

MỤC LỤC

LỜI NÓI ĐẦU	2
YÊU CẦU CHUNG ĐỐI VỚI MỖI BUỔI THỰC HÀNH.....	3
BÀI TẬP THỰC HÀNH 1: LÀM QUEN VỚI IDE	4
BÀI TẬP THỰC HÀNH 2: CÁC THÀNH PHẦN CƠ SỞ TRONG JAVA	6
BÀI TẬP THỰC HÀNH 3: GÓI VÀ LỚP ĐỐI TƯỢNG TRONG JAVA	9
BÀI TẬP THỰC HÀNH 4: GÓI, LỚP VÀ GIAO DIỆN TRONG JAVA.....	14
BÀI TẬP THỰC HÀNH 5: TOÁN TỬ TẠO LẬP VÀ QUAN HỆ KẾ THỪA.....	19
BÀI TẬP THỰC HÀNH 6: XỬ LÝ NGOẠI LỆ VÀ GIAO DIỆN TRONG JAVA	23
BÀI TẬP THỰC HÀNH 7: MẢNG TRONG JAVA.....	25
BÀI TẬP THỰC HÀNH 8: CÁC LỚP CƠ SỞ TRONG GÓI JAVA.LANG	31
BÀI TẬP THỰC HÀNH 9: XỬ LÝ CHUỖI.....	35
BÀI TẬP THỰC HÀNH 10: BÀI TẬP TỔNG HỢP	39

LỜI NÓI ĐẦU

Giáo án này chúng tôi biên soạn dựa trên các kiến thức trong giáo án lý thuyết của môn học. Các bài tập được biên soạn theo tuần tự các chương tương ứng với phần lý thuyết. Các bài tập được thiết kế theo mức độ hoàn thiện dần. Mỗi chương đều được bố cục theo các nội dung gồm mục đích, yêu cầu, nội dung các bài tập thực hành và cuối cùng là bài tập mẫu tương ứng với phần kiến thức trong chương.

Rất mong nhận được các ý kiến góp ý của các bạn sinh viên và các đồng nghiệp để giáo án thực hành môn học Lập trình hướng đối tượng với Java ngày càng hoàn thiện hơn.

Xin trân thành cảm ơn!

Nhóm biên soạn

YÊU CẦU CHUNG ĐỐI VỚI MỖI BUỔI THỰC HÀNH

❖ Yêu cầu chung đối với từng buổi thảo luận:

○ Đối với sinh viên

- Có khả năng thực hành thành thạo một ngôn ngữ lập trình
- Chuẩn bị và cài đặt trước các nội dung thực hành của mỗi buổi thực hành
- Trao đổi với giáo viên hướng dẫn những vấn đề chưa hiểu
- Hoàn thành các nội dung thực hành theo yêu cầu của giáo viên.

○ Đối với giảng viên:

- Tóm tắt các lý thuyết liên quan đến từng buổi thực hành.
- Cụ thể hóa các yêu cầu đối với nội dung thực hành từng buổi thực hành.
- Giao nhiệm vụ thực hành cho sinh viên.
- Giám sát và điều hành các hoạt động diễn ra trong buổi thực hành.
- Hướng dẫn, trả lời, giải đáp các thắc mắc của sinh viên.
- Đánh giá, nhận xét kết quả, ý thức của sinh viên cuối mỗi buổi thực hành.

❖ Hình thức đánh giá:

- Kết quả thực hành của sinh viên trong từng buổi có thể được lưu lại và lấy trung bình xem như một điểm kiểm tra.

❖ Thời gian thực hành

- Theo lịch trình môn học.

BÀI TẬP THỰC HÀNH 1: LÀM QUEN VỚI IDE

1. Mục tiêu:

Về kiến thức:

- Sinh viên làm quen với môi trường tích hợp phát triển IDE (netbean hoặc eclipse)
- Viết chương trình ứng dụng đơn giản với các thao tác nhập xuất dữ liệu chuẩn

Về kỹ năng:

- Sinh viên phải biết cài đặt một IDE (netbean, eclipse) và sử dụng được để chạy một chương trình java.

Về thái độ:

- + Tự giác chuẩn bị các câu hỏi và bài tập.
- + Thực hiện các bài tập trên máy tính.

2. Yêu cầu

- Sinh viên chuẩn bị trước các bài tập thực hành,
- Thực hành các bài tập trong chương.

3. Nội dung thực hành

3.1 Bài thực hành mẫu

Bài 1:

Viết chương trình nhập vào 2 số nguyên từ bàn phím. In ra tổng, hiệu của 2 số đó.

Chương trình:

```
//Bai1.java
```

```
Import java.util.Scanner;
```

```
public class Bai1{
```

```
    public static void main(String args[]){
```

```
        Scanner w = new Scanner(System.in);
```

```
        int a=0, b=0;
```

```
        System.out.println("Nhap so a=");
```

```
        a=w.nextInt();
```

```
        System.out.println("Nhap so b=");
```

```
        b=w.nextInt();
```

```
        System.out.println("tong a+b="+ (a+b) + "hieu a-b="+ (a-b));
```

```
    }
```

```
}
```

3.2 Bài thực hành cơ bản

Bài 1

1. Cài đặt JDK và cài đặt NetBean (hoặc eclipse)
2. Thực hiện tạo mới 1 project java in ra màn hình kết quả dòng chữ “xin chào” trong hàm main()
3. Làm quen với các thành phần NetBean (eclipse)

Bài 2

Viết chương trình:

1. Khai báo (import) lớp Scanner trong gói java.util
2. Sử dụng phương thức nextLine () của lớp để nhập vào một xâu và in xâu vừa nhập ra màn hình.

Bài 3

Viết chương trình:

1. Sử dụng lớp Scanner trong gói java.util
2. Sử dụng phương thức nextInt() để nhập vào hai số a, b in kết quả tổng, hiệu của a và b ra màn hình.

3.3 Bài tập thực hành nâng cao

Bài 1

Viết chương trình nhập số cạnh của đa giác, chiều dài các cạnh sau đó hiển thị lên màn hình tất cả các thông tin đã nhập, tính chu vi đa giác.

Bài 2

Viết chương trình: Nhập vào 2 số kiểu số thực float, double (sử dụng với các phương thức nextFloat(), nextDouble(), in ra các kết quả của các phép tính, tổng, hiệu, tích, thương, lấy phần dư của các số đó.

