TRƯỜNG ĐẠI HỌC GIA ĐỊNH

KHOA CÔNG NGHỆ THÔNG TIN

\*\*\*\*\*

**ĐỒ ÁN MÔN HỌC**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG NÂNG CAO**

**FACTORY PATTERN**

GVHD: Ths. LÊ HUỲNH PHƯỚC

SVTH: TRƯƠNG CAO ANH HUY 2008110179 –14DCPM01

LÊ TRẦN THÀNH LONG 2008110275 –14DCPM01

NGUYỄN ĐÌNH ĐẠT 2008110117 –14DCPM01

BÙI QUỐC KHÁNH 1731102060 – 14DCPM01

Tháng 8 – Năm 2022

**LỜI CẢM ƠN**

*Xin gửi lời cảm ơn chân thành đến ban giám hiệu trường Đại học Gia Định đã tạo điều kiện cho em thực hiện đồ án này. Trong quá trình học tập và thực hiện báo cáo đồ án, em đã học hỏi được thêm nhiều điều giúp ích cho chuyên ngành của mình sau này.*

*Xin gửi lời cảm ơn thầy Lê Huỳnh Phước đã truyền đạt kiến thức, tận tâm chỉ dạy trong quá trình học tập tại trường và đưa ra đồ án này để em có thể học hỏi được nhiều điều.*

*Chân thành cảm ơn.*

**ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN**

*………………, ngày ……… tháng ……… năm …………*

**Giảng viên hướng dẫn**

(Ký tên, ghi rõ họ tên)

Mục lục

[I. Giới thiệu chung: 1](#_Toc112391794)

[1. Lập trình hướng đối tượng (OOP) là gì? 1](#_Toc112391795)

[2. Java Swing là gì? 4](#_Toc112391796)

[3. Creational Design Pattern là gì ? 5](#_Toc112391797)

[4. Factory Pattern là gì ? 7](#_Toc112391798)

[II. Công dụng của Factory Pattern: 8](#_Toc112391799)

[1. Lợi ích của Factory Pattern: 8](#_Toc112391800)

[2. Ưu điểm và nhược điểm của Factory Method Pattern: 9](#_Toc112391801)

[III. Khởi tạo Factory Pattern: 9](#_Toc112391802)

[1. Factory Pattern sử dụng khi nào ? 10](#_Toc112391803)

[2. Khởi tạo: 10](#_Toc112391804)

[IV. Ví dụ về mẫu Factory Pattern: 11](#_Toc112391805)

[1. Đề bài: 11](#_Toc112391806)

[2. Thực thi: 11](#_Toc112391807)

[V. Tài liệu tham khảo: 16](#_Toc112391808)

1. **Giới thiệu chung:**
2. **Lập trình hướng đối tượng (OOP) là gì?**

* OOP (viết tắt của Object Oriented Programming) – lập trình hướng đối tượng là một phương pháp lập trình dựa trên khái niệm về lớp và đối tượng. OOP tập trung vào các đối tượng thao tác hơn là logic để thao tác chúng.
* OOP là nền tảng của các design pattern hiện nay.
* Mục tiêu của OOP là tối ưu việc quản lý source code, giúp tăng khả năng tái sử dụng và quan trọng hơn hết là giúp tóm gọn các thủ tục đã biết trước tính chất thông qua việc sử dụng các đối tượng.
* Trong OOP bao gồm:
* **Đối tượng (Object)**
* Đối tượng trong OOP bao gồm 2 thành phần chính:
* Thuộc tính (Attribute): là những thông tin, đặc điểm của đối tượng
* Phương thức (Method): là những hành vi mà đối tượng có thể thực hiện
* Để dễ hình dung, ta có một ví dụ thực tế về đối tượng là smartphone. Đối tượng này sẽ có:
* Thuộc tính: màu sắc, bộ nhớ, hệ điều hành…
* Phương thức: gọi điện, chụp ảnh, nhắn tin, ghi âm…
* **Lớp (Class)**
* Lớp là sự trừu tượng hóa của đối tượng. Những đối tượng có những đặc tính tương tự nhau sẽ được tập hợp thành một lớp. Lớp cũng sẽ bao gồm 2 thông tin là thuộc tính và phương thức.
* Một đối tượng sẽ được xem là một thực thể của lớp.
* Tiếp nối ví dụ ở phần đối tượng (object) phía trên, ta có lớp (class) smartphone gồm 2 thành phần:
* Thuộc tính: màu sắc, bộ nhớ, hệ điều hành…
* Phương thức: gọi điện, chụp ảnh, nhắn tin, ghi âm…
* Các đối tượng của lớp này có thể là: iPhone, Samsung, Oppo, Huawei…

