|  |  |
| --- | --- |
| logo-dai-hoc-gia-dinh | **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC GIA ĐỊNH**  **KHOA CÔNG NGHỆ THÔNG TIN** |

**BÁO CÁO**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG NÂNG CAO**

**TÊN ĐỀ TÀI**

**STATE PATTERN**

**Giảng viên hướng dẫn: Th.s LÊ HUỲNH PHƯỚC**

**Sinh viên thực hiện: VÕ VĂN THÁI………………….….MSSV: 2004110036**

**HUỲNH NGUYỄN MINH THIỆN .MSSV: 2008110318**

**NGUYỄN VĂN HOÀNG ………….MSSV: 2010110039**

**HỒ NGUYỄN THANH THẮNG… MSSV: 2008110234**

**Lớp: K14DCPM02**

*Học kỳ 3, ngày 22 tháng 08 năm 2022*

Lời mở đầu

Lập trình theo phương pháp hướng đối tượng là một phương pháp lập trình tương đối mới (xuất hiện từ những năm 1990) và được hầu hết các ngôn ngữ lập trình hiện nay hỗ trợ. Giáo trình này sẽ giới thiệu các đặc trưng của phương pháp lập trình hướng đối tượng như tính đóng gói, tính kế thừa và tính đa hình. Chúng tôi chọn ngôn ngữ java để minh họa, vì đây là ngôn ngữ lập trình hướng đối tượng dễ học và phổ dụng nhất hiện nay. Sau khi hoàn tất giáo trình này, sinh viên sẽ biết được cách mô hình hóa các lớp đối tượng trong thế giới thực thành các lớp đối tượng trong java và cách phối hợp các đối tượng này để giải quyết vấn đề đang quan tâm.

Trước khi tìm hiểu chi tiết về phương pháp lập trình hướng đối tượng, sinh viên nên đọc trước phần phụ lục cơ bản về ngôn ngữ java để làm quen với các kiểu dữ liệu, các cấu trúc điều khiển trong ngôn ngữ java. Sau khi đã nắm bắt được phương pháp lập trình hướng đối tượng.

**Mục lục**

[I. Khái niệm State pattern. 3](#_Toc31848)

[1.State Pattern là gì? 3](#_Toc20641)

[2.Sơ đồ UML của State Pattern. 3](#_Toc23196)

[3.Lợi ích của State Pattern là gì? 4](#_Toc5866)

[4.Sử dụng State Pattern khi nào? 4](#_Toc18612)

[II. Code minh họa. 5](#_Toc24296)

[1. Gumball Machine 5](#_Toc14977)

[2. Thiết kế mới với State Pattern 7](#_Toc31676)

[III . Tổng kết. 9](#_Toc20117)

[1. Kết quả đạt được 10](#_Toc32030)

[2. Thuận lợi và khó khăn 10](#_Toc11744)

*[Lời cảm ơn](#_Toc23844)* [10](#_Toc23844)

1. **Khái niệm State pattern.**

**1.State Pattern là gì?**

State Pattern là một trong những Pattern thuộc nhóm hành vi (Behavior Pattern). Behavioral Pattern : Các Design Pattern thuộc nhóm này thiết kế dựa trên khả năng tương tác, giao tiếp của 1 class với các thành phần khác.

Nó cho phép một đối tượng thay đổi hành vi của nó khi trạng thái hiện tại của nó thay đổi. Điều này được thực hiện bằng cách cung cấp cho mỗi object này một hoặc nhiều object state.

**2.Sơ đồ UML của State Pattern.**



**Các thành phần tham gia State Pattern:**

- Context là một class có code phụ thuộc với state

- State : là một interface hoặc abstract class xác định các đặc tính cơ bản của tất cả các đối tượng ConcreteState.

- ConcreteState : là lớp cụ thể của state ứng với từng trạng thái

**Một vài điểm cần ghi nhớ khi áp dụng pattern này:**

- Một đối tượng nên thay đổi hành vi của nó khi trạng thái bên trong của nó thay đổi.

- Mỗi State nên được xác định độc lập.

- Thêm các trạng thái mới sẽ không làm ảnh hưởng đến các trạng thái hoặc chức năng khác.

**3.Lợi ích của State Pattern là gì?**

**Lợi ích:**

- Đảm bảo nguyên tắc Single responsibility principle (SRP) : tách biệt mỗi State tương ứng với 1 class riêng biệt.

- Đảm bảo nguyên tắc Open/Closed Principle (OCP) : chúng ta có thể thêm một State mới mà không ảnh hưởng đến State khác hay Context hiện có.

- Giữ hành vi cụ thể tương ứng với trạng thái.

- Giúp chuyển trạng thái một cách rõ ràng.

