**Trường đại học Tài nguyên và Môi trường thành phố hồ chí minh**

**khoa: hệ thống thông tin viễn thám**



**Báo cáo đồ án**

**lập trình hướng đối tượng**

Thành viên nhóm:

Trương kiều My\_ 1050080145

Phan Thị Hoàng Oanh\_10500801

Bùi Thị Thu Thảo\_10500801

Giảng viên hỗ trợ: Phạm Trọng Huỳnh

**Thành phố Hồ Chí Minh tháng 3 năm 2023**

**Trường đại học Tài nguyên và Môi trường thành phố hồ chí minh**

**khoa: hệ thống thông tin viễn thám**



**Báo cáo đồ án**

**lập trình hướng đối tượng**

Thành viên nhóm:

Trương kiều My\_ 1050080145

Phan Thị Hoàng Oanh\_10500801

Bùi Thị Thu Thảo\_10500801

Giảng viên hỗ trợ: Phạm Trọng Huỳnh

**Thành phố Hồ Chí Minh tháng 3 năm 2023**

**Mục lục**

1. Giới thiệu đồ án…………………………………………………………...…3
2. Phân tích……………………..………………………………………………..4
3. Thiết kế……………………………………………..……………………..…..6
4. Hiện thực …………………………………………..……….…………..…….7
5. Kết luận và hướng phát triển………………………………….……………..24
6. Tài liệu tham khảo ………………………………………….……………….25
7. Phụ lục………………………………………………………………………26

# 

# 1. Giới thiệu

Đồ án là một trong những yếu tố khá quan trọng trong cuộc đời của sinh viên. Một đồ án hoàn chỉnh cũng phản ánh lên tinh thần học tập của sinh viên có thật thực nghiêm túc, tìm hiểu và thực hành nó có hiệu quả hay không.

Với nhiếp ảnh, mỗi thể loại đều mang một ý nghĩa riêng và chúng đều có mục đích khác nhau. Do đó việc xác định rõ các thể loại trong nhiếp ảnh là vấn đề rất cần thiết đối với những người quan tâm đến loại hình nghệ thuật này. Trong bài viết này, chung tôi sẽ giới thiệu đến bạn các thể loại nhiếp ảnh phổ biến nhất mà chúng tôi biết.

 Nhiếp ảnh phong cảnh là ghi lại một thiên nhiên mà con người trong ảnh (nếu có) không chiếm một vị trí quá lớn. Ảnh phong cảnh bao giờ cũng mang một ý đồ nghệ thuật, một nội dung tư tưởng rõ ràng. Thành công của một bức ảnh phong cảnh ngoài nội dung tư tưởng, ảnh phải mang được cái hồn của đất nước, địa danh, hoặc xứ sở nào đó. Đó là dấu ấn chủ quan hay là cái riêng của nhà nhiếp ảnh; đồng thời ảnh phải mang giá trị thẩm mỹ cao qua bố cục, đường nét, ánh sáng.

Việc có được một bức ảnh phong cảnh đẹp phụ thuộc rất nhiều vào màu sắc, bối cảnh xung quanh, máy ảnh và hơn cả là sự chuyên nghiệp chỉnh chu của các anh thợ chụp ảnh. Một bức ảnh đẹp phụ thuộc vào rất nhiều yếu tố, đặc biệt là ảnh phong cảnh thì phụ thuộc rất nhiều vào không gian xung quanh. Chính vì biết được điều này mà nên nhiều IT đã cho dã ứng dụng quản lý tranh ảnh giúp việc lưu trữ quản lý và tìm kiếm thêm dễ dàng hơn, giúp tiết kiệm thời gian và công sức cho các nhiếp ảnh gia phong cảnh.

