Imagina que na nossa classe teriamos que usar o Get, Set, ToString... com o Lombok fica mais facil.

package com.allephnogueira.api.medico;  
  
import com.allephnogueira.api.endereco.Endereco;  
import jakarta.persistence.\*;  
import lombok.AllArgsConstructor;  
import lombok.EqualsAndHashCode;  
import lombok.Getter;  
import lombok.NoArgsConstructor;  
  
@Table (name = "medicos") // Alterando o nome da tabela  
@Entity (name = "Medico") // Alterando nome do banco  
**@Getter // Aqui vai gerar os metodos getter  
@NoArgsConstructor // Aqui vamos gerar o construtor sem argumentos  
@AllArgsConstructor // Vai ter um construtor que recebe todos os campos  
@EqualsAndHashCode(of = "id") // Pesquisar mais**  
public class Medico {  
 @Id  
 @GeneratedValue (strategy = GenerationType.*IDENTITY*)  
 private Long id;  
 private String nome;  
 private String email;  
 private String crm;  
 @Enumerated (EnumType.*STRING*)  
 private Especialidade especialidade;  
 @Embedded // Aqui estamos dizendo para ele ficar em uma classe separada, mas ele vai ficar no mesmo banco de dados, fazendo parte da tabela medicos.  
 // Isso é somente para por o codigo mais organizado.  
 // Para isso funcionar também devemos por essa anotação na classe que vamos usar no caso ENDERECO  
 private Endereco endereco;  
}

package com.allephnogueira.api.endereco;  
  
  
import jakarta.persistence.Embeddable;  
  
import lombok.AllArgsConstructor;  
import lombok.Getter;  
import lombok.NoArgsConstructor;  
  
@Embeddable // Agora precisamos declarar aqui também para ele reconhecer.  
  
**@Getter // Aqui vai gerar os metodos getter  
@NoArgsConstructor // Aqui vamos gerar o construtor sem argumentos  
@AllArgsConstructor // Vai ter um construtor que recebe todos os campos**  
  
  
public class Endereco {  
 private String logradouro;  
 private String bairro;  
 private String cep;  
 private String numero;  
 private String complemento;  
 private String cidade;  
 private String uf;  
}