**Nhược điểm**

- Việc sử dụng state pattern có thể quá mức cần thiết nếu chỉ có một vài trạng thái hoặc hiếm khi thay đổi có thể dẫn đến việc tăng độ phức tạp của code

**4.Sử dụng State Pattern khi nào?**

Sử dụng State pattern khi bạn có một object hoạt động khác nhau tùy thuộc vào trạng thái hiện tại của nó, số lượng trạng thái là rất lớn và code của trạng thái cụ thể thường xuyên thay đổi.

Khi cần thay đổi hành vi object dựa trên trạng thái object.

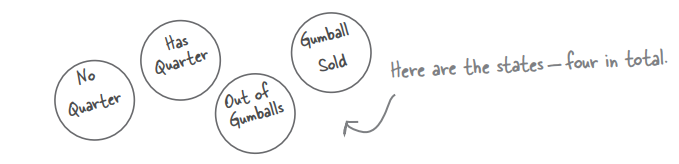
**5. Strategy Pattern vs State Pattern.**

State Pattern gần giống Strategy Pattern, chuyển đổi các trạng thái thông qua các phương thức được định nghĩa trong interface. Tuy nhiên State Pattern không hạn chế sự phụ thuộc giữa các trạng thái cụ thể, cho phép chúng thay đổi trạng thái của ngữ cảnh theo ý muốn.

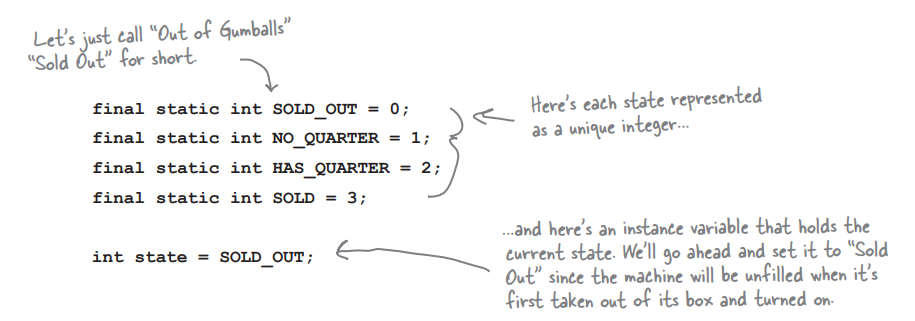
**II. Code minh họa.**

## ****1. Gumball Machine****

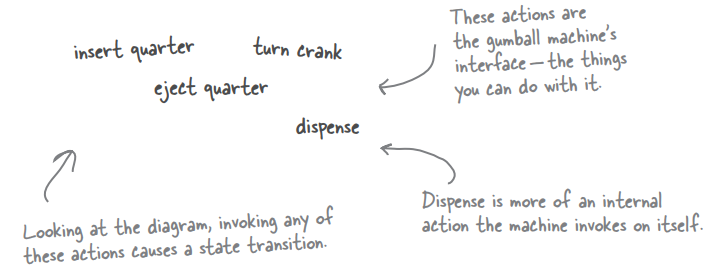
****Đầu tiên, tập hợp các trạng thái :****



****Tiếp theo, tạo một biến đối tượng để giữ trạng thái hiện tại và xác định các giá trị cho từng trạng thái:****



****Tập hợp các hành động có thể xảy ra trong hệ thống:****





## 

## ****Các trạng thái hiện tại rất lộn xộn****

**Điều nào sau đây mô tả trạng thái hiện tại của project trên?** **(Chọn tất cả các đáp án đúng.)**

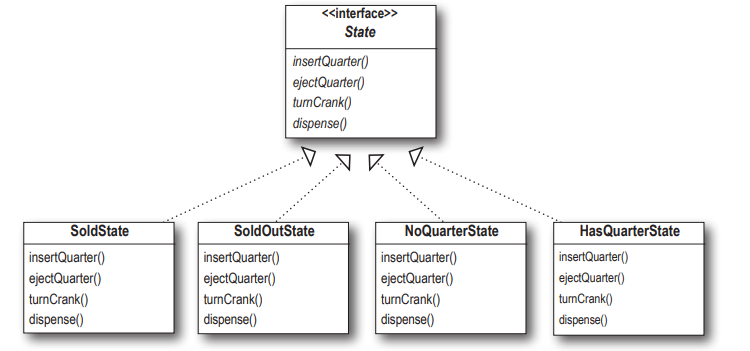
* ❏ A. Code này chắc chắn không tuân thủ Nguyên tắc Đóng mở.
* ❏ B. Code này sẽ làm cho một lập trình viên FORTRAN(một ngôn ngữ cũ) tự hào.
* ❏ C. Thiết kế này không phải là hướng đối tượng.
* ❏ C. Chuyển trạng thái không rõ ràng.
* ❏ D. Ở đây, húng ta chưa đóng gói bất cứ những gì khác biệt.
* ❏ E. Các bổ sung khác có khả năng gây ra lỗi trong code làm việc.