Đồ án quản lý ảnh phong cảnh được sử dụng nhiều cho các photography, các nhiếp ảnh gia dùng trong việc lưu trữ và quản lý ảnh ảnh thêm phần hiệu quả và tiết kiệm nhiều thời gian công sức quản lý cũng như tìm kiếm tranh ảnh. Quản lý kho lớn phần tranh ảnh không khỏi những sai sót bất cẩn, chưa tính đến việc khi 1 tìm kiếm 1 trong nhiều bức ảnh được lưu trữ không dễ dàng. Chính vì thế việc tạo ra một chương trình quản lý tranh ảnh là điều rất cần thiết và quan trọng đối với nhiếp ảnh gia. Quản lý tranh ảnh phong cảnh giúp nhiếp ảnh gia có thể phân loại ảnh, lưu trữ tìm kiếm dễ dàng hơn.

Đồ án về quản lý ảnh phong cảnh là một đề tài giúp ích được cho công việc khác ( chụp ảnh ), nó thật sự cần thiết cho các nhiếp ảnh, photography,… những người yêu nghệ thuật,yêu cái đẹp của thiên nhiên, yêu môi trường sống quanh mình và muốn lưu trữ lại nó. Tranh ảnh phong cảnh không chỉ là nơi lưu trữ cái đẹp mà còn lưu trữ thiên nhiên , cuộc sống xung quanh và là tình cảm của con người đối với cảnh vật xung quanh nơi mình sinh sống thông các các bức ảnh mà họ lưu trữ.

**2. Phân tích đề tài**

Để quản lý được kho ảnh phong cảnh ta cần thiết kế được mô hình UML để thiết kế chi tiết việc quản lý kho ảnh theo từng mục. Việc xây dựng kho quản lý ảnh phong cảnh còn phụ thuộc vào thư viện lưu trữ, kích thước của kho lưu trữ, kích thước ảnh, phần mô tả ảnh, ….

Việc xây dựng kho lưu trữ cho một kho ảnh phong cảnh khá đơn giản vì đã cho chúng ta biết được rõ đối tượng cần xây dựng mô nhình quản lý. Chính vì thế khiết kế UML cũng như xây dựng quản lý đối tượng đơn giản hơn. Cụ thể mình có thể xem cách xây dựng cũng như phân tích thiết kế như sau:

***Quản lý kho ảnh***:

***Ảnh Phong Cảnh:***

***Ảnh thiên nhiên:***

***Ảnh đời sống***

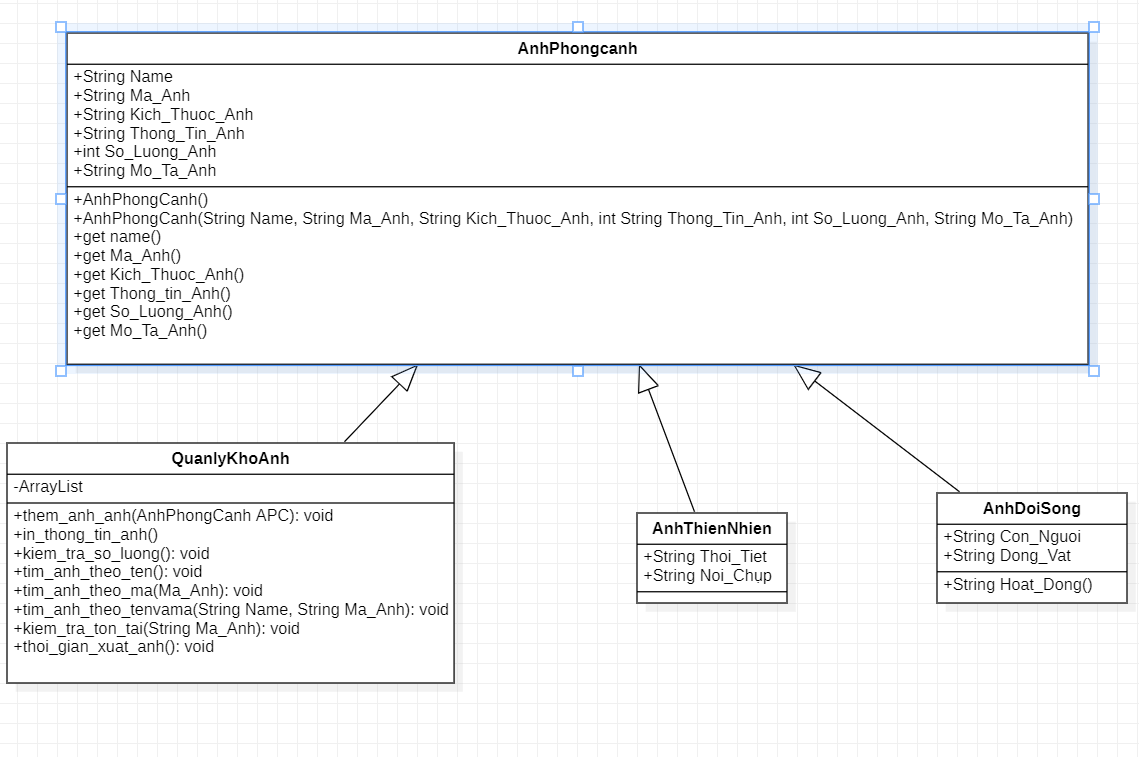
**Yêu cầu chức năng:**

Yêu cầu chính của đề tài là làm rõ các vấn đề sau: kho có bao nhiêu thư viện, số lượng ảnh có thể chứa của mỗi thư viện, ảnh thuộc thư viện nào, ảnh có dung lượng bao nhiêu, mô tả chi tiết về ảnh.

***Yêu cầu phi chức năng* :**

Dung lượng, sức chứa, độ tin cậy, bảo mật, hiệu suất

# Thiết kế

**Mô hình thiết kế UML**

# Hiện thực

Một số mã giả của thuật toán áp dụng cho hệ thống dùng mã lệnh java sever code

* **Quản lý kho ảnh**

package qlka;

import java.util.List;

public class QuanLyKhoAnh {

ArrayList<AnhPhongCanh> APCn = new ArrayList<>();

Scanner sc = new Scanner(System.in);

public void them\_hinh\_anh(AnhPhongCanh APC) {

APCn.add(APC);

}

public void in\_thong\_tin\_anh() {

for (AnhPhongCanh AnhPhongCanh : APCn) {

System.out.println(AnhPhongCanh.toString());

}

}

public void in\_AnhThienNhien() {

for (AnhPhongCanh AnhPhongCanh : APCn) {

if (AnhPhongCanh instanceof AnhDoiong) {

System.out.println(AnhPhongCanh.toString());

}

}

}

public void in\_AnhDoiSong() {

for (AnhPhongCanh AnhPhongCanh : APCn) {

if ( AnhPhongCanh instanceof AnhThienNhien) {

System.out.println(AnhPhongCanh.toString());

}

}

}

public int kiem\_tra\_so\_luong() {

return APCn.size();

}

public void tim\_anh\_theo\_ten(String name) {

for (AnhPhongCanh AnhPhongCanh : APCn) {

if (AnhPhongCanh.getName().indexOf(name) >= 0) {

System.out.println(AnhPhongCanh);

}

}

}

public void tim\_may\_theo\_ma(String Ma\_Anh) {

for (AnhPhongCanh AnhPhongCanh : APCn) {

if (AnhPhongCanh.getMa\_Anh().indexOf(Ma\_Anh) >= 0) {

System.out.println(AnhPhongCanh);

}

}

}

public void tim\_may\_theo\_tenvama(String name, String Ma\_Anh) {

for (AnhPhongCanh AnhPhongCanh : APCn) {

if (AnhPhongCanh.getName().indexOf(name) >= 0 && AnhPhongCanh.getMa\_Anh().indexOf(Ma\_Anh) >= 0) {

System.out.println(AnhPhongCanh);

}

}

}

public void kiem\_tra\_ton\_tai(String Ma\_Anh) {

for (AnhPhongCanh AnhPhongCanh : APCn) {

if (APCn.contains(Ma\_Anh)) {

System.out.println(AnhPhongCanh);

