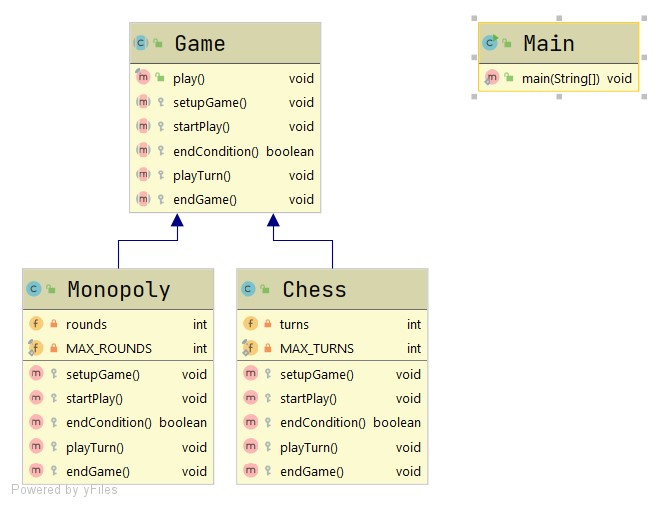
**Mẫu Template Method:**

**Giới thiệu:**

Định nghĩa cấu trúc của một thuật toán trong một phương thức, để các lớp con có thể cung cấp các triển khai cụ thể mà không thay đổi cấu trúc tổng thể.

**Sau đây là ví dụ về mẫu Template Method:**



Đây là ví dụ về cách sử dụng mẫu Template Method để xây dựng một framework cho các ứng dụng mô phỏng trò chơi.

**Cấu trúc:**

1. Abstract class Game:

Đây là một lớp trừu tượng đóng vai trò là bản thiết kế chi tiết để tạo logic trò chơi.

Nó xác định các hằng số như MAX\_ROUNDS và MAX\_TURNS để chỉ định số vòng và lượt tối đa mà trò chơi có thể có theo mặc định.

Nó có các phương thức xác định cấu trúc tổng thể của trò chơi.

Các phương thức trừu tượng trong lớp này (setupSpecifics, playTurn, endCondition và endSpecifics) đóng vai trò giữ chỗ cho các lớp con để triển khai logic dành riêng cho trò chơi.

public abstract class Game {

// Constants

protected static final int ***MAX\_ROUNDS*** = 10;

protected static final int ***MAX\_TURNS*** = 5;

// Template methods

public final void setupGame() {

System.***out***.println("Setting up the game...");

setupSpecifics();

}

public final void startPlay() {

System.***out***.println("Starting the game...");

for (int round = 1; round <= ***MAX\_ROUNDS***; round++) {

System.***out***.println("--- Round " + round + " ---");

for (int turn = 1; turn <= ***MAX\_TURNS***; turn++) {

playTurn();

if (endCondition()) {

break;

}

}

}

}

public final void endGame() {

System.***out***.println("Ending the game...");

endSpecifics();

}

// Hook methods

protected abstract void setupSpecifics();

protected abstract void playTurn();

protected abstract boolean endCondition();

protected abstract void endSpecifics();

}

1. Class Monopoly:

Lớp này kế thừa từ lớp Game và đại diện cho một trò chơi Monopoly.

Nó ghi đè các phương thức trừu tượng từ lớp Trò chơi để cung cấp các triển khai dành riêng cho Monopoly.

public class Monopoly extends Game {

*@Override*

protected void setupSpecifics() {

System.***out***.println("Setting up Monopoly...");

// Monopoly-specific setup

}

*@Override*

protected void playTurn() {

System.***out***.println("Playing a turn in Monopoly...");

// Monopoly-specific play turn

}

*@Override*

protected boolean endCondition() {

System.***out***.println("Checking end condition in Monopoly...");

// Monopoly-specific end condition

return false; // Replace with actual condition

}

*@Override*

protected void endSpecifics() {

System.***out***.println("Ending Monopoly...");

// Monopoly-specific end actions

}

}

1. Class Chess:

Lớp này kế thừa từ lớp Game và đại diện cho một ván Cờ vua.

Nó ghi đè các phương thức trừu tượng từ lớp Trò chơi để cung cấp các triển khai dành riêng cho Cờ vua.

public class Chess extends Game {

*@Override*

protected void setupSpecifics() {

System.***out***.println("Setting up Chess...");

// Chess-specific setup

}

*@Override*

protected void playTurn() {

System.***out***.println("Playing a turn in Chess...");

// Chess-specific play turn

}

*@Override*

protected boolean endCondition() {

System.***out***.println("Checking end condition in Chess...");

// Chess-specific end condition

return false; // Replace with actual condition

}

*@Override*

protected void endSpecifics() {

System.***out***.println("Ending Chess...");

// Chess-specific end actions

}

}

1. Class Main:

Lớp này đóng vai trò là điểm vào của chương trình.

Nó tạo ra các phiên bản của trò chơi Cờ vua và Cờ vua.

Nó gọi các phương thức setupGame, startPlay và endGame trên các đối tượng trò chơi tương ứng để bắt đầu và kết thúc trò chơi.

public class Main {

public static void main(String[] args) {

// Create a Monopoly game instance

Game monopolyGame = new Monopoly();

// Start the Monopoly game

monopolyGame.setupGame();

monopolyGame.startPlay();

monopolyGame.endGame();

// Create a Chess game instance

Game chessGame = new Chess();

// Start the Chess game

chessGame.setupGame();

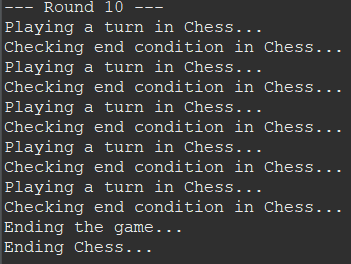
chessGame.startPlay();

chessGame.endGame();

}

}

**Kết quả và thử nghiệm:**



Chương trình sẽ bắt đầu trò chơi, kiểm tra điều kiện và thực hiện các bước đi, kết thúc game nếu thỏa điều kiện.

Mẫu thiết kế Template Method có thể được áp dụng trong việc xây dựng một framework cho các ứng dụng mô phỏng trò chơi. Ở đây, chúng ta sẽ tạo ra một lớp trừu tượng Game với một số phương thức mẫu xác định luồng chính của trò chơi, trong khi để các lớp con ghi đè các chi tiết cụ thể của từng trò chơi.

**Lợi ích:**

Tăng tính linh hoạt: Mẫu Template Method cho phép các lớp con tùy chỉnh các bước cụ thể của thuật toán, giúp tăng tính linh hoạt cho framework và cho phép các trò chơi cụ thể có thể thực hiện các quy tắc và hành vi khác nhau.

Tăng tính tái sử dụng: Mẫu Template Method giúp tái sử dụng mã chung giữa các trò chơi khác nhau, giúp tiết kiệm thời gian và công sức cho việc phát triển.

Dễ bảo trì và sửa chữa.