BÀI TẬP THỰC HÀNH 2: CÁC THÀNH PHẦN CƠ SỞ TRONG JAVA

1. Mục tiêu:

Về kiến thức:

- Sinh viên có thể lập trình giải các bài toán đơn giản bằng Java, vận dụng được các thức cơ bản trong java: cấu trúc một chương trình java, khai báo biến, các kiểu dữ liệu nguyên thủy, các phép toán cơ bản, các cú pháp điều khiển, các cú pháp chuyển vị...

Về kỹ năng:

- Sinh viên phải biết cài đặt NetBean IDE và sử dụng được Netbean để chạy một chương trình java.

Về thái độ:

- Tự giác chuẩn bị các câu hỏi và bài tập.
- Thực hiện các bài tập trên máy tính.

2. Yêu cầu

- Sinh viên chuẩn bị trước các bài tập thực hành,
- Thực hành các bài tập trong chương.

3. Nội dung thực hành

3.1 Bài thực hành mẫu

Bài 1: Xây dựng lớp ToanHoc chứa phương thức tính tổng dãy và hiển thị kết quả:

$$S = \frac{1}{2} + \frac{2}{3} + + \frac{n-1}{n}$$

Chương trình

//ToanHoc.java

Import java.util.Scanner;

public class ToanHoc{

public static double tinhTong(int n){

double S=0.0;

for(int i=1; i<n;i++){

S=S + i/(i+1);

}

return S;

}

```

public static void main(String args[]){
    int n=0;
    Scanner w = new Scanner(System.in);
    System.out.println("Nhap n=");
    n=w.nextInt();
    System.out.println("Tong S="+tinhTong(n));
}
}

```

3.2 Bài thực hành cơ bản

Bài 1

Viết chương trình in ra giá trị tăng giảm của biến **i** cho trước.

Bài 2

Viết các chương trình nhập vào một số nguyên n và in ra số nhị phân tương ứng với n.

Bài 3:

Nhập vào ngày, tháng, năm và cho biết đó là ngày thứ bao nhiêu trong năm (biết rằng một năm có 365 ngày hoặc 336 ngày với năm nhuận).

Bài 4

Viết chương trình tìm các số nguyên tố nhỏ hơn một số n cho trước

- Chỉ sử dụng chu trình **for**
- Chỉ sử dụng chu trình **while**
- Chỉ sử dụng chu trình **do-while**

Bài 5:

Xây dựng lớp **ToanHoc**, với n là số nguyên dương.

- Kiểm tra xem một số nguyên n có phải là số nguyên tố hay không ?
- Kiểm tra xem một nguyên n có phải là số hoàn hảo hay không ?
- In ra các số chính phương $\leq n$ (n nhập từ bàn phím).
- Tính phần tử thứ n của dãy Fibonacci: `public static int tinhFibo(int n)`

Với công thức truy_hồi được tính như sau:

$$L_n := \begin{cases} 2 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ L_{n-1} + L_{n-2} & \text{if } n > 1. \end{cases}$$

Bài 6

1. Tính tổng dãy

$$S=1+2-3+\dots+(-1)^{n+1}.n$$

2. Tính tổng dãy:

$$S=1!+2!+3!+\dots+n!$$

3. Nhập n, nếu n lẻ : tính tổng các số lẻ <n, nếu n chẵn: tính tổng các số chẵn <n.

4. Tính tổng dãy

$$S = \frac{1}{2} + \frac{2}{3} + \dots + \frac{n-1}{n}$$

3.3 Bài thực hành nâng cao

Bài 1:

Tính tổng dãy:

$$S = \frac{1!}{x+k} + \frac{2!}{x^2-k^2} + \dots + \frac{n!}{x^n + (-1)^{n+1}.k^n}$$

BÀI TẬP THỰC HÀNH 3: GÓI VÀ LỚP ĐỐI TƯỢNG TRONG JAVA

1. Mục tiêu:

Về kiến thức:

- Sinh viên viết được các chương trình giải các bài toán theo cách tiếp cận hướng đối tượng với Java. Vận dụng được các kiến thức về lớp, đối tượng, phương thức, truyền thông điệp trong java.

Về kỹ năng:

- Sinh viên phải biết cách tạo đối tượng, lớp, biết tư duy bài toán theo tư tưởng hướng đối tượng. Biết sử dụng và cài đặt NetBean IDE và sử dụng được Netbean để chạy một chương trình java.

Về thái độ:

- Tự giác chuẩn bị các câu hỏi và bài tập.
- Thực hiện các bài tập trên máy tính.

2. Yêu cầu

- Sinh viên chuẩn bị trước các bài tập thực hành,
- Thực hành các bài tập trong chương.

3. Nội dung thực hành

3.1 Bài thực hành mẫu

Bài 1:

Viết chương trình xây dựng lớp KháchHang với các thuộc tính: Họ tên, địa chỉ, số điện thoại, email và các phương thức gồm:

- Các toán tử tạo lập
- Nhập thông tin khách hàng
- Hiển thị thông tin khách hàng
- Phương thức main, nhập vào thông tin của một khách hàng và hiển thị lên màn hình thông tin vừa nhập

Chương trình

```
// KháchHang.java
```

```

import java.io.*;
public class KhachHang{
    String hoTen;
    String diaChi;
    String sdt;
    String email;
    public KhachHang(){
        //Toan tu tao lap mac dinh
    }
    public KhachHang(String ht,String dc,String sdthoai,String emailAdd){
        hoTen=ht;
        diaChi=dc;
        sdt=sdthoai;
        email=emailAdd;
    }
    public static String nhapString(){
        String s;
        DataInputStream stream=new DataInputStream(System.in);
        try{
            s=stream.readLine();
            s=s.trim();
            s=s.toLowerCase();
        }catch(IOException e){
            s="";
        }
        return s;
    }
    public void nhapKH(){
        String str;
        //nhap ho ten
        System.out.print("-Ho ten : ");
        str=nhapString();
        this.hoTen=str;
        //nhap dia chi
        System.out.print("-Dia chi : ");
        str=nhapString();
        this.diaChi=str;
        //nhap so dien thoai
        System.out.print("-So dien thoai : ");
        str=nhapString();
    }
}

```

```

        this.sdt=str;
        //nhap dia chi email
        System.out.print("--Dia chi Email : ");
        str=nhapString();
        this.email=str;
    }
    public void inKH(){
        System.out.println("--Ho ten:"+this.hoTen);
        System.out.println("--Dia chi:"+this.diaChi);
        System.out.println("--So dien thoai:"+this.sdt);
        System.out.println("--Email:"+this.email);
    }
    public static void main(String[] args){
        KhachHang kh=new KhachHang();
        kh.nhapKH();
        kh.inKH();
        KhachHang k1=new KhachHang("nam","k1b
cntt","862422","namcntt@yahoo.com");
        k1.inKH();
    }
}

```

3.2 Bài thực hành cơ bản

Bài 1

Viết chương trình:

1. Xây dựng lớp *Ngươi* gồm;
 - ✓ Các thuộc tính chung : *hoTen*, *diaChi*, *namSinh*
 - ✓ Các phương thức:
 - Phương thức: `public void nhap TT()` { // nhập thông tin cho *Ngươi* }
 - Phương thức: `Public void inTT()` { // in thông tin cho *Ngươi* }, `public void nhapTT()` { // nhập thông tin từ bàn phím cho các thông tin của *Ngươi* }
2. Xây dựng lớp *VanDung* chứa phương thức `main()`: Cho phép nhập thông tin của 1 người và hiển thị thông tin của người vừa nhập ra màn hình.