}

}

}

public boolean xoa\_anh\_theo\_code(AnhPhongCanh APC) {

return APCn.remove(APC);

}

public void ngay\_xuat\_kho() {

Calendar cal = Calendar.getInstance();

Date date = cal.getTime();

System.out.print("Thoi gian xuat anh cua he thong: " + date);

System.out.println("\n");

}

}

}

* **Ảnh phong cảnh**

package qlka;

public class AnhPhongCanh {

public class AnhPhongCanh implements Comparable<AnhPhongCanh> {

private String Name, Ma\_Anh, Kich\_Thuoc\_Anh, Thong\_Tin\_Anh, Mo\_Ta\_Anh,

private int So\_Luong\_Anh;

private NgayNhapXuat date;

public AnhPhongCanh() {

}

public AnhPhongCanh(String Ma\_Anh) {

this.Ma\_Anh = Ma\_Anh;

}

public AnhPhongCanh(String Name, String Ma\_Anh, String Kich\_Thuoc\_Anh, String Thong\_Tin\_Anh, String Mo\_Ta\_Anh, int So\_Luong\_Anh, NgayNhapXuat date) {

this.Name = Name;

this.Ma\_Anh = Ma\_Anh;

this.Mo\_Ta\_Anh = Mo\_Ta\_Anh;

this.So\_Luong\_Anh = So\_Luong\_Anh;

this.Thong\_Tin\_Anh = Thong\_Tin\_Anh;

this.date = date;

}

public String getName() {

return Name;

}

public void setName(String Name) {

this.Name = Name;

}

public String getMa\_Anh() {

return Ma\_Anh;

}

public void setMa\_Anh(String Ma\_Anh) {

this.Ma\_Anh = Ma\_Anh;

}

public String getMo\_Ta\_Anh() {

return Mo\_Ta\_Anh;

}

public void setMo\_Ta\_Anh(String Mo\_Ta\_Anh) {

this.Mo\_Ta\_Anh = Mo\_Ta\_Anh;

}

public String getThong\_Tin\_Anh() {

return Thong\_Tin\_Anh;

}

public void setThong\_Tin\_Anh(String Thong\_Tin\_Anh) {

this.Thong\_Tin\_Anh = Thong\_Tin\_Anh;

}

public int getSo\_Luong\_Anh() {

return So\_Luong\_Anh;

}

public void setSo\_Luong\_Anh(int So\_Luong\_Anh) {

this.So\_Luong\_Anh = So\_Luong\_Anh;

}

public NgayNhapXuat getDate() {

return date;

}

public void setDate(NgayNhapXuat date) {

this.date = date;

}

@Override

public String toString() {

return "Name=" + Name + ", Ma\_Anh=" + Ma\_Anh + ", Kich\_Thuoc\_Anh=" + Kich\_Thuoc\_Anh + ", So\_Luong\_Anh=" + So\_Luong\_Anh + ", Thong\_Tin\_Anh=" + Thong\_Tin\_Anh + ", date=" + date;

}

@Override

public int compareTo(AnhPhongCanh o) {

return this.Ma\_Anh.compareTo(o.Ma\_Anh);

}

@Override

public int hashMa\_Anh() {

return Ma\_Anh.compareTo(Ma\_Anh);

}

@Override

public boolean equals(Object obj) {

if(obj instanceof AnhPhongCanh) {

AnhPhongCanh another = (AnhPhongCanh) obj;

if (this.Ma\_Anh.equals(another.Ma\_Anh)) {

return true;

}

}return false;

