**Báo cáo Thực hành Lập trình hướng đối tượng Lab 2**

**Họ và tên: Dương Thái Anh**

**Lớp: 744520**

**MSSV: 20226099**

1. Đề bài

Thiết kế hệ thống mới cho dự án AIMS.

1. Yêu cầu bài toán
2. Đối với người mua hàng

* Duyệt danh sách các DVD có sẵn
* Tìm kiếm DVD theo tiêu đề, danh mục, giá cả
* Xem thông tin chi tiết của 1 DVD
* Thêm DVD vào giỏ hàng
* Xem giỏ hàng
* Sắp xếp DVD theo tiêu đề hoặc chi phí
* Cập nhật số lượng DVD trong giỏ hàng
* Đặt hàng

1. Đối với người quản lý cửa hàng

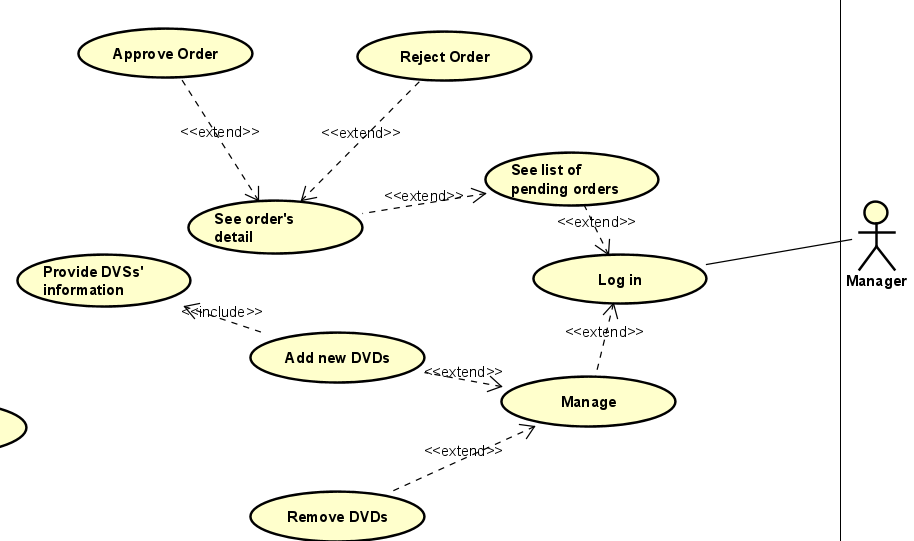
* Đăng nhập, kiểm tra quyền
* Xem danh sách các đơn hàng đang được xử lý
* Xem chi tiết đơn hàng, chấp nhận, từ chối đơn hàng
* Thêm mới, xóa danh sách sản phẩm

1. Use case diagram
2. Use case diagram với người mua hàng

A diagram of a diagram

Description automatically generated

1. Use case diagram với người quản lý cửa hàng



1. Class diagram

A screenshot of a computer

Description automatically generated

1. Các đối tượng
   1. AIMS (đối tượng chính)

Phương thức: void()

* 1. Cart

Thuộc tính:

- qtyOrdered

- MAX\_NUMBER\_ORDER: int

- itemOrdered: DigitalVideoDisc[]

Phương thức:

-addDigitalVideoDisc(disc: DigitalVideoDisc): int

-totalCost(): float

-removeDigitalVideoDisc(disc: DigitalVideoDisc): int

1.3. DigitalVideoDisc

Thuộc tính:

- title: String

- category: String

- director: String

- length: int

- cost: float

Phương thức:

- DigitalVideoDisc(title: String): DigitalVideoDisc

- DigitalVideoDisc(title: String, category: String, cost: float): DigitalVideoDisc

- DigitalVideoDisc(title: String, category: String, director: String, cost: float): DigitalVideoDisc

- DigitalVideoDisc(title: String, category: String,director: String, length: int, cost: float): DigitalVideoDisc