Bài 2:

Thư viện của trường đại học KHTN có nhu cầu cần quản lý việc mượn sách. Sinh viên đăng ký và tham gia mượn sách thông qua các thẻ mượn mà thư viện đã thiết kế.

- Với mỗi thẻ mượn, có các thông tin sau: số phiếu mượn, ngày mượn, hạn trả, số hiệu sách, và các thông tin riêng về mỗi sinh viên đó.
- Các thông tin riêng về mỗi sinh viên đó bao gồm: Họ tên, năm sinh, tuổi, lớp.

1. Hãy xây dựng các lớp: SinhVien để quản lý các thông tin riêng về mỗi sinh viên, lớp TheMuon để quản lý việc mượn sách của mỗi đọc giả.
2. Viết chương trình nhập và hiển thị thông tin của một thẻ mượn.

Bài 3:

Để quản lý các biên lai thu tiền điện, người ta cần các thông tin như sau:

Với mỗi biên lai, có các thông tin sau: thông tin về hộ sử dụng điện, chỉ số cũ, chỉ số mới, số tiền phải trả của mỗi hộ sử dụng điện

Các thông tin riêng của mỗi hộ sử dụng điện gồm: Họ tên chủ hộ, số nhà, mã số công tơ của hộ dân sử dụng điện.

1. Hãy xây dựng các lớp: **KhachHang** để lưu trữ các thông tin riêng của mỗi hộ sử dụng điện, lớp **BienLai** để quản lý việc sử dụng và thanh toán tiền điện của các hộ dân. Xây dựng các phương thức nhập, và hiển thị thông tin của mỗi hộ sử dụng điện.

2. Cài đặt chương trình thực hiện các công việc sau:

+ Nhập vào các thông tin cho n hộ sử dụng điện

+ Tính tiền điện phải trả cho mỗi hộ dân, biết rằng tiền phải trả được tính theo công thức sau: $Số tiền phải trả = (Số mới - số cũ) * 850000$.

3.3 Bài thực hành nâng cao

Bài 1:

1. Hãy xây dựng lớp DaGiac gồm có:

Các thuộc tính:

+ Số cạnh của đa giác

+ Mảng các số thực chứa kích thước các cạnh của đa giác

Các phương thức:

+ Nhập vào giá trị cho các cạnh của đa giác

+ Tính chu vi của đa giác

- + Hiển thị giá trị các cạnh của đa giác.
- 2. Xây dựng lớp TamGiac kế thừa từ lớp DaGiac, thực hiện ghi đè các phương thức của lớp TamGiac lên các phương thức của lớp DaGiac.
- 3. Xây dựng hàm main() nhập kích thước của một tam giác và hiển thị chu vi, diện tích của tam giác đó.

BÀI TẬP THỰC HÀNH 4: GÓI, LỚP VÀ GIAO DIỆN TRONG JAVA

1. Mục tiêu:

Về kiến thức:

- Sinh viên viết được các chương trình giải các bài toán theo cách tiếp cận hướng đối tượng với Java. Vận dụng được các kiến thức về lớp, đối tượng, phương thức, truyền thông điệp trong java.

Về kỹ năng:

- Sinh viên phải biết cách tạo đối tượng, lớp, biết tư duy bài toán theo tư tưởng hướng đối tượng. Biết sử dụng Cài đặt NetBean IDE và sử dụng được Netbean để chạy một chương trình java.

Về thái độ:

- Tự giác chuẩn bị các câu hỏi và bài tập.
- Thực hiện các bài tập trên máy tính.

2. Yêu cầu

- Sinh viên chuẩn bị trước các bài tập thực hành,
- Thực hành các bài tập trong chương.

3. Nội dung thực hành

3.1 Bài thực hành mẫu

Bài 1: Tạo 1 lớp Person chứa các thông tin về người, tạo tiếp 2 lớp Students và Teachers kế thừa lớp Person, tạo lớp Execute chứa hàm main để chạy chương trình.

Chương trình:

```
import corejava.*;
abstract class Person{
    //cai nay goi la cac property hay state-thuoc tinh cua doi tuong
    String hoten;
    int age;
    String diachi;
    int luong;
    //cac constructor
    public Person(int age)
    {
```

```

        this.age = age;
    }
    //cac method hay behavior-hanh vi cua doi tuong
    public void Nhap()
    {
        hoten = Console.readLine("Nhap ho ten:");
        diachi = Console.readLine("Nhap dia chi:");
    }

    public abstract void In();
    public abstract int Tinhluong();
import corejava.*;
class Students extends Person
{
    int MaSV, Malop;
    public void Nhap()
    {
        super.Nhap();
        MaSV = Console.readInt("Nhap ma SV:");
        Malop = Console.readInt("Nhap ma lop:");
    }
    public void In()
    {
        System.out.println(hoten);
        System.out.println(diachi);
        System.out.println(MaSV);
        System.out.println(Malop);
    }
    public int Tinhluong()
    {
        return 150000;
    }
}
import corejava.*;
class Teachers extends Person
{
    int Makhoa;
    public void Nhap()
    {
        super.Nhap();

```

```

        Makhoa = Console.readInt("Nhap ma khoa:");
    }
    public void In()
    {
        System.out.println(hoten);
        System.out.println(diachi);
        System.out.println(Makhoa);
    }
    public int Tinhluong()
    {
        return 500000;
    }
}

```

```

--lop Execute-
import corejava.*;
class Execute
{
    public static void main(String args[])
    {
        Students st = new Students();
        st.Nhap();
        st.In();
        st.luong=st.Tinhluong();
        Teachers tc = new Teachers();
        tc.Nhap();
        tc.In();
        tc.luong=tc.Tinhluong();
    }
}