**4 đặc tính cơ bản của OOP:**

**Tính đóng gói (Encapsulation)** cho phép che giấu thông tin và những tính chất xử lý bên trong của đối tượng. Các đối tượng khác không thể tác động trực tiếp đến dữ liệu bên trong và làm thay đổi trạng thái của đối tượng mà bắt buộc phải thông qua các phương thức công khai do đối tượng đó cung cấp. Tính chất này giúp tăng tính bảo mật cho đối tượng và tránh tình trạng dữ liệu bị hư hỏng ngoài ý muốn.

**Tính kế thừa (Inheritance),** Đây là tính chất được sử dụng khá nhiều. Tính kế thừa cho phép xây dựng một lớp mới (lớp Con), kế thừa và tái sử dụng các thuộc tính, phương thức dựa trên lớp cũ (lớp Cha) đã có trước đó. Các lớp Con kế thừa toàn bộ thành phần của lớp Cha và không cần phải định nghĩa lại. Lớp Con có thể mở rộng các thành phần kế thừa hoặc bổ sung những thành phần mới. Ví dụ:

Lớp Cha là smartphone, có các thuộc tính: màu sắc, bộ nhớ, hệ điều hành…

Các lớp Con là iPhone, Samsung, Oppo cũng có các thuộc tính: màu sắc, bộ nhớ, hệ điều hành…

**Tính đa hình (Polymorphism)** trong lập trình OOP cho phép các đối tượng khác nhau thực thi chức năng giống nhau theo những cách khác nhau. Ví dụ:

Ở lớp smartphone, mỗi một dòng máy đều kế thừa các thành phần của lớp cha nhưng iPhone chạy trên hệ điều hành iOS, còn Samsung lại chạy trên hệ điều hành Android.

Chó và mèo cùng nghe mệnh lệnh “kêu đi” từ người chủ. Chó sẽ “gâu gâu” còn mèo lại kêu “meo meo”.

**Tính trừu tượng (Abstraction)** giúp loại bỏ những thứ phức tạp, không cần thiết của đối tượng và chỉ tập trung vào những gì cốt lõi, quan trọng. Ví dụ: Quản lý nhân viên thì chỉ cần quan tâm đến những thông tin như:

* Họ tên
* Ngày sinh
* Giới tính

Chứ không cần phải quản lý thêm thông tin về:

* Chiều cao
* Cân nặng
* Sở thích
* Màu da
* …

## **Java Swing là gì?**

* **Java Swing** là cách gọi rút gọn khi người ta nhắc đến Swing của [Java Foundation](https://en.wikipedia.org/wiki/Java_Foundation_Classes) (JFC). Nó là bộ công cụ GUI mà Sun Microsystems phát triển để xây dựng các ứng dụng tối ưu dùng cho window (bao gồm các thành phần như nút, thanh cuộn,…).
* [Swing](https://en.wikipedia.org/wiki/Swing_(Java)) được xây dựng trên AWT API và hoàn toàn được viết bằng Java. Tuy nhiên, nó lại khác với AWT ở chỗ bộ công cụ này thuộc loại nền tảng độc lập, bao gồm các thành phần nhẹ và phức tạp hơn AWT.
* Các gói javax.swing bao gồm các lớp cho Java Swing API như JMenu, JButton, JTextField, JRadioButton, JColorChooser,…
* Việc xây dựng ứng dụng sẽ trở nên dễ dàng hơn với **Java Swing** vì chúng ta có các bộ công cụ GUI giúp đỡ công việc.
* Swing được chính thức phát hành vào tháng 3/1998. Nó đi kèm với thư viện Swing 1.0 với hơn 250 lớp, 80 giao tiếp.
* Hiện nay con số này đã được tăng lên, ở phiên bản Swing 1.4 có 451 lớp và 85 giao tiếp.
* Các bản phát hành Java 2 (SDK 1.2 và các phiên bản mới hơn) đều bao gồm Swing trong runtime environment.
* **Java Swing** được dùng để hỗ trợ tạo giao diện đồ hoạ người dùng (với Java).
* Bộ công cụ này cung cấp các bộ điều khiển nâng cao như thanh trượt, colorpicker, Tree, TabbedPane và bảng điều khiển,..
* Swing có những đặc điểm:
* Độc lập với thiết bị
* Có thể tuỳ chỉnh, mở rộng
* Khá nhẹ
* Có thể cấu hình
* Ngoài ra bạn cũng có thể tùy chỉnh các điều khiển xoay một cách dễ dàng mà không ảnh hưởng đến các thành phần khác.