- getTitle(): String

- getCategory(): String

- getDirector(): String

- getLength(): int

- getCost(): float

1. Source code
2. DigitalVideoDisc.java

|  |
| --- |
| /\*  \* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license  \* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template  \*/  /\*\*  \*  \* @author Admin  \*/  public class DigitalVideoDisc {  public DigitalVideoDisc(String title, String category, String director, int length, float cost) {  this.title = title;  this.category = category;  this.director = director;  this.length = length;  this.cost = cost;  }  public DigitalVideoDisc(String title, String category, String director, float cost) {  this.title = title;  this.category = category;  this.director = director;  this.cost = cost;  }  public DigitalVideoDisc(String title, String category, float cost) {  this.title = title;  this.category = category;  this.cost = cost;  }  public DigitalVideoDisc(String title) {  this.title = title;  }  public String getTitle() {  return title;  }  public String getCategory() {  return category;  }  public String getDirector() {  return director;  }  public int getLength() {  return length;  }  public float getCost() {  return cost;  }  private String title;  private String category;  private String director;  private int length;  private float cost;  } |

Câu hỏi: Nếu bạn tạo một constructor method để xây dựng 1 đĩa DVD theo tiêu đề sau đó tạo một constructor method để xây dựng 1 đĩa DVD theo danh mục. Java có cho phép bạn làm điều này không?

Trả lời: Có, Java cho phép một lớp có nhiều constructor, thông qua khái niệm **constructor overloading**. Đây là kỹ thuật tạo nhiều phương thức khởi tạo trong cùng một lớp nhưng với các danh sách tham số khác nhau. Mỗi constructor có thể nhận các tham số khác biệt để khởi tạo đối tượng với các giá trị ban đầu khác nhau, tùy theo tham số truyền vào.

1. Cart.java

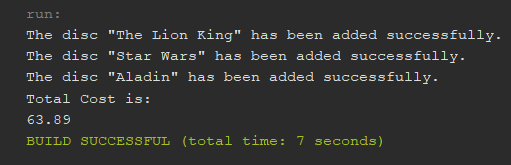
|  |
| --- |
| public class Cart {  public static final int MAX\_NUMBERS\_ORDERED = 20;  private DigitalVideoDisc itemsOrdered[] = new DigitalVideoDisc[MAX\_NUMBERS\_ORDERED];  private int qtyOrdered = 0;    public void addDigitalVideoDisc(DigitalVideoDisc disc) {  if (qtyOrdered < MAX\_NUMBERS\_ORDERED) {  itemsOrdered[qtyOrdered] = disc;  qtyOrdered++;  System.out.println("The disc \"" + disc.getTitle() + "\" has been added successfully.");  } else {  System.out.println("Cannot add \"" + disc.getTitle() + "\": Order is full.");  }  }  public void removeDigitalVideoDisc(DigitalVideoDisc disc) {  for (int i = 0; i < qtyOrdered; i++) {  if (itemsOrdered[i] == disc) {  for (int j = i; j < qtyOrdered - 1; j++) {  itemsOrdered[j] = itemsOrdered[j + 1];  }  itemsOrdered[qtyOrdered - 1] = null;  qtyOrdered--;  System.out.println("The disc \"" + disc.getTitle() + "\" has been removed successfully.");  return;  }  }  System.out.println("The disc \"" + disc.getTitle() + "\" was not found in the order.");  }  public float totalCost(){  float total = 0;  for (int i = 0; i < qtyOrdered; i++){  total += itemsOrdered[i].getCost();  }  return total;  }  } |

1. Aims.java

|  |
| --- |
| public class Aims {  public static void main(String[] args) {  // Tạo một giỏ hàng mới  Cart anOrder = new Cart();  // Tạo các đối tượng DVD mới và thêm chúng vào giỏ hàng  DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);  anOrder.addDigitalVideoDisc(dvd1);  DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);  anOrder.addDigitalVideoDisc(dvd2);  DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);  anOrder.addDigitalVideoDisc(dvd3);  // In tổng chi phí của các mặt hàng trong giỏ hàng  System.out.println("Total Cost is: ");  System.out.println(anOrder.totalCost());  }  } |

1. Demo chương trình

Chạy chương trình như mã nguồn ở phần V



Xóa đĩa dvd3

A screenshot of a computer program

Description automatically generated

Xóa thêm dvd1

A screenshot of a computer program

Description automatically generated

Kết quả trên đúng với lý thuyết!

1. Reading Assignment