}

}

}

* **Ảnh Đời Sống**

public class AnhDoiSong {

public class AnhDoiSong extends AnhPhongCanh {

private String Con\_Nguoi, Dong\_Vat, Hoat\_Dong;

public AnhDoiSong() {

}

public AnhDoiSong(String Name, String Ma\_Anh, String Thong\_Tin\_Anh, String Mo\_Ta\_Anh, int So\_Luong\_Anh, NgayNhapXuat date, String Con\_Nguoi, String Dong\_Vat, String Hoat\_Dong)

this.Con\_Nguoi = Con\_Nguoi;

this.Dong\_Vat = Dong\_Vat;

this.Hoat\_Dong = Hoat\_Dong;

}

public String getCon\_Nguoi() {

return Con\_Nguoi;

}

public void setCon\_Nguoi(String Con\_Nguoi) {

this.Con\_Nguoi = Con\_Nguoi;

}

public String getDong\_Vat() {

return Dong\_Vat;

}

public void setDong\_Vat(String Dong\_Vat) {

this. = Dong\_Vat;

}

public String getHoat\_Dong() {

return Hoat\_Dong;

}

public void setHoat\_Dong(String Hoat\_Dong) {

this.Hoat\_Dong = Hoat\_Dong;

}

@Override

public String toString() {

return "AnhDoiSong{" + super.toString() + ", Con\_Nguoi=" + Con\_Nguoi + ", Dong\_Vat=" + Dong\_Vat + ", Hoat\_Dong=" + Hoat\_Dong + '}';

}

}

}

* **Ảnh thiên nhiên**

public class AnhThienNhien {

public class AnhThienNhien extends AnhPhongCanh {

private String Thoi\_Tiet, Noi\_Chup;

public AnhThienNhien() {

}

public AnhThienNhien(String Name, String Ma\_Anh, String Kich\_Thuoc\_Anh, String Thong\_Tin\_Anh, String Mo\_Ta\_Anh, int So\_Luong\_Anh, NgayNhapXuat date, String Thoi\_Tiet, String Noi\_Chup) {

super(Name, NSX, Code, CPU, CardDoHoa, RAM, ROM, Price, date);

this.Thoi\_Tiet = Thoi\_Tiet;

this.Noi\_Chup = Noi\_Chup;

}

public String getThoi\_Tiet() {

return Thoi\_Tiet;

}

public void setThoi\_Tiet(String Thoi\_Tiet) {

this.Thoi\_Tiet = Thoi\_Tiet;

}

public String getNoi\_Chup() {

return Noi\_Chup;

}

public void setNoi\_Chup(String Noi\_Chup) {

this.Noi\_Chup = Noi\_Chup;

}

@Override

public String toString() {

return "AnhDoiSong{" + super.toString() + ", Noi\_Chup=" + Noi\_Chup + ", Thoi\_Tiet=" + Thoi\_Tiet + '}';

}

}

}

* **Main**

public class Main {

public static void main(String[] args) {

Main qlka = new Main();

Scanner sc = new Scanner(System.in);

int Anh;

do {

System.out.println("Chon Anh---------------------------");

System.out.println("1. Them thong tin hinh anh");

System.out.println("2. In thong tin toan bo anh phong canh cos trong kho");

System.out.println("3. Chi in thong tin anh thien nhien");

System.out.println("4. Chi in thong tin anh doi song");

System.out.println("5. Tim kiem anh theo ten");

System.out.println("6. Tim kiem anh theo ma");

System.out.println("7. Tim kiem anh theo ten va ma");

System.out.println("8. Xoa anh");

System.out.println("9. Xuat anh");

System.out.println("0. Thoat chuong trinh-Xin cam on");

System.out.print("Nhap: ");

Anh = sc.nextInt();

switch (Anh) {

case 1:

int Anh\_1;

do {

System.out.println("Anh----------");

System.out.println("1. Anh Thien nhien");

System.out.println("2. Anh Doi Song");

System.out.println("0. Quay lai chuong trinh");

System.out.print("Nhap: ");

Anh\_1 = sc.nextInt();

switch (Anh\_1) {

case 1: {

sc.nextLine();

System.out.print("Nhap ten: ");

String Name = sc.nextLine();