1. **Creational Design Pattern là gì ?**
   * + Những mẫu thiết kế này là tất cả về khởi tạo lớp. Mẫu này có thể được chia thành các mẫu tạo lớp và mẫu tạo đối tượng. Trong khi các mẫu tạo lớp sử dụng hiệu quả kế thừa trong quy trình khởi tạo, các mẫu tạo đối tượng sử dụng ủy quyền hiệu quả để hoàn thành công việc.”
     + “Trong công nghệ phần mềm, các mẫu thiết kế sáng tạo là các mẫu thiết kế xử lý các cơ chế tạo đối tượng, cố gắng tạo các đối tượng theo cách phù hợp với tình huống. Hình thức cơ bản của việc tạo đối tượng có thể dẫn đến các vấn đề thiết kế hoặc tăng thêm độ phức tạp cho thiết kế. Các mẫu thiết kế sáng tạo giải quyết vấn đề này bằng cách nào đó kiểm soát việc tạo đối tượng này.”

***Đó là:***

* Abstract Factory
  + - * "Tạo một thể hiện của một số họ các lớp. " "Tạo các đối tượng của các đối tượng phụ thuộc liên quan Builder - Xây dựng các đối tượng phức tạp bằng cách sử dụng từng bước một"
* Builder
  + - * "Tách xây dựng đối tượng từ đại diện của nó"
* Factory Method
  + - * "Tạo một thể hiện của một số lớp dẫn xuất. Tạo đối tượng của một số lớp liên quan mà không chỉ định đối tượng chính xác được tạo."
* Object Pool
  + - * "Tránh mua lại và giải phóng tài nguyên đắt tiền bằng cách tái chế các đối tượng không còn sử dụng"
* Prototype
  + - * Một trường hợp khởi tạo đầy đủ sẽ được sao chép hoặc nhân bản (clone)
* Singleton
  + - * "Một lớp chỉ tồn tại một thể hiện duy nhất. " "Đảm bảo rằng hầu hết chỉ một thể hiện của một đối tượng tồn tại trong suốt ứng dụng."

1. **Factory Pattern là gì ?**

* Factory Pattern là một trong những Pattern phổ biến trong [lập trình hướng đối tượng](https://chiasekinang.com/lap-trinh-huong-doi-tuong/).
* Factory Pattern thuộc nhóm Creational Design Pattern. Vì nhiệm vụ của mẫu này là quản lý và trả về các đối tượng theo yêu cầu, cung cấp một trong những cách tốt nhất để tạo một đối tượng. Giúp cho việc khởi tạo đổi tượng một cách linh hoạt hơn.
* Factory pattern là một trong những mẫu thiết kế được sử dụng nhiều nhất trong Java.
* Trong Factory Pattern, chúng ta tạo đối tượng mà không để lộ logic tạo đối tượng ở phía người dùng và tham chiếu đến đối tượng mới được tạo ra bằng cách sử dụng một interface chung.