```

3.2 Bài thực hành cơ bản

Bài 1

Xây dựng lớp ToanHoc có một biến thành phần là số nguyên n, với phương thức tĩnh (static) sau:

1. Kiểm tra xem một số nguyên k có phải là số nguyên tố hay không ?

```
public static boolean kiểmtraNguyenTo(int k)
```

2. Kiểm tra xem một nguyên n có phải là số hoàn hảo hay không ?

public static boolean kiểmtraHoanHao(int n)

3. Kiểm tra xem một nguyên n có phải là số chính phương hay không ?

public static boolean kiểmtraSoChinhPhuong(int n)

4. Tính phần tử thứ n của dãy Fibonacci: public static int tinhFibo(int n)

Với công thức truy hồi được tính như sau:

$$L_n := \begin{cases} 2 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ L_{n-1} + L_{n-2} & \text{if } n > 1. \end{cases}$$

5. Tính tổng dãy

$$S = \frac{1}{2} + \frac{2}{3} + \dots + \frac{n-1}{n}$$

public static double tinhTong(int n)

6. Tính tổng dãy

$$S = \frac{1!}{x+k} + \frac{2!}{x^2-k^2} + \dots + \frac{n!}{x^n + (-1)^{n+1}.k^n}$$

public static double tinhTong(int n)

7. Tính tổng dãy

$$S = 1 + 2 - 3 + \dots + (-1)^{n+1}.n$$

public static double tinhTong(int n)

8. Tính tổng dãy:

$$S = 1! + 2! + 3! + \dots + n!$$

public static double tinhTong(int n)

Xây dựng lớp VanDung có phương thức main() sử dụng các phương thức trong lớp ToanHoc ở trên.

Bài 2:

Viết chương trình xây dựng lớp Diem gồm:

1. Các thuộc tính double x,y tương ứng là hai giá trị hoành độ và tung độ và các phương thức:
2. Các phương thức tạo lập:

- a. *public Diem(){}*
- b. *public Diem(double x,double y){x=x1;y=y1}*
3. Các phương thức
 - a. *public void inTTDiem(){ // in giá trị x,y ra màn hình }*
 - b. *public double tinhKhoangCach(Diem d) {tính khoảng cách giữa điểm hiện thời và đối số d}*
4. Cài đặt lớp Main thực hiện tạo ra hai điểm $D1=new Diem(3,7); D2= new Diem(9,3)$ và in khoảng cách của hai điểm trên.

3.3 Bài thực hành nâng cao

Bài 1

Viết chương trình:

1. Xây dựng lớp SoPhuc gồm:
 - a. Các thuộc tính riêng gồm: *phanThuc, phanAo* kiểu *double*;
 - b. Các phương thức:
 - + Các toán tử tạo lập : *SoPhuc(), SoPhuc(float pt, float pa)*
 - + Phương thức nhập vào một số phức: *: void nhapSoPhuc()*
 - + Phương thức hiển thị một số phức: *void inSoPhuc()*
 - + Phương thức cộng hai số phức : *SoPhuc congSoPhuc(SoPhuc sp)*
 - + Phương thức nhân hai số phức: *SoPhuc nhanSoPhuc(SoPhuc sp)*
2. Cài đặt chương trình(lớp Main) nhập vào hai số phức A và B, sau đó tính số phức tổng $C1=A.congSoPhuc(B)$, nhân $C=A.nhanSoPhuc(B)$ rồi hiển thị kết quả ra màn hình như sau:

Tổng: $(a + b.i) + (c + d.i) = (a + c) + (b + d).i$

Tích: $(a + b.i)(c + d.i) = (a.c - b.d) + (b.c + a.d).i$

Chia :

$$\frac{a + b * i}{c + d * i} = \frac{(a + b * i)(c - d * i)}{(c + d * i)(c - d * i)} = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2} * i$$

BÀI TẬP THỰC HÀNH 5: TOÁN TỬ TẠO LẬP VÀ QUAN HỆ KẾ THỪA

1. Mục tiêu:

Về kiến thức:

- Sinh viên viết được các chương trình giải các bài toán theo cách tiếp cận hướng đối tượng với Java. Vận dụng được các kiến thức về lớp, đối tượng, phương thức, truyền thông điệp trong java, các khái niệm về toán tử tạo lập và lập trình sử dụng quan hệ kế thừa trong java.

Về kỹ năng:

- Sinh viên phải biết cách tạo đối tượng, lớp, biết tư duy bài toán theo tư tưởng hướng đối tượng. Biết sử dụng cài đặt NetBean IDE và sử dụng được Netbean để chạy một chương trình java. Biết xây dựng các ví dụ sử dụng quan hệ kế thừa trong java.

Về thái độ:

- Tự giác chuẩn bị các câu hỏi và bài tập.
- Thực hiện các bài tập trên máy tính.

2. Yêu cầu

- Sinh viên chuẩn bị trước các bài tập thực hành,
- Thực hành các bài tập trong chương.

3. Nội dung thực hành

3.1 Bài thực hành mẫu

Bài 1:

Xây dựng giao diện và các lớp để thực thi giao diện này.

Chương trình:

```
public interface Product  
{
```

```
    //hai state duoi day la final, tuc la lop implements khong duoc phep doi gia tri  
    static string maker = "My Corp";  
    static string phone = "555-7767";  
    //behavior duoi day la abstract, tuc la khong co noi dung
```

```

        public int getPrice(int id);
    }
    public class Shoe implements Product
    {
        public int getPrince(int id)
        {
            return (id==1)?5:10;
        }
        public String getMaker()
        {
            return maker;
        }
    }
    package Transport;
    public class Car
    {
        public String manufacturer;
        public int year;
    }
    package Transport;
    public class Bicycle
    {
        public int cost;
        public Bicycle(int cost)
        {
            this.cost = cost;
        }
    }
    import Transport.*;
    class TestProgram
    {
        public static void main(String args[])
        {
            Car myCar = new Car();
            myCar.manufacturer = "Toyota";
            Bicycle myBicycle = new Bicycle(1500);
        }
    }

```

3.2 Bài thực hành cơ bản

Bài 1:

Viết chương trình:

1. Xây dựng lớp *Ngnoi* gồm;

✓ Các thuộc tính chung : *hoTen*, *diaChi*, *namSinh*

✓ Các phương thức:

- Các toán tử tạo lập : `public Ngnoi(){} , public Ngnoi(String ht,String dc,int ns){// nội dung tạo lập }`

- Phương thức: `public void nhap TT(){// nhập thông tin cho Ngnoi}`

- Phương thức: `Public void inTT(){//in thông tin cho Ngnoi}`, `public void nhapTT(){// nhập thông tin từ bàn phím cho các thông tin của Ngnoi }`,

4. Xây dựng lớp *NhanSu* kế thừa từ lớp *Ngnoi* và có thêm các thuộc tính:

maNhanSu, *heSoChucVu*, *heSoLuong* . Các phương thức: thực hiện viết đề các

phương thức: *static int luongCoBan* và *nhapTTNhanSu()*, *inTTNhanSu()*

phương thức tĩnh *static void setLuongCoBan(int lcb){luongCoBan=lcb}*

public double tinhLuong(){ }

theo tiêu chí:

$luong = (heSoLuong + heSoChucVu) * luongCoBan - heSoLuong * luongCoBan * 5\%$,

Tại lớp *Main* tạo ra hai đối tượng thuộc lớp *Ngnoi*, hai đối tượng thuộc lớp *NhanSu* nhập thông tin *luongCoBan* cho lớp *NhanSu* và nhập, in thông tin của 4 đối tượng trên

Bài 2

Xây dựng lớp *Ngnoi* gồm có các thuộc tính chung : *hoTen*, *diaChi*, *namSinh* và các phương thức:

+Các toán tử tạo lập : `public Ngnoi(){} , public Ngnoi(String ht,String dc,int ns){// nội dung tạo lập }`

+Phương thức: `public void nhap TT(){// nhập thông tin cho Ngnoi}`

+Phương thức: `Public void inTT(){//in thông tin cho Nguoi}, public void nhapTT(){// nhập thông tin từ bàn phím cho các thông tin của Nguoi }`

Xây dựng lớp `SinhVien` kế thừa từ lớp người thêm các thuộc tính `maSV`, `tenLop`, `diem1`, `diem2`, `diem3` và các phương thức:

+Các toán tử tạo lập.

+Tính điểm TB: `public float tinhDiemTrungBinh(){};`

+viết đề hai phương thức `nhapTT()`, `inTT()`

Tại lớp `Main` tạo ra một đối tượng thuộc lớp `Nguoi`, một đối tượng thuộc lớp `SinhVien` nhập thông tin `diem1`, `diem2`, `diem3` cho lớp `SinhVien` và nhập, in thông tin của 2 đối tượng trên.

3.3 Bài thực hành nâng cao

Bài 1:

Để quản lý các hộ dân trong một khu phố, người ta quản lý các thông tin như sau:

- Với mỗi hộ dân, có các thuộc tính:
 - + Số thành viên trong hộ (số người)
 - + Số nhà của hộ dân đó (Số nhà được gán cho mỗi hộ dân)
 - + Thông tin về mỗi cá nhân trong hộ gia đình.
- Với mỗi cá nhân, người ta quản lý các thông tin như: họ và tên, tuổi, năm sinh, nghề nghiệp.

1. Hãy xây dựng các lớp: `NhanSu` để quản lý thông tin về mỗi cá nhân; lớp `HoDan` để quản lý thông tin về các hộ gia đình. Viết các phương thức để nhập, hiển thị thông tin cho mỗi cá nhân.

2. Xây dựng hàm `main()` cho phép nhập và hiển thị thông tin của 1 nhân sự.

BÀI TẬP THỰC HÀNH 6: XỬ LÝ NGOẠI LỆ VÀ GIAO DIỆN TRONG JAVA

1. Mục tiêu:

Về kiến thức:

- Sinh viên viết được các chương trình giải các bài toán theo cách tiếp cận hướng đối tượng với Java. Vận dụng được các kiến thức về lớp, đối tượng, phương thức, truyền thông điệp trong java. Biết cách xử lý ngoại lệ và xây dựng giao diện trong java.

Về kỹ năng:

- Sinh viên phải biết cách tạo đối tượng, lớp, biết tư duy bài toán theo tư tưởng hướng đối tượng. Biết cách bắt và xử lý ngoại lệ trong java.

Về thái độ:

- Tự giác chuẩn bị các câu hỏi và bài tập.
- Thực hiện các bài tập trên máy tính.

2. Yêu cầu

- Sinh viên chuẩn bị trước các bài tập thực hành,
- Thực hành các bài tập trong chương.

3. Nội dung thực hành

3.1 Bài thực hành mẫu

Bài 1:

Viết chương trình chạy tiếp và không thoát ra, ta đón "bắt" ngoại lệ này, đưa ra biến e, cuối cùng in e (để xem là ngoại lệ gì).

```
class FinallyDemo
{
    String name;
    int no1,no2;
    FinallyDemo(String args[])
    {
        try
        {
            name=new String("Aptech Limited");
            no1=Integer.parseInt(args[0]);
```

```

no2=Integer.parseInt(args[1]);
System.out.println(name);
System.out.println("Division Result is" + no1/no2);
}
catch(ArithmeticException i)
{
System.out.println("Cannot Divide by zero");
}
finally
{
name=null; // clean up code
System.out.println("Finally executed");
}
}
public static void main(String args[])
{
new FinallyDemo(args);
} }

```

3.2 Bài thực hành cơ bản

Bài 1

Xét thao tác vào ra (I/O) trong một tập tin. Nếu việc chuyển đổi kiểu dữ liệu không thực hiện đúng, một ngoại lệ sẽ xảy ra và chương trình bị hủy mà không đóng tập tin lại. Lúc đó tập tin dễ bị hư hại và các nguồn tài nguyên được cấp phát cho tập tin không được trả lại cho hệ thống.

Bài 2:

Viết chương trình gây ra ngoại lệ khi người sử dụng không nhập tham số nào vào từ dòng lệnh. Chương trình phải hiện thị số tham số nếu có tham số được nhập vào từ dòng lệnh.

Bài 3:

Viết chương trình gây ra ngoại lệ, nếu không có số nào được nhập vào từ dòng lệnh. Ngược lại, chương trình hiển thị giá trị lập phương của số nhập vào

3.3 Bài thực hành nâng cao

Bài 3: Viết chương trình gây ra ngoại lệ nếu như lớp không thể truy nhập.

BÀI TẬP THỰC HÀNH 7: MẢNG TRONG JAVA

1. Mục tiêu:

Về kiến thức:

- Sinh viên viết chương trình giải các bài toán về mảng một cách thành thạo. Khai thác được các thuộc tính và các phương thức của lớp mảng.

Về kỹ năng:

- Sinh viên phải biết sử dụng mảng trong các bài tập.

