Báo cáo Lab04

Phùng Ngọc Vinh – 20194719

Phần 1:

Bài 1:

1.1

Order.javaẢnh có chứa văn bản, màn hình, ảnh chụp màn hình, màu bạc

Mô tả được tạo tự độngẢnh có chứa văn bản, màn hình, ảnh chụp màn hình

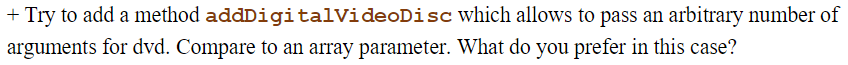
Mô tả được tạo tự động

Aims.javaẢnh có chứa văn bản

Mô tả được tạo tự động

Kết quả: Ảnh có chứa văn bản

Mô tả được tạo tự động

Câu hỏi: 

Trả lời:

Khi truyền vào tham số mảng ta không cần gọi lại hàm DigitalVideoDisc() nhiều lần.

1.2

Order.java Ảnh có chứa văn bản

Mô tả được tạo tự động

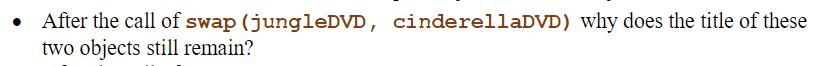
Aims.java Ảnh có chứa văn bản

Mô tả được tạo tự động

Kết quả: Ảnh có chứa văn bản

Mô tả được tạo tự động

Bài 2:



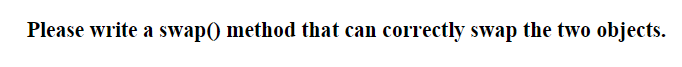
Trả lời:

Vì đây là phương thức được truyền giá trị nên việc thay đổi giá trị chỉ có hiệu lực ở trong phương thức được truyền mà không có làm thay đổi giá trị ở bên ngoài phương thức . Do đó phương thức 2 giá trị jungleDVD và cinderellaDVD chỉ trao đổi giá trị trong hàm Swap => Tittle của 2 object vẫn giữ nguyên



Trả lời:

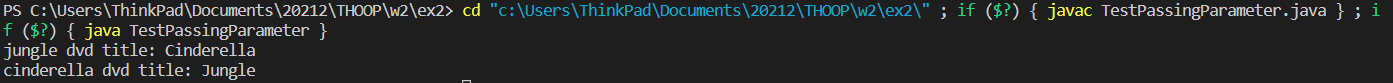
Trong phương thức này, ta đã gọi đến dvd.setTitlle() với tham số truyền vào là cinderellaDVD.getTittle(), nên dù ở bên phương thức thì Tittle của jungleDVD đã bị thay đổi.



TestPassingParameter.java

Ảnh có chứa văn bản

Mô tả được tạo tự động

Kết quả: 

Bài 3:

Aims.java

// package ex3;

public class Aims {

    public static void main(String[] args) {

        Order anOrder = new Order();

        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King");

        dvd1.setCategory("Animation");

        dvd1.setCost(19.95f);

        dvd1.setDirector("Roger Allers");

        dvd1.setLength(87);

        anOrder.addDigitalVideoDisc(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Stat Wars");

        dvd2.setCategory("Science Fiction");

        dvd2.setCost(24.95f);

        dvd2.setDirector("George Lucas");

        dvd2.setLength(124);

        anOrder.addDigitalVideoDisc(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Alddin");

        dvd3.setCategory("Animation");

        dvd3.setCost(18.99f);

        dvd3.setDirector("John Musker");

        dvd3.setLength(90);

        anOrder.addDigitalVideoDisc(dvd3);

        DigitalVideoDisc list[] = new DigitalVideoDisc[3];

        list[0] = dvd1;

        list[1] = dvd2;

        list[2] = dvd3;

        anOrder.addDigitalVideoDisc(list);

        anOrder.addDigitalVideoDisc(list);

        // anOrder.addDigitalVideoDisc(dvd1, dvd2);

        // anOrder.addDigitalVideoDisc(dvd1);

        // anOrder.addDigitalVideoDisc(dvd3, dvd2);

        // System.out.print("Total Cost is: ");

        // System.out.println(anOrder.totalCost());

        MyDate stringDate = new MyDate();

        stringDate.setDay(26);

        stringDate.setMonth(7);

        stringDate.setYear(2001);

        anOrder.setDateOrder(stringDate);

        anOrder.printFull();

        System.out.println(anOrder.getNbOrder());