System.out.print("Nhap Ma Anh: ");

String Ma\_Anh = sc.nextLine();

System.out.print("Nhap Kich Thuoc Anh: ");

String Kich\_Thuoc\_Anh = sc.nextLine();

System.out.print("Nhap So Luong Anh: ");

int So\_Luong\_Anh = sc.nextInt();

System.out.print("Nhap Mo Ta Anh: ");

String Mo\_Ta\_Anh = sc.nextLine();

System.out.print("Nhap Noi\_Chup: ");

String Noi\_Chup = sc.nextDouble();

System.out.print("Nhap Thoi\_Tiet: ");

String Thoi\_Tiet = sc.nextLine();

System.out.print("Nhap Thong Tin Anh: ");

String Thong\_Tin\_Anh = sc.nextDouble();

NgayNhapXuat date = new NgayNhapXuat();

System.out.print("Nhap vao nam hien tai: ");

int namhientai = sc.nextInt();

System.out.println("Nhap vao dd/mm/yyyy");

System.out.print("Nhap dd: ");

date.setDay(sc.nextInt());

if (date.getDay() < 1 || date.getDay() > 31) {

do {

System.out.println("Khong hop le, nhap lai");

System.out.print("Nhap lai ngay: ");

date.setDay(sc.nextInt());

} while (date.getDay() < 1 || date.getDay() > 31);

}

System.out.print("Nhap mm: ");

date.setMonth(sc.nextInt());

if (date.getMonth() < 1 || date.getMonth() > 12) {

do {

System.out.println("Khong hop le, nhap lai");

System.out.print("Nhap lai thang: ");

date.setMonth(sc.nextInt());

} while (date.getMonth() < 1 || date.getMonth() > 12);

}

System.out.print("Nhap yyyy: ");

date.setYear(sc.nextInt());

if (date.getYear() < 1943 || date.getYear() > namhientai) {

do {

System.out.println("Khong hop le, nhap lai");

System.out.print("Nhap lai nam: ");

date.setYear(sc.nextInt());

} while (date.getYear() < 1943 || date.getYear() > namhientai);

}

AnhPhongCanh APC = new AnhThienNhien(Name, Ma\_Anh, Kich\_Thuoc\_Anh, So\_Luong\_Anh, Mo\_Ta\_Anh, Thong\_Tin\_Anh, Noi\_Chup, Thoi\_Tiet, date);

qlka.Them\_Anh(APC);

System.out.println("Them thanh cong-----------");

}

break;

case 2: {

sc.nextLine();

System.out.print("Nhap ten: ");

String Name = sc.nextLine();

System.out.print("Nhap Ma\_Anh: ");

String Ma\_Anh = sc.nextLine();

System.out.print("Nhap Mo\_Ta: ");

String Mo\_Ta = sc.nextLine();

System.out.print("Nhap Kich\_Thuoc\_Anh: ");

String Kich\_Thuoc\_Anh = sc.nextLine();

System.out.print("Nhap So\_Luong\_Anh: ");

int So\_Luong\_Anh = sc.nextInt();

System.out.print("Nhap Mo\_Ta\_Anh: ");

String Mo\_Ta\_Anh = sc.nextLine();

System.out.print("Nhap Thong\_Tin\_Anh: ");

String Thong\_Tin\_Anh = sc.nextLine();

System.out.print("Con\_Nguoi: ");

String Con\_Nguoi = sc.nextLine();

System.out.print("Dong\_Vat: ");

String Dong\_Vat = sc.nextDouble();

System.out.print("Nhap Hoat\_Dong: ");

String Hoat\_Dong = sc.nextDouble();

System.out.print("Nhap true hoac false cho Con\_Nguoi: ");

boolean Con\_Nguoi = sc.nextBoolean();

NgayNhapXuat date = new NgayNhapXuat();

System.out.print("Nhap vao nam hien tai: ");

int namhientai = sc.nextInt();

System.out.println("Nhap vao dd/mm/yyyy");