Về thái độ:

- Tự giác chuẩn bị các câu hỏi và bài tập.
- Thực hiện các bài tập trên máy tính.

2. Yêu cầu

- Sinh viên chuẩn bị trước các bài tập thực hành,
- Thực hành các bài tập trong chương.

3. Nội dung thực hành

3.1 Bài thực hành mẫu

Bài 1

Xây dựng lớp QLKhachHang (mỗi khách hàng gồm các thông tin cần quản lý như bài tập mẫu chương 3) gồm:

- *Các thuộc tính: số lượng khách hàng, mảng các khách hàng*
- *Các phương thức:*
 - o *Khởi tạo*
 - o *Nhập danh sách các khách hàng*
 - o *Hiển thị lên màn hình danh sách khách hàng*
 - o *Tìm kiếm khách hàng theo họ hoặc theo tên*
 - o *Tìm kiếm khách hàng theo địa chỉ*

Chương trình

```
import java.io.*;
public class QLKhachHang{
    int soKH;
    KhachHang[] mkh;
    public QLKhachHang(){
        //Tao lap mac dinh
    }
}
```

```

public QLKhachHang(int so){
    soKH=so;
    mkh=new KhachHang[so];
}
public static int nhapInt(){
    String s;int n;
    /*
    DataInputStream stream=new DataInputStream(System.in);
    try{
        s=stream.readLine();
    }catch(IOException e){
        s="";
    }
    */
    s=nhapString();
    try{
        n=Integer.parseInt(s);
        //hoac n=Integer.valueOf(s).intValue();
    }catch(NumberFormatException nfe){
        n=0;
    }
    return n;
}
public static String nhapString(){
    String s;
    DataInputStream stream=new DataInputStream(System.in);
    try{
        s=stream.readLine();
        s=s.trim();
        s=s.toLowerCase();
    }catch(IOException e){
        s="";
    }
    return s;
}
public void nhapMangKH(){
    int n;
    //KhachHang mang[];
    System.out.print("Nhap vao so KHACH HANG : ");
    n=nhapInt();

```

```

        this.soKH=n;
        this.mkh=new KhachHang[n];
        for(int i=0;i<n;i++){
            System.out.println("-----");
            System.out.println(" * Khach hang thu "+i);
            this.mkh[i]=new KhachHang();
            this.mkh[i].nhapKH();
        }
    }

    public void inMangKH(){
        System.out.println("\n\nIN MANG KHACH HANG\n");

        for(int i=0;i<this.soKH;i++){
            System.out.println("-----");
            System.out.println(" + Khach hang thu "+i);
            this.mkh[i].inKH();
        }
        System.out.println("=====");
    }

    public void timTenHo(){
        String ten;
        int n=this.soKH;
        boolean danhDau[];
        danhDau=new boolean[n];
        System.out.print("\n\n -->> Nhap Ten hoac Ho can tim : ");
        ten=nhapString();
        for(int i=0;i<n;i++){
            if(this.mkh[i].hoTen.lastIndexOf(ten) >= 0){
                danhDau[i]=true;
            }
        }

        for(int i=0;i<n;i++){
            if(danhDau[i]){
                System.out.println();
                this.mkh[i].inKH();
            }
        }
    }

    public void timDiaChi(){
        String dchi;

```

```

        int n=this.soKH;
        boolean danhDau[];
        danhDau=new boolean[n];
        System.out.print("\n\n -->>> Nhap Dia Chi can tim : ");
        dchi=nhapString();
        for(int i=0;i<n;i++){
            if(this.mkh[i].diaChi.lastIndexOf(dchi) >= 0){
                danhDau[i]=true;
            }
        }
        for(int i=0;i<n;i++){
            if(danhDau[i]){
                System.out.println();
                this.mkh[i].inKH();
            }
        }
    }
}

public static void main(String[] args){
    QLKhachHang ql=new QLKhachHang();
    ql.nhapMangKH();
    ql.inMangKH();
    ql.timTenHo();
    ql.timDiaChi();
    //ql.inMangKH();
}
}

```

3.2 Bài thực hành cơ bản

Bài 1:

Xây dựng lớp DaySo gồm:

- ✓ Các thuộc tính riêng (private) sau đây:
 - private int n; //số phần tử của dãy
 - private int m[] ; //lưu trữ các giá trị của dãy
- ✓ Các toán tử tạo lập: DaySo(int spt), DaySo(int m1[]), DaySo() {}
- ✓ Các phương thức:
 - public void inDaySo() //in dãy ra màn hình
 - public void nhapDaySo()//nhập dãy số từ bàn phím

- `public void inSoNguyenTo(){} //in ra các số nguyên tố thuộc dãy số`
- `public void inSoHoanHao()`
- `public void sapXepTang() //sắp xếp dãy theo chiều tăng`
- `public void sapXepGiam()//sắp xếp dãy theo chiều giảm`
- `public DaySo congDay(DaySo d1)// cộng hai dãy thành một dãy số`

Bài 2:

Xây dựng lớp `MaTran` gồm:

- ✓ Các thuộc tính riêng(`private`)
 - `private int n,m; // số dòng và cột của ma trận`
 - `private double M[][]; // lưu trữ các phần tử của ma trận`
- ✓ Các phương thức tạo lập:
 - `public MaTran(){} // tạo ma trận rỗng`
 - `public MaTran(int dong,int cot) // tạo ma trận với số dòng và cột`
 - `public MaTran(double M1[][])`
- ✓ Các phương thức:
 - `public void inMaTran()// in ma trận hiện thời`
 - `public void nhapMaTran()// nhập ma trận từ bàn phím`
 - `public boolean kiemTraDoiXung()// kiểm tra tính đối xứng của mt qua đường chéo chính`
 - `public MaTran congMaTran(MaTran M1)`
 - `public MaTran nhanMaTran(MaTran M1)`

Bài 3:

Viết chương trình định nghĩa lớp **Mang** với các thuộc được định nghĩa như sau:

- + `private int n; // số phần tử mảng`
- + `private int A[]; // Danh sách các phần tử của mảng`

1. Định nghĩa lớp **Mang** ở trên có các phương thức sau:

- + **nhapDL()** để nhập giá trị cho một mảng
- + **hienThiDL()** để hiện thị dữ liệu của mảng hiện thời

- + **lietKeNT** để hiển thị tất cả các số nguyên tố của mảng ra màn hình.
- + **lietKeHH** để hiển thị tất cả các số hoàn hảo của mảng ra màn hình.
- + **sapXep** để sắp xếp các phần tử của mảng theo thứ tự giảm dần

2. Cài đặt chương trình thực hiện nhập vào một mảng gồm n số nguyên (n - nhập từ bàn phím thoả $2 \leq n \leq 100$), sau đó hiển thị danh sách tất cả các số nguyên tố của mảng trên theo thứ tự tăng dần.