        Order anOrder2 = new Order();

        anOrder2.setDateOrder(stringDate);

        System.out.println(anOrder.getNbOrder());

    }

}

DigitalVideoDisc.java

// package ex1;

// package ex3;

public class DigitalVideoDisc {

    private String title;

    private String category;

    private String director;

    private int length;

    private float cost;

    public String getTitle() {

        return title;

    }

    public void setTitle(String title) {

        this.title = title;

    }

    public String getCategory() {

        return category;

    }

    public void setCategory(String category) {

        this.category = category;

    }

    public String getDirector() {

        return director;

    }

    public void setDirector(String director) {

        this.director = director;

    }

    public int getLength() {

        return length;

    }

    public void setLength(int length) {

        this.length = length;

    }

    public float getCost() {

        return cost;

    }

    public void setCost(float cost) {

        this.cost = cost;

    }

    public DigitalVideoDisc(String title) {

        super();

        this.title = title;

    }

    // public DigitalVideoDisc(String category) {

    // super();

    // this.category = category;

    // }

    public DigitalVideoDisc(String title, String category) {

        super();

        this.title = title;

        this.category = category;

    }

    public DigitalVideoDisc(String title, String category, String director) {

        super();

        this.title = title;

        this.category = category;

        this.director = director;

    }

    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {

        super();

        this.title = title;

        this.category = category;

        this.director = director;

        this.length = length;

        this.cost = cost;

    }

}

Order.java

// package ex3;

public class Order {

    public static final int MAX\_NUMBERS\_ORDERED = 10;

    private DigitalVideoDisc itemOdered[] = new DigitalVideoDisc[MAX\_NUMBERS\_ORDERED];

    private int qtyOrdered = 0;

    private MyDate dateOrdered = new MyDate();

    public static final int MAX\_LIMITTED\_ORDERED = 5;

    private static int nbOrders = 0;

    public void setDateOrder(MyDate dateOrder) {

        if (nbOrders < MAX\_LIMITTED\_ORDERED) {

            this.dateOrdered = dateOrder;

            System.out.println("The order has been added");

            // nbOrders++;

            if (nbOrders == MAX\_LIMITTED\_ORDERED - 1) {

                System.out.println("The list order is almost full");

            }

        } else {

            System.out.println("The list order is full");

        }

    }

    public Order() {

        if (nbOrders < MAX\_LIMITTED\_ORDERED) {

            nbOrders++;

        } else

            return;

    }

    public int getNbOrder() {

        return nbOrders;

    }

    public void addDigitalVideoDisc(DigitalVideoDisc disc) {

        if (qtyOrdered < MAX\_NUMBERS\_ORDERED) {

            itemOdered[qtyOrdered++] = disc;

            System.out.println(qtyOrdered);

            System.out.println(itemOdered[qtyOrdered - 1].getTitle());

            System.out.print("The disc has been added\n");

            if (qtyOrdered == MAX\_NUMBERS\_ORDERED - 1)

                System.out.print("The order is almost full\n");

        } else {

            System.out.print("The order is full\n");

        }

    }

    public void removeDigitalVideoDisc(DigitalVideoDisc disc) {

        for (int i = 0; i < MAX\_NUMBERS\_ORDERED; i++) {

            if (itemOdered[i] == disc) {

                for (int j = i; j < MAX\_NUMBERS\_ORDERED - 1; j++) {

                    itemOdered[j] = itemOdered[j + 1];

                }

            }

        }

    }

    public float totalCost() {

        float totalCost = 0;

        for (int i = 0; i < qtyOrdered; i++) {

            totalCost += itemOdered[i].getCost();

        }

        return totalCost;

    }

    public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {

        for (int i = 0; i < dvdList.length; i++) {

            if (qtyOrdered >= MAX\_NUMBERS\_ORDERED) {

                System.out.print("The order is full\n");

                return;

            } else {

                itemOdered[qtyOrdered++] = dvdList[i];

                System.out.println(qtyOrdered);

                System.out.println(dvdList[i].getTitle());

                System.out.print("The disc has been added\n");

                if (qtyOrdered == MAX\_NUMBERS\_ORDERED - 1) {

                    System.out.print("The order is almost full\n");

                }

            }

        }

    }

    public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {

        if (qtyOrdered >= MAX\_NUMBERS\_ORDERED) {

            System.out.print("The order is full\n");

            return;

        }

        if (qtyOrdered == MAX\_NUMBERS\_ORDERED - 1) {

            itemOdered[qtyOrdered++] = dvd1;

            System.out.println(qtyOrdered);

            System.out.println(itemOdered[qtyOrdered - 1].getTitle());

            System.out.print("The disc has been added\n");

            System.out.print("The order is full\n");

        }

        if (qtyOrdered < MAX\_NUMBERS\_ORDERED - 1) {

            itemOdered[qtyOrdered++] = dvd1;

            System.out.println(qtyOrdered);

            System.out.println(itemOdered[qtyOrdered - 1].getTitle());

            System.out.print("The disc has been added\n");

            itemOdered[qtyOrdered++] = dvd2;

            System.out.println(qtyOrdered);

            System.out.println(itemOdered[qtyOrdered - 1].getTitle());

            System.out.print("The disc has been added\n");

            if (qtyOrdered == MAX\_NUMBERS\_ORDERED - 1) {

                System.out.print("The order is almost full\n");

            }

            if (qtyOrdered == MAX\_NUMBERS\_ORDERED) {

                System.out.print("The order is full\n");

            }

        }

    }

    public void printFull() {

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Order\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

        System.out.println("Date: " + this.dateOrdered.getDay() + "/" + this.dateOrdered.getMonth() + "/"

                + this.dateOrdered.getYear());

        System.out.println("Ordered Item: ");

        for (int i = 0; i < qtyOrdered; i++) {

            System.out.println(i + 1 + ". " + "DVD - " + itemOdered[i].getTitle() + " - " + itemOdered[i].getCategory()

                    + " - "

                    + itemOdered[i].getDirector() + " - " + itemOdered[i].getLength() + ": " + itemOdered[i].getCost());

        }

        System.out.println("Total cost: " + this.totalCost());

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

    }

}

MyDate.java

// package ex3;

import java.util.Scanner;

public class MyDate {

    private int day;

    private int month;

    private int year;

    public int getDay() {

        return day;

    }

    public void setDay(int day) {

        this.day = day;

    }

    public int getMonth() {

        return month;

    }

    public void setMonth(int month) {

        this.month = month;

    }

    public int getYear() {

        return year;

    }

    public void setYear(int year) {

        this.year = year;

    }

    public MyDate(int day, int month, int year) {

        super();

        this.day = day;

        this.month = month;

        this.year = year;

    }

    public MyDate() {

        super();

    }

    public MyDate(String strDate) {

        String list[] = strDate.split(" ");

        String months[] = { "January", "Feburary", "March", "April", "May", "June", "July", "August", "September",

                "October", "November", "December" };

        // set day

        if (list[1].equals("1st") == true) {

            this.day = 1;

        } else if (list[1].equals("2nd") == true) {

            this.day = 2;

        } else if (list[1].equals("3rd") == true) {

            this.day = 3;

        } else {

            String days[] = list[1].split("th");

            this.day = Integer.parseInt(days[0]);

        }

        // set month

        for (int i = 0; i < 12; i++) {

            if (list[0].equals(months[i]) == true) {

                this.month = i + 1;

                break;

            }

        }

        // set year

        this.year = Integer.parseInt(list[2]);

    }

    public void accept() {

        System.out.println("enter date:");

        Scanner keyboard = new Scanner(System.in);

        String date = keyboard.nextLine();

        String ele[] = date.split(" ");

        String strMonth = ele[0];

        String strDay = ele[1];

        String strYear = ele[2];

        String months[] = { "January", "Feburary", "March", "April", "May", "June", "July", "August", "September",

                "October", "November", "December" };

        int i;

        for (i = 0; i < 12; i++) {

            if (strMonth.equals(months[i])) {

                this.month = i + 1;

                break;

            }

        }

        // set year

        int year = Integer.parseInt(strYear);

        this.year = year;

        // set day

        if (strDay.equals("1st") == true) {

            this.day = 1;

        } else if (strDay.equals("2nd") == true) {

            this.day = 2;

        } else if (strDay.equals("3rd") == true) {

            this.day = 3;

        } else {

            String days[] = strDay.split("th");

            this.day = Integer.parseInt(days[0]);

        }

    }

}

Kết quả: Ảnh có chứa văn bản

Mô tả được tạo tự động

Phần 2:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Trả lời:

Trong java, chúng ta có thể nạp chồng phương thức main(), nghĩa là trong một class có thể có nhiều phương thức main() bằng cách nạp chồng phương thức. Khi biên dịch thì máy chỉ biên dịch phương thức main() có đối số truyền vào là một mảng các chuỗi ký tự.Đó là main(String[] args){}.