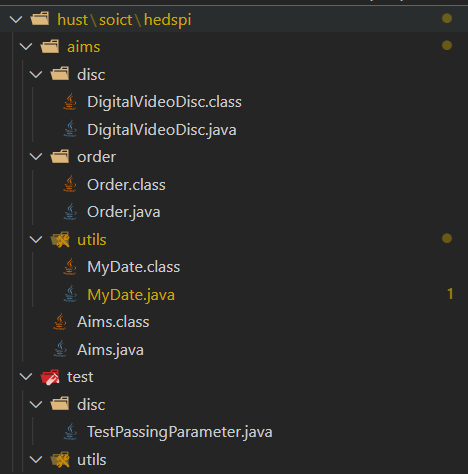
Báo cáo Lab05

Phùng Ngọc Vinh – 20194719

Bài 1:



Bài 2:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Bài 3:

Order.java

package order;

import disc.DigitalVideoDisc;

import utils.MyDate;

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("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

    }

    public DigitalVideoDisc getALuckyitem() {

        int min = 0;

        int max = qtyOrdered - 1;

        int range = max - min + 1;

        int lucky = (int) (Math.random() \* range) + min;

        System.out.println("\*\*\* Lucky number is: " + lucky + " \*\*\*");

        return itemOdered[lucky - 1];

    }

}

DigitalVideoDisc.java

package disc;

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;

    }

    public boolean search(String title) {

        String titleCopy = new String(title);

        titleCopy = titleCopy.toLowerCase();

        String titleList[] = titleCopy.split(" ");

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

            if (this.title.toLowerCase().indexOf(titleList[i]) == -1) {

                return false;

            }

        }

        return true;

    }

}

DiskTest.java

import disc.DigitalVideoDisc;

import order.Order;

import utils.MyDate;

public class DiskTest {

    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);

        DigitalVideoDisc lucky = anOrder.getALuckyitem();

        System.out.println("Lucky item is: " + lucky.getTitle());

        lucky.setCost(0);

        anOrder.printFull();

        System.out.println(lucky.search("The Lion King"));

    }

}

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Bài 4:

ConcatenationinLoops.java

import java.util.Random;

public class ConcatenationinLoops {

    public static void main(String[] args) {

        Random r = new Random(123);

        long start = System.currentTimeMillis();

        String s = "";

        for (int i = 0; i < 65536; i++) {

            s += r.nextInt(2);

        }

        System.out.println(System.currentTimeMillis() - start);

        r = new Random(123);

        start = System.currentTimeMillis();

        StringBuilder sb = new StringBuilder();

        for (int i = 0; i < 65536; i++) {

            sb.append(r.nextInt(2));

        }

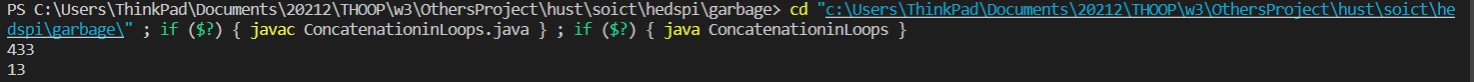
        s = sb.toString();

        System.out.println(System.currentTimeMillis() - start);

    }

}

Kết quả:



GarbageCreator.java

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class GarbageCreator {

    public static void main(String[] args) {

        long start = System.currentTimeMillis();

        String s = "";

        try {

            File fin = new File("C:\\Users\\ThinkPad\\Documents\\text.txt");

            Scanner myScaner = new Scanner(fin);

            while (myScaner.hasNextLine()) {

                s += myScaner.nextLine();

            }

            myScaner.close();

        } catch (FileNotFoundException e) {

            System.out.println("An error");

            e.printStackTrace();

        }

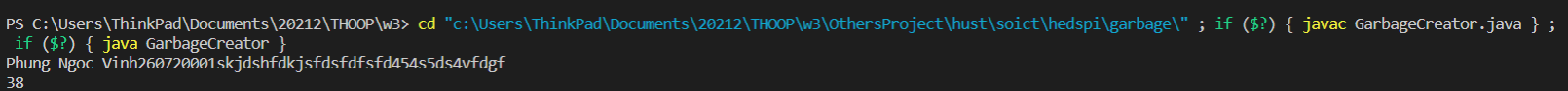
        System.out.println(s);

        System.out.println(System.currentTimeMillis() - start);

    }

}

Kết quả:



NoGarbage.java

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.util.Scanner;

public class NoGarbage {

    public static void main(String[] args) {

        long start = System.currentTimeMillis();

        StringBuffer sb = new StringBuffer();

        try {

            Scanner scan = new Scanner(new FileReader("C:\\Users\\ThinkPad\\Documents\\text.txt"));

            while (scan.hasNextLine()) {

                sb.append(scan.nextLine());

            }

            scan.close();

        } catch (FileNotFoundException e) {

            System.out.println("An error");

            e.printStackTrace();

        }

        System.out.println(sb);

        System.out.println(System.currentTimeMillis() - start);

    }

}

Kết quả:

