth;
19      private LocalDate admissionDate;
20      private String diagnosis;
21
22⊖     public Integer getPatientId() {
23          return patientId;
24      }
25
26⊖     public void setPatientId(Integer patientId) {
27          this.patientId = patientId;
28      }
29
30⊖     public String getPatientName() {
31          return patientName;
32      }
33
```


PatientReposito
package com.infy.hospitalmanagement.dto;
import java.time.LocalDate;
import javax.validation.constraints.FutureOrPresent;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Pattern;
import org.springframework.format.annotation.DateTimeFormat;
import com.infy.hospitalmanagement.entity.Patient;
public class PatientDTO {
private Integer patientId;
@NotNull(message="{patient.name.notpresent}")
@Pattern(regexp="[a-zA-Z]+([\sa-zA-Z])*",message="{patient.name.invalid}")
private String patientName;
@NotNull(message="{patient.gender.notpresent}")
@Pattern(regexp="(Female|Male|Others)",message="{patient.gender.invalid}")
private String gender;
private LocalDate dateOfBirth;
@NotNull(message="{patient.admissiondate.notpresent}")
@FutureOrPresent(message="{patient.admissiondate.invalid}")
private LocalDate admissionDate;
@NotNull(message="{patient.diagnosis.notpresent}")
@Pattern(regexp="[a-zA-Z0-9/s]+" ,message="{patient.diagnosis.invalid}")
private String diagnosis;
public Integer getPatientId() {
return patientId;
}
public void setPatientId(Integer patientId) {
this.patientId = patientId;
}
Writable
Smart Insert
29 : 1 : 1022
Execute
View Result


import javax.validation.constraints.NotNull;
import javax.validation.constraints.Pattern;
import org.springframework.format.annotation.DateTimeFormat;
import com.infy.hospitalmanagement.entity.Patient;
public class PatientDTO {
private Integer patientId;
@NotNull(message="{patient.name.notpresent}")
@Pattern(regexp="[a-zA-Z]+([\sa-zA-Z])*",message="{patient.name.invalid}")
private String patientName;
@NotNull(message="{patient.gender.notpresent}")
@Pattern(regexp="(Female|Male|Others)",message="{patient.gender.invalid}")
private String gender;
private LocalDate dateOfBirth;
@NotNull(message="{patient.admissiondate.notpresent}")
@FutureOrPresent(message="{patient.admissiondate.invalid}")
private LocalDate admissionDate;
@NotNull(message="{patient.diagnosis.notpresent}")
@Pattern(regexp="[a-zA-Z0-9/s]+" ,message="{patient.diagnosis.invalid}")
private String diagnosis;
public Integer getPatientId() {
return patientId;
}
public void setPatientId(Integer patientId) {
this.patientId = patientId;
}
/s is not \s is correct line 27
Execute
View Resu
You're being proctored!

```
import javax.validation.constraints.Pattern;
import org.springframework.format.annotation.DateTimeFormat;

import com.infy.hospitalmanagement.entity.Patient;

public class PatientDTO {

    private Integer patientId;
    @NotNull(message="{patient.name.notpresent}")
    @Pattern(regexp="[a-zA-Z]+([\sa-zA-Z])*",message="{patient.name.invalid}")
    private String patientName;
    @NotNull(message="{patient.gender.notpresent}")
    @Pattern(regexp="(Female|Male|Others)",message="{patient.gender.invalid}")
    private String gender;
    private LocalDate dateOfBirth;
    @NotNull(message="{patient.admissiondate.notpresent}")
    @FutureOrPresent(message="{patient.admissiondate.invalid}")
    private LocalDate admissionDate;
    @NotNull(message="{patient.diagnosis.notpresent}")
    @Pattern(regexp="[a-zA-Z0-9/s]+" ,message="{patient.diagnosis.invalid}")
    private String diagnosis;

    public Integer getPatientId() {
        return patientId;
    }

    public void setPatientId(Integer patientId) {
        this.patientId = patientId;
    }

    public String getPatientName() {
```


PatientDTO.java
Patient.java
PatientAdmissio
PatientReposito
1 package com.infy.hospitalmanagement.repository;
2
3 import java.util.List;
8
9 public interface PatientRepository extends CrudRepository<Patient, Integer>{
10 List<Patient>getPatientBydiagnosis(String diagnosis);
11 }
12

PatientValidato | HospitalManagem | PatientAPI.java | ExceptionContro