System.out.print("Nhap dd: ");

date.setDay(sc.nextInt());

if (date.getDay() < 1 || date.getDay() > 31) {

do {

System.out.println("Khong hop le, nhap lai");

System.out.print("Nhap lai ngay: ");

date.setDay(sc.nextInt());

} while (date.getDay() < 1 || date.getDay() > 31);

}

System.out.print("Nhap mm: ");

date.setMonth(sc.nextInt());

if (date.getMonth() < 1 || date.getMonth() > 12) {

do {

System.out.println("Khong hop le, nhap lai");

System.out.print("Nhap lai thang: ");

date.setMonth(sc.nextInt());

} while (date.getMonth() < 1 || date.getMonth() > 12);

}

System.out.print("Nhap yyyy: ");

date.setYear(sc.nextInt());

if (date.getYear() < 1943 || date.getYear() > namhientai) {

do {

System.out.println("Khong hop le, nhap lai");

System.out.print("Nhap lai nam: ");

date.setYear(sc.nextInt());

} while (date.getYear() < 1943 || date.getYear() > namhientai);

}

AnhPhongCanh APC = new AnhDoiSong(Name, Ma\_Anh, Kich\_Thuoc\_Anh, So\_Luong\_Anh,Thong\_Tin\_Anh, Mo\_Ta, , Con\_Nguoi, Dong\_Vat, Hoat\_Dong, date);

qlka.Them\_Anh(APC);

System.out.println("Them thanh cong-----------");

}

break;

default:

if (Anh\_1 == 0) {

System.out.println("Thoat chuong trinh");

} else {

System.out.println("Vui long nhap lai lua chon");

}

}

} while (Anh\_1 != 0);

break;

case 2:

if (qlka.Kiem\_Tra\_So\_luong() == 0) {

System.out.println("Hien khong co may trong kho");

} else {

System.out.println("So may hien tai la: " + qlka.Kiem\_Tra\_So\_Luong());

}

qlka.in\_Thong\_Tin\_Anh();

System.out.println("\n");

break;

case 3:

qlka.in\_AnhThienNhien();

System.out.println("\n");

break;

case 4:

qlka.in\_AnhDoiSong();

System.out.println("\n");

break;

case 5:

System.out.println("So anh hien tai la: " + qlka.kiem\_Tra\_So\_Luong());

System.out.println("\n");

break;

case 6:

sc.nextLine();

System.out.print("Nhap ten anh can tim: ");

String Name = sc.nextLine();

qlka.tim\_anh\_theo\_ten(Name);

System.out.println("\n");

break;

case 7:

sc.nextLine();

System.out.print("Nhap ma can tim: ");

String Code = sc.nextLine();

qlka.tim\_anh\_theo\_ma(Ma\_Anh);

System.out.println("\n");

break;

case 8:

sc.nextLine();

System.out.print("Nhap ten can tim: ");

Name = sc.nextLine();

System.out.print("Nhap ma can tim: ");

Ma\_Anh = sc.nextLine();

qlka.tim\_anh\_theo\_tenvama(Name, Ma\_Anh);

System.out.println("\n");

break;

case 9:

sc.nextLine();

System.out.print("Nhap vao anh can xoa: ");

Ma\_Anh = sc.nextLine();

AnhPhongCanh APC = new AnhPhongCanh(APC);

qlka.xoa\_anh\_theo\_ten(Ma\_Anh);

break;

case 10:

sc.nextLine();

System.out.print("Nhap ma anh: ");

Ma\_Anh = sc.nextLine();

qlka.tim\_anh\_theo\_ma(Ma\_Anh);

AnhPhongCanhc APC1 = new AnhPhongCanh(Ma\_Anh));

qlka.xoa\_anh\_tinh\_theo\_ma(APC1);

qlka.ngay\_xuat\_kho();

System.out.println("Da xuat kho!!!!!");

break;

default:

if (menu == 0) {

System.out.println("Thoat chuong trinh");