****Đáp án: A,B,C,D,E****

## ****Thiết kế mới với State Pattern****

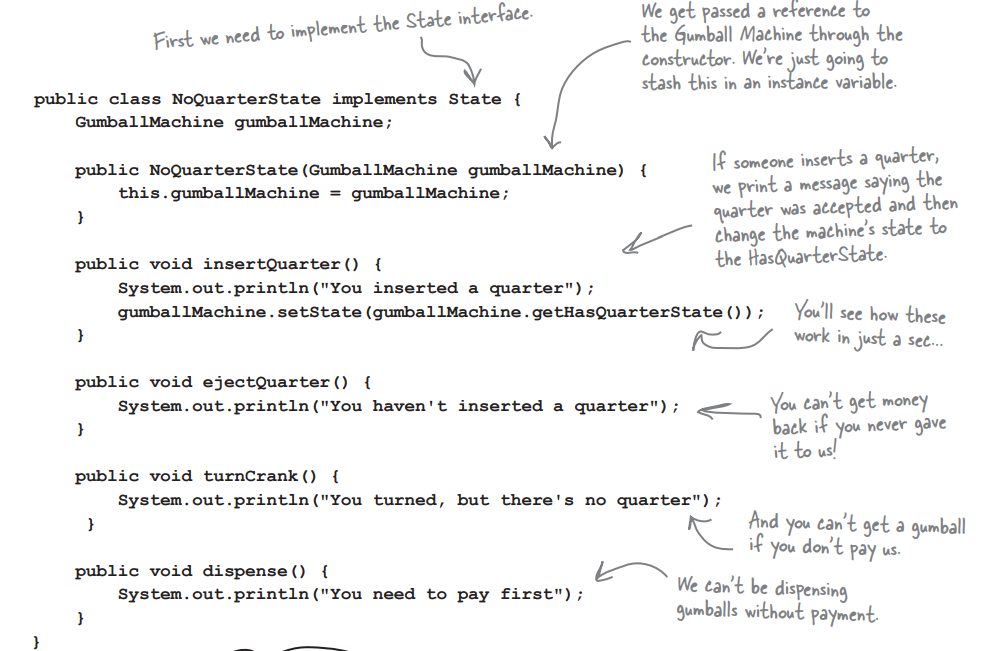
1. Đầu tiên, chúng ta sẽ khởi tạo một State interface có chứa các phương thức cho mọi hành động trong Gumball Machine.
2. Sau đó, chúng ta sẽ triển khai lớp State cho từng trạng thái của máy. Các lớp này sẽ chịu trách nhiệm cho hành vi của máy khi nó ở trạng thái tương ứng.
3. Cuối cùng, chúng ta sẽ loại bỏ tất cả các code điều kiện (if) và thay vào đó ủy thác cho lớp State để thực hiện công việc .

## ****Định nghĩa interface và class của State Pattern****



## 

## ****Implementing State classes****



**III . Tổng kết.**

1. **Kết quả đạt được**

Sau một thời gian học tập và nghiên cứu , nhóm em đã hoàn thành báo cáo đúng tiến độ do Thầy đề ra , hiểu rõ được đề tài và xác định được đúng yêu cầu của đề tài.

Tuy nhiên do thời gian và kiến thức có hạn không thể tránh khỏi những thiếu sót . Nhóm em rất mong được sự quan tâm và chỉ bảo của Thầy và các bạn

1. **Thuận lợi và khó khăn**

**- Thuận lợi**

\* Được sự hướng dẫn tận tình của GVHD Lê Huỳnh Phước

\*Được sự giúp đỡ của quý Thầy, Cô trong khoa CNTT.

\* Được sự quan tâm của BGH Trường Đại Học Gia Định.

**- Khó khăn**

\* Do thời gian còn hạn chế vì phải thực hiện trong lúc ôn thi nhiều môn khác

***Lời cảm ơn***

Sau thời gian học tập dưới mái trường Đại Học Gia Định, được sự truyền đạt kiến thức và giúp đỡ tận tình của quý thầy cô giảng viên là hành trang quý báu cho sự nhận thức và hiểu biết của chúng em ngày hôm nay. Chúng em xin tỏ lòng biết ơn chân thành nhất đối với tất cả các thầy cô giảng viên và đặc biệt là thầy Lê Huỳnh Phước, người đã tận tình hướng dẫn chúng em trong học tập.

Do kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ chưa hoàn hảo nên bài báo cáo sẽ còn nhiều thiếu sót, kính mong sự góp ý và giúp đỡ từ quý thầy cô. Chúng em xin chân thành cảm ơn!