1. **Công dụng của Factory Pattern:**
2. **Lợi ích của Factory Pattern:**

* Giúp việc khởi tạo các Objects mà che giấu đi xử lí logic của việc khởi tạo đó. Người dùng không biết logic thực sự được khởi tạo bên dưới phương thức factory.
* Mẫu thiết kế này cho phép các lớp con chọn kiểu đối tượng cần tạo.
* Nó thúc đẩy sự liên kết lỏng lẻo bằng cách loại bỏ sự cần thiết phải ràng buộc các lớp cụ thể vào code. Nghĩa là code chỉ tương tác với interface hoặc lớp abstract, để nó sẽ làm việc với bất kỳ lớp nào implements interface đó hoặc extends lớp abstract.
* Factory Pattern giúp giảm sự phụ thuộc giữa các module: cung cấp 1 hướng tiếp cận với Interface thay vì các implement. Giúp chuơng trình độc lập với những lớp cụ thể mà chúng ta cần tạo 1 **đối tượng**, code ở phía client sẽ không bị ảnh hưởng khi thay đổi logic ở factory hay sub class.
* Việc mở rộng code dễ dàng hơn: khi cần mở rộng, chỉ việc tạo ra những sub class và implement thêm vào factory method.
* Dễ dạng quản lý life cycle của các Object được tạo bởi **Factory Method Pattern**.
* Thống nhất về mặt naming convention: giúp cho các developer có thể hiểu về cấu trúc source code.

1. **Ưu điểm và nhược điểm của Factory Method Pattern:**
2. **Ưu điểm:**

* Factory Method Pattern giúp hạn chế sự phụ thuộc giữa creator và concrete products.
* Factory Method Pattern giúp gom các đoạn code tạo ra product vào một nơi trong chương trình, nhờ đó giúp dễ theo dõi và thao tác.
* Với Pattern này, chúng ta có thể dễ dàng mở rộng, thêm những đoạn code mới vào chương trình mà không cần phá vỡ các đối tượng ban đầu.

1. **Nhược điểm:**

* Mã code có thể trở nên phức tạp hơn mức bình thường do đòi hỏi phải sử dụng nhiều class mới có thể cài đặt được pattern này.

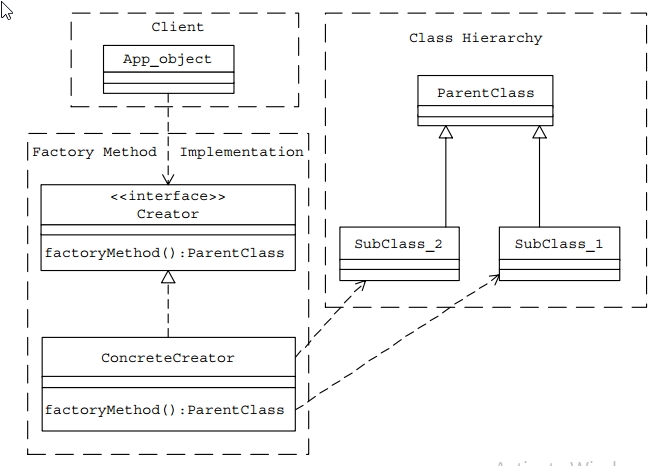
1. **Khởi tạo Factory Pattern:**
2. **Factory Pattern sử dụng khi nào ?**

* Chúng ta có một super class với nhiều class con và dựa trên đầu vào, chúng ta cần trả về một class con. Mô hình này giúp chúng ta đưa trách nhiệm của việc khởi tạo một lớp từ phía người dùng (client) sang lớp Factory.
* Chúng ta không biết sau này sẽ cần đến những lớp con nào nữa. Khi cần mở rộng, hãy tạo ra sub class và implement thêm vào factory method cho việc khởi tạo sub class này.

1. **Khởi tạo:**

Một Factory Pattern bao gồm các thành phần cơ bản sau:

* **Super Class**: môt supper class trong Factory Pattern có thể là một **interface**, **abstract class** hay một **class** thông thường.
* **Sub Classes**: các sub class sẽ implement các phương thức của **supper class** theo nghiệp vụ riêng của nó.
* **Factory Class**: một class chịu tránh nhiệm khởi tạo các đối tượng **sub class** dựa theo tham số đầu vào. Factory class sử dụng if-else hoặc switch-case để xác định class con đầu ra.

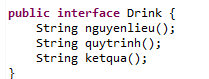


1. **Ví dụ về mẫu Factory Pattern:**
2. **Đề bài:**

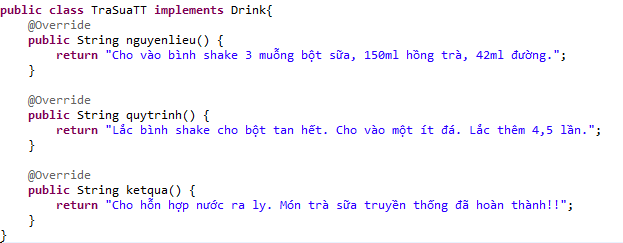
* Thể hiện quy trình tạo ra một số món nước của một quán trà sữa.

1. **Thực thi:**

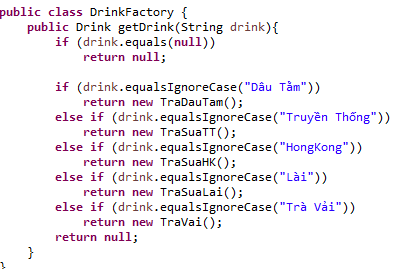
* Tạo một **Supper Class** chứa các method mà các món nước cần có để thẻ hiện quy trình.



* Tạo một số **Sub Classes** ứng với các món nước mà quán có.



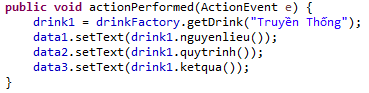
* Tạo một **Factory Class** để gọi các **Sub Class**(món nước) mà ta muốn pha chế ra.



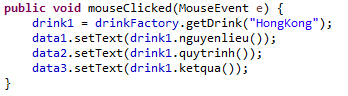
* Cuối cùng gọi và tạo mới lại **Super Class** và **Factory Class** ở Client để truyền chuỗi vào gọi ra các **Sub Class** tương ứng.



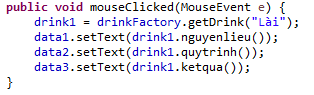
* Truyền chuỗi vào button “TSTT”



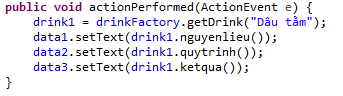
* Truyền chuỗi vào button “TS HongKong”



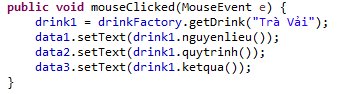
* Truyền chuỗi vào button “TS Lài”



* Truyền chuỗi vào button “Trà Dâu Tằm”



* Truyền chuỗi vào button “Trà Vải”



* Kết quả khi chạy chương trình:
* Ấn vào button “TSTT”



* Ấn vào button “Trà Dâu Tằm”



1. **Tài liệu tham khảo:**

* [Tổng hợp Design Pattern - Factory Pattern là gì? Ví dụ - chia sẻ kĩ năng (chiasekinang.com)](https://chiasekinang.com/factory-pattern/)
* [Hướng dẫn Java Design Pattern - Factory Method - GP Coder (Lập trình Java)](https://gpcoder.com/4352-huong-dan-java-design-pattern-factory-method/)
* [Creational Design Pattern (viblo.asia)](https://viblo.asia/p/creational-design-pattern-bWrZnpjr5xw)
* [OOP là gì? 4 đặc tính cơ bản của OOP - ITviec Blog](https://itviec.com/blog/oop-la-gi/?utm_source=google&utm_medium=dis_cpc&utm_campaign=hcm_performancemax&gclid=CjwKCAjwu5yYBhAjEiwAKXk_eKWmxbXXBjS4FWn1pGzX4DMJ-qXTIhazSv3E1aSAF1jHelrhsJokMhoCyvoQAvD_BwE)
* [Java Swing là gì? Cách học lập trình Java Swing cơ bản | Ironhack VN (ironhackvietnam.edu.vn)](https://ironhackvietnam.edu.vn/java-swing/)