3.3 Bài thực hành nâng cao

Bài 1:

Thư viện của trường đại học KHTN có nhu cầu cần quản lý việc mượn sách. Sinh viên đăng ký và tham gia mượn sách thông qua các thẻ mượn mà thư viện đã thiết kế.

- Với mỗi thẻ mượn, có các thông tin sau: số phiếu mượn, ngày mượn, hạn trả, số hiệu sách, và các thông tin riêng về mỗi sinh viên đó.
- Các thông tin riêng về mỗi sinh viên đó bao gồm: Họ tên, năm sinh, tuổi, lớp.

1. Hãy xây dựng các lớp: SinhVien để quản lý các thông tin riêng về mỗi sinh viên, lớp TheMuon để quản lý việc mượn sách của mỗi đọc giả.

2. Viết chương trình nhập và hiển thị thông tin của một danh sách gồm m thẻ mượn, đếm số lượng thẻ mượn của lớp “K3A”

BÀI TẬP THỰC HÀNH 8: CÁC LỚP CƠ SỞ TRONG GÓI JAVA.LANG

1. Mục tiêu:

Về kiến thức:

- Sinh viên viết chương trình giải các bài toán về sử dụng các lớp trong gói java.lang một cách thành thạo.

Về kỹ năng:

- Sinh viên phải biết sử dụng và vận dụng các lớp đã có trong các bài tập.

Về thái độ:

- Tự giác chuẩn bị các câu hỏi và bài tập.
- Thực hiện các bài tập trên máy tính.

2. Yêu cầu

- Sinh viên chuẩn bị trước các bài tập thực hành,
- Thực hành các bài tập trong chương.

3. Nội dung thực hành

3.1 Bài thực hành mẫu

Bài 1: Một vài phương thức trong gói java.lang.String.

```
package vidu;
import java.lang.String;
public class FirstSample {
    public static void main(String[] args) {
        String x = "Z is the set of integers";
        String x2 = "    Do Nhu Vy ";
        char y1 = x.charAt(4);
        int y2 = x.codePointAt(6);
        int y3 = x.offsetByCodePoints(2, 6);
        int y4 = x.compareTo("Z is the set of integers");
        boolean y5 = x.endsWith("r");
        boolean y6 = !x.equals("Z");
        boolean y7 = x.equalsIgnoreCase("z is the set of integers");
    }
}
```

```

        int y8 = x.indexOf("s");
        int y9 = x.lastIndexOf("s");
        int y10 = x.length();
        int y11 = x.codePointCount(4, 10);
        String y12 = x.toLowerCase();
        String y14 = x.toUpperCase();
        String y15 = x2.trim();
        System.out.println(y15);
    }

```

3.2 Bài thực hành cơ bản

Bài 1:

Viết chương trình:

1. *Xây dựng lớp SinhVien gồm:*

✓ Các thuộc tính:

- String maSV, hoTen, diaChi;
- double d1, d2, d3;

✓ Các phương thức:

- Public void nhapTTTSV(), public void inTTTSV(),
- public double tinhDiemTrungBinh

1. *Xây dựng lớp DanhSachSinhVien gồm:*

✓ Các thuộc tính riêng:

- int n; // số sinh viên
- SinhVien dsSinhVien[]; // lưu trữ thông tin của các sinh viên

✓ Các toán tử tạo lập: DanhSachSinhVien(int ssv)

- DanhSachSinhVien(SinhVien mSinhVien[])

✓ Các phương thức:

- Public void nhapDanhSachSinhVien();
- Public void inDanhSachSinhVien();

- `Public void inDanhSachSinhVien(String dc)//in danh sách sinh viên có địa chỉ là dc`

Bài 2:

Xây dựng lớp `MaTran` thực hiện thao tác sau:

1. Nhập ma trận.
2. Hiển thị ma trận.
3. Tìm hàng, cột hoặc đường chéo có tổng các phần tử lớn nhất.
4. Tìm ma trận chuyển vị của A
5. Tìm định thức của A
6. Tìm ma trận nghịch đảo của A

Xây dựng phương thức `main()` sử dụng các phương thức trên.

Bài 3:

Viết chương trình quản lý khách hàng, mỗi khách hàng cần quản lý các thông tin : Họ tên, địa chỉ, điện thoại, mail. Số khách hàng tự nhập từ bàn phím, tìm khách hàng theo tên, địa chỉ.

3.3 Bài thực hành nâng cao

Bài 1:

Để quản lý các hộ dân trong một khu phố, người ta quản lý các thông tin như sau:

- Với mỗi hộ dân, có các thuộc tính:
 - + Số thành viên trong hộ (số người)
 - + Số nhà của hộ dân đó (Số nhà được gán cho mỗi hộ dân)
 - + Thông tin về mỗi cá nhân trong hộ gia đình.
- Với mỗi cá nhân, người ta quản lý các thông tin như: họ và tên, tuổi, năm sinh, nghề nghiệp.

1. Hãy xây dựng các lớp: `NhanSu` để quản lý thông tin về mỗi cá nhân; lớp `HoDan` để quản lý thông tin về các hộ gia đình. Viết các phương thức để nhập, hiển thị thông tin cho mỗi cá nhân.

2. Cài đặt chương trình thực hiện các công việc sau:

- + Nhập vào một dãy gồm n hộ dân (n - nhập từ bàn phím).
- + Hiển thị ra màn hình thông tin về các hộ trong khu phố.

BÀI TẬP THỰC HÀNH 9: XỬ LÝ CHUỖI

1. Mục tiêu:

Về kiến thức:

- Sinh viên viết chương trình giải các bài toán về mảng và chuỗi một cách thành thạo. Khai thác được các thuộc tính và các phương thức của các lớp cơ bản trong gói java.lang

Về kỹ năng:

- Sinh viên phải biết sử dụng chuỗi trong các bài tập.

Về thái độ:

- Tự giác chuẩn bị các câu hỏi và bài tập.
- Thực hiện các bài tập trên máy tính.

2. Yêu cầu

- Sinh viên chuẩn bị trước các bài tập thực hành,
- Thực hành các bài tập trong chương.

