**COMMAND PATTERN:**

package commandpatternexample;

public class CommandPatternExample {

public static void main(String[] args) {

Light livingRoomLight = new Light();

// Create concrete commands

Command lightOn = new LightOnCommand(livingRoomLight);

Command lightOff = new LightOffCommand(livingRoomLight);

// Create invoker

RemoteControl remote = new RemoteControl();

// Turn on the light

remote.setCommand(lightOn);

remote.pressButton();

// Turn off the light

remote.setCommand(lightOff);

remote.pressButton();

}

}

// Command.java

interface Command {

void execute();

}

// LightOnCommand

class LightOnCommand implements Command {

private Light light;

public LightOnCommand(Light light) {

this.light = light;

}

@Override

public void execute() {

light.turnOn();

}

}

// LightOffCommand

class LightOffCommand implements Command {

private Light light;

public LightOffCommand(Light light) {

this.light = light;

}

@Override

public void execute() {

light.turnOff();

}

}

// RemoteControl.java

class RemoteControl {

private Command command;

public void setCommand(Command command) {

this.command = command;

}

public void pressButton() {

command.execute();

}

}

// Light

class Light {

public void turnOn() {

System.out.println("The light is on.");

}

public void turnOff() {

System.out.println("The light is off.");

}

}

**LINEAR AND BINARY SEARCH:**

package searchingdsa;

import java.util.Arrays;

public class SearchingDSA {

public static void main(String[] args) {

Product[] products = {

new Product(1, "Laptop", "Electronics"),

new Product(2, "Smartphone", "Electronics"),

new Product(3, "Tablet", "Electronics"),

new Product(4, "Headphones", "Accessories"),

new Product(5, "Keyboard", "Accessories")

};

// Linear search example

int targetId = 3;

Product result = SearchAlgorithms.linearSearch(products, targetId);

System.out.println("Linear Search Result: " + result);

// Binary search example

Arrays.sort(products, (a, b) -> a.getProductId() - b.getProductId()); // Ensure array is sorted

result = SearchAlgorithms.binarySearch(products, targetId);

System.out.println("Binary Search Result: " + result);

}

}

class Product {

private int productId;

private String productName;

private String category;

public Product(int productId, String productName, String category) {

this.productId = productId;

this.productName = productName;

this.category = category;

}

public int getProductId() {

return productId;

}

public String getProductName() {

return productName;

}

public String getCategory() {

return category;

}

@Override

public String toString() {

return "Product{" +

"productId=" + productId +

", productName='" + productName + '\'' +

", category='" + category + '\'' +

'}';

}

}

class SearchAlgorithms {

public static Product linearSearch(Product[] products, int targetId) {

for (Product product : products) {

if (product.getProductId() == targetId) {

return product;

}

}

return null;

}

public static Product binarySearch(Product[] products, int targetId) {

int left = 0;

int right = products.length - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (products[mid].getProductId() == targetId) {

return products[mid];

}

if (products[mid].getProductId() < targetId) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return null;

}

}