Builder Pattern là một trong những mẫu thiết kế trong lập trình hướng đối tượng, được sử dụng để xây dựng các đối tượng phức tạp bằng cách xây dựng các đối tượng từ các đối tượng cơ bản và các bước xử lý khác nhau.

Mô hình Builder Pattern bao gồm một lớp Builder, một lớp Product (đối tượng được xây dựng), và các phương thức để thiết lập các thuộc tính của đối tượng Product. Builder Pattern cho phép chúng ta tạo ra các đối tượng phức tạp bằng cách tách chúng thành các phần nhỏ hơn và xây dựng từng phần một, điều này giúp tăng tính linh hoạt và dễ bảo trì của chương trình.

Ví dụ về Builder Pattern trong mô hình bệnh viện:

Giả sử rằng ta đang phát triển một ứng dụng quản lý bệnh viện, và bạn muốn xây dựng một đối tượng Patient (bệnh nhân) với các thuộc tính như sau:

* Tên bệnh nhân
* Tuổi bệnh nhân
* Giới tính bệnh nhân
* Bệnh án của bệnh nhân

Bây giờ, để xây dựng một đối tượng Patient phức tạp, ta có thể sử dụng mô hình Builder Pattern như sau:

Đầu tiên, ta sẽ tạo một lớp PatientBuilder để xây dựng đối tượng Patient.

Lớp PatientBuilder sẽ có các phương thức để thiết lập các thuộc tính của đối tượng Patient. Ví dụ: setName(), setAge(), setGender(), setMedicalRecord()

Sau đó, ta sẽ tạo một đối tượng PatientBuilder và sử dụng các phương thức của nó để thiết lập các thuộc tính của đối tượng Patient.

Cuối cùng, ta sẽ sử dụng phương thức build() của lớp PatientBuilder để xây dựng đối tượng Patient từ các thuộc tính đã thiết lập.

public class Patient {  
 private String name;  
 private int age;  
 private String gender;  
 private String medicalRecord;  
  
 private Patient(PatientBuilder builder) {  
 this.name = builder.name;  
 this.age = builder.age;  
 this.gender = builder.gender;  
 this.medicalRecord = builder.medicalRecord;  
 }  
  
 public static class PatientBuilder {  
 private String name;  
 private int age;  
 private String gender;  
 private String medicalRecord;  
  
 public PatientBuilder setName(String name) {  
 this.name = name;  
 return this;  
 }  
  
 public PatientBuilder setAge(int age) {  
 this.age = age;  
 return this;  
 }  
  
 public PatientBuilder setGender(String gender) {  
 this.gender = gender;  
 return this;  
 }  
  
 public PatientBuilder setMedicalRecord(String medicalRecord) {  
 this.medicalRecord = medicalRecord;  
 return this;  
 }  
  
 public Patient build() {  
 return new Patient(this);  
 }  
 }  
}

Trong đoạn mã trên:

* Lớp Patient có các thuộc tính tên, tuổi, giới tính và bệnh án, và một phương thức khởi tạo private. Điều này đảm bảo rằng đối tượng Patient chỉ có thể được xây dựng thông qua lớp PatientBuilder.
* Lớp PatientBuilder có các phương thức để thiết lập các thuộc tính của đối tượng Patient. Các phương thức này đều trả về chính đối tượng PatientBuilder, cho phép chúng ta gọi các phương thức liên tiếp nhau.
* Phương thức build() của lớp PatientBuilder được sử dụng để xây dựng đối tượng Patient từ các thuộc tính đã thiết lập.

Ta sẽ sử dụng Builder Pattern để xây dựng đối tượng Patient:

public class Main {  
 public static void main(String[] args) {  
 Patient patient = new Patient.PatientBuilder()  
 .setName("John")  
 .setAge(23)  
 .setGender("Male")  
 .setMedicalRecord("High blood pressure")  
 .build();  
 System.*out*.println(patient);  
 }  
}