```
 1 package com.infy.hospitalmanagement.api;
 2
 3 import java.util.List;
20 @RestController
21 @RequestMapping(value="/api")
22 @Validated
23 public class PatientAPI {
24 @Autowired
25     private PatientService patientService;
26 @GetMapping(value="/patients/{diagnosis}")
27
28     public ResponseEntity<List<PatientDTO>> getPatientsByDiagnosis(@PathVariable
29             @Pattern(regexp="[a-zA-Z0-9\s]+",message="{Patient.Diagnosis.Invalid}")
30     String diagnosis)
31             throws PatientAdmissionException {
32         List<PatientDTO>patientDTO=patientService.getListOfPatients(diagnosis);
33         return new ResponseEntity<>(patientDTO,HttpStatus.OK);
34     }
35
36     @PostMapping(value="/patients")
37     public ResponseEntity<PatientDTO> admitPatient(PatientDTO patientDTO)
38             throws PatientAdmissionException {
39         PatientDTO patientDTO1=patientService.admitPatient(patientDTO);
40         return new ResponseEntity<>(patientDTO1,HttpStatus.CREATED);
41     }
42 }
43
```

```
 1 package com.infy.hospitalmanagement.utility;
 2
 3⊕import java.util.stream.Collectors;
20
21 @RestControllerAdvice
22 public class ExceptionControllerAdvice {
23
24     private static final Log LOGGER = LogFactory.getLog(ExceptionControllerAdvice.class)
25⊖@Autowired
26     private Environment environment;
27⊖    @ExceptionHandler(PatientAdmissionException.class)
28     public ResponseEntity<ErrorInfo> patientAdmissionExceptionHandler(PatientAdmissionE
29         LOGGER.error(exception.getMessage(), exception);
30         ErrorInfo errorInfo = new ErrorInfo();
31         errorInfo.setErrorCode(HttpStatus.BAD_REQUEST.value());
32         errorInfo.setErrorMessage(environment.getProperty(exception.getMessage()));
33         return new ResponseEntity<>(errorInfo, HttpStatus.BAD_REQUEST);
34     }
35⊖    @ExceptionHandler(Exception.class)
36     public ResponseEntity<ErrorInfo> generalExceptionHandler(Exception exception) {
37         LOGGER.error(exception.getMessage(), exception);
38         ErrorInfo errorInfo = new ErrorInfo();
39         errorInfo.setErrorCode(HttpStatus.INTERNAL_SERVER_ERROR.value());
40         errorInfo.setErrorMessage(environment.getProperty("General.EXCEPTION_MESSAGE"));
41         return new ResponseEntity<>(errorInfo, HttpStatus.INTERNAL_SERVER_ERROR);
42     }
43⊖    @ExceptionHandler({MethodArgumentNotValidException.class,ConstraintViolationExceptic
44     public ResponseEntity<ErrorInfo> validatorExceptionHandler(Exception exception) {
45         LOGGER.error(exception.getMessage(), exception);
46         String errorMsg;
47         if (exception instanceof MethodArgumentNotValidException) {
48             MethodArgumentNotValidException manvException = (MethodArgumentNotValidExcep
49             errorMsg = manvException.getBindingResult().getAllErrors().stream().map(Obje
50                     .collect(Collectors.joining(", "));
51         } else {
```

```java
package com.infy.hospitalmanagement.validator;

import java.time.LocalDate;

public class PatientValidator {


    public static void validatePatient(PatientDTO patientDTO)
            throws PatientAdmissionException
    {
        if(! isValidDateOfBirth(patientDTO.getDateOfBirth())) {
            throw new PatientAdmissionException("PatientValidator.INVALID_DOB");
        }

    }

    public static Boolean isValidDateOfBirth(LocalDate dateOfBirth) {
        if(dateOfBirth.isAfter(LocalDate.now())) {
            return false;
        }
        else if(LocalDate.now().getYear()-dateOfBirth.getYear()>100) {
            return false;
        }else {
            return true;
        }

    }
}
```

```java
package com.infy.hospitalmanagement.service;

import java.util.ArrayList;
@Service(value="patientService")
@Transactional
public class PatientServiceImpl implements PatientService{
    @Autowired
    private PatientRepository patientRepository;

    @Override
    public List<PatientDTO> getListOfPatients(String diagnosis)
            throws PatientAdmissionException
    {
        List<Patient>patient=patientRepository.getPatientBydiagnosis(diagnosis);
        if(patient.isEmpty()||patient.size()==0)
        {
            throw new PatientAdmissionException("PatientService.PATIENT_UNAVAILABLE");
        }
        List<PatientDTO>dtos=new ArrayList<>();
        patient.stream().forEach(x->dtos.add(new PatientDTO().prepareDTO(x)));
        dtos.sort((x1,x2)->x1.getAdmissionDate().compareTo(x2.getAdmissionDate()));
        return dtos;
    }

    @Override
    public PatientDTO admitPatient(PatientDTO patientDTO)
            throws PatientAdmissionException
    {
        PatientValidator.validatePatient(patientDTO);
        Patient patient=PatientDTO.prepareEntity(patientDTO);
        patient=patientRepository.save(patient);
        patientDTO.setPatientId(patient.getPatientId());
        return patientDTO;
    }

}
```