} else {

System.out.println("Vui long nhap lai lua chon");

}

}

} while (Anh != 0);

}

}

}

}

1. **Kết luận và định hướng phát triển**

Việc viết ra một thư mục để quản lí một tệp tin chưa bao giờ là dễ dàng. Nó mắc khá nhiều thời gian trong việc viết, chạy, chỉnh sửa và cho vào sử dụng. Chính vì thế để sử dụng một ngôn ngữ lập trình một cách thuần thục vào đời sống chưa bao giờ là đơn giản. Một trường trình muốn được ra đời phải thông qua việc phân tích, khảo sát, tìm hiểu mục đích sử dụng,....Chính vì thế việc viết ra chương trình quản lí kho ảnh phong cảnh thật sự là một khó khăn. Tuy nhiên việc tạo ra được một truong trình như vậy đã giúp ích được cho rất nhiều nhiếp ảnh gia, các photograpy chuyên nghiệp trong việc quản lí kho ảnh.

Quản lí ảnh bằng ngôn ngữ java (lập trình hướng đối tượng) là một trong những phương pháp tiện lợi. không phải chỉ quản lí về tranh ảnh phong cảnh mà còn quản lí về quản lí về ảnh sinh hoạt đời thường, ảnh phác thảo, ảnh chân dung, ….. Ngoài lĩnh vực nhiếp ảnh thì ngôn ngữ lập trình còn ứng dụng vào nhiều lĩnh vực trong đời sống. Việc tạo ra một ứng dụng chỉnh ảnh, ứng dụng vẽ, hay thậm chí lập trình ra một web để quản lí nhân sự, quản lí việc làm. tất cả những ứng dụng trên đề giúp cho việc quản lí trở nên dễ dàng hơn, thuật tiện hơn cho nhu cầu làm việc, năng cao hiệu suất làm việc cũng như tiết kiểm thời gian quản lí. Đồng thời việc kiểm soát cũng dễ dàng hơn. Tránh những nhầm lẫn chẳng đáng xảy ra. Hiện nay lập trình hướng đối tượng không phải chỉ có ngô ngữ java mà còn rất nhiều ngôn ngữ khác có thể kể đến như: C++, PYTHON, .NET, PHP, JAVASCRIPT.

1. **Tài liệu tham khảo**

Tài liệu tham khảo tại: <https://www.w3schools.com>

Tài liệu tham khảo tại: <https://topdev.vn/blog/lap-trinh-huong-doi-tuong-oops-trong-java>

Tài liệu tham khảo tại: bing.com/videos

Tài liệutham khảo tại: [Lập trình hướng đối tượng trong Java: lớp và đối tượng | Codelearn](https://codelearn.io/learning/object-oriented-programming-in-java/953136)

Sách tham khảo: Hướng Dẫn lập trình hướng đối tượng Java code

Sách tham khảo: Giáo trình lập trình java

Tài liệu tham khảo: <https://hoclaptrinh.vn>

1. **PHỤ LỤC:**

* C++, PYTHON, .NET, PHP, JAVA, JAVASCRIPT: là ngôn ngữ lập trình
* JPG: là một **hình ảnh raster** được lưu ở định dạng JPEG, thường được sử dụng để lưu trữ ảnh kỹ thuật số và đồ họa được tạo bởi phần mềm chỉnh sửa hình ảnh.
* Package: hay còn gọi là gói trong java là một nhóm các loại class (lớp), interface (giao diện) và gói con tương tự nhau Package
* Override: là hiện tượng 1 phương thức thuộc lớp cha được định nghĩa lại ở lớp con.
* Mô hình UML: **Ngôn ngữ mô hình hóa thống nhất** (tiếng Anh: Unified Modeling Language, viết tắt thành **UML**) là một ngôn ngữ mô hình gồm các ký hiệu đồ họa mà các phương pháp hướng đối tượng sử dụng để thiết kế các hệ thống thông tin một cách nhanh chóng.