3. Nội dung thực hành

3.1 Bài thực hành mẫu

Bài 1: Khởi tạo một chuỗi trong java

```
public class StringInitialization {  
    public static void main(String a[]){  
        String objStr_01 = "This is a string object";  
        String objStr_02 = new String("This is also string object");  
        char[] arrChar = {'V','N','L','T','V','E','S'};  
        String objStr_03 = new String(arrChar);  
        String objStr_04 = objStr_03 + " This is another String object";  
        System.out.println("objStr_01: " + objStr_01);  
        System.out.println("objStr_02: " + objStr_02);  
        System.out.println("objStr_03: " + objStr_03);  
        System.out.println("objStr_04: " + objStr_04);  
    }  
}
```

Bài 2: So sánh chuỗi trong java

```

package javaandroidvn;
public class JavaAndroidVn {
    public static void main(String[] args) {
        String str1 = "Android.Vn Android.Vn";
        String str2 = "android.vn android.vn";
        System.out.println("So sánh phân biệt chữ hoa chữ thường: "+str1.equals(str2));
        System.out.println("So sánh không phân biệt hoa thường: "+str1.equalsIgnoreCase(str2));

        // So sánh thứ tự a, b, c ....
        String str3 = "abc";
        String str4 = "bcde";
        System.out.println("So sánh 2 chuỗi thường!");
        System.out.println("'+str3.compareTo(str4)); //str3 < str4 , so sánh trả về -1
        System.out.println("'+str4.compareTo(str3)); // str4 > str3, so sánh trả về 1
        System.out.println("'+str3.compareTo(str3)); // str3 = str3, so sánh trả về 0

        System.out.println("So sánh 2 chuỗi hoa và thường");
        String str5 = "ANDroid.Vn";
        String str6 = "android.vn";
        System.out.println("'+str5.compareToIgnoreCase(str6)); //KHông phân biệt hoa và thường
        System.out.println("'+str5.compareTo(str6)); // Phân biệt chữ hoa và chữ thường!

        System.out.println("Chuỗi này có là tập con của chuỗi kia không?");
        String str7 = "android.vn";
        String str8 = "vn";
        //Quá trình so sánh có phân biệt chữ hoa và chữ thường!
        System.out.println("Chuỗi vn nằm ở vị trí thứ "+str7.indexOf(str8)+" của chuỗi android.vn"
    );
        //Khi so sánh không tìm thấy thì sẽ trả về -1
        str8 = "TN";
        System.out.println("Chuỗi TN nằm ở vị trí thứ "+str7.indexOf(str8)+" của chuỗi android.vn
    ");
        System.out.println("Chuỗi này có bắt đầu hay kết thúc bằng chuỗi kia không?");
        str7 = "android.vn";
        String str9 = "and";
        String str10 = "roid.vn";
        System.out.println("str7.startsWith(str9) = "+str7.startsWith(str9));
        System.out.println("str7.endsWith(str10) = "+str7.endsWith(str10));
    }
}

```

3.2 Bài thực hành cơ bản

Bài 1:

Viết chương trình

1. Xây dựng lớp Xau có thuộc tính dữ liệu là *private String st* và các phương thức:

- ✓ Nhập dữ liệu : `public void nhapXau()`
- ✓ In dữ liệu `public void inXau()`
- ✓ Thay thế trong xâu hiện thời các xâu con `st1` bằng xâu con `st2`
 - `public void thayThe(String st1,String st2)`
- ✓ Sắp xếp một mảng xâu theo thứ tự giảm dần của từ điển như sau :
 - `public static sapXepXauGiam(String mangXau[])`

2. Xây dựng lớp `ChuongTrinh` nhập vào `n` xâu từ bàn phím thực hiện sắp xếp rồi in kết quả ra màn hình

Bài 2:

Để xử lý các văn bản, người ta xây dựng lớp `VanBan` có thuộc tính riêng là một xâu ký tự. Hãy:

1. Xây dựng lớp `VanBan` có:

- + Các toán tử tạo lập : `VanBan()`, `VanBan(String st)`.
- + Phương thức đếm số từ của xâu thuộc tính trong lớp hiện tại.
- + Phương thức đếm số từ kết thúc bởi ký tự 'G' (không phân biệt chữ hoa/thường).

2. Chuẩn hoá xâu thuộc tính trong lớp hiện tại theo các tiêu chí sau:

- + Ở đầu và cuối của xâu không có ký tự trống
- + Ở giữa xâu, tại vị trí bất kỳ không tồn tại hai ký tự trống đứng liền nhau.

3.3 Bài thực hành nâng cao

Bài 1:

1. Viết chương trình định nghĩa lớp **XauKyTu** với các phương thức thực hiện các công việc sau:

- + các toán tử tạo lập
- + Nhập dữ liệu cho xâu hiện thời

- + Đếm số từ của xâu
 - + Chuẩn hoá xâu theo các tiêu chí sau: Ở đầu và cuối xâu không còn ký tự trống, Ở giữa xâu không tồn tại 2 ký tự trống đứng liền nhau).
 - + Đếm số từ của xâu, số từ kết thúc bởi ký tự ‘G’ (không phân biệt hoa/thường)
 - + Kiểm tra tính đối xứng của xâu
2. Cài đặt chương trình thực hiện nhập vào một xâu ký tự bất kỳ từ bàn phím, sau đó cho biết số từ của xâu vừa nhập; chuẩn hoá xâu rồi hiển thị kết quả ra màn hình, cho biết số từ kết thúc bởi ký tự ‘G’ (không phân biệt hoa/thường) của xâu vừa nhập; In ra xâu đảo ngược của xâu hiện tại (nếu xâu không đối xứng).

BÀI TẬP THỰC HÀNH 10: BÀI TẬP TỔNG HỢP

1. Mục tiêu:

Về kiến thức:

- Sinh viên viết chương trình giải các bài toán về mảng và chuỗi một cách thành thạo. Khai thác được các thuộc tính và các phương thức của các lớp cơ bản trong gói java.lang

Về kỹ năng:

- Sinh viên phải biết sử dụng mảng, chuỗi và các thư viện java hỗ trợ trong các bài tập.

Về thái độ:

- Tự giác chuẩn bị các câu hỏi và bài tập.
- Thực hiện các bài tập trên máy tính.

2. Yêu cầu

- Sinh viên chuẩn bị trước các bài tập thực hành,
- Thực hành các bài tập trong chương.

3. Nội dung thực hành

3.1 Bài thực hành mẫu

Bài 1: Lọc các số có trong chuỗi.

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class JavaExtractNumbersInString {
    public static void main(String[] args) {
        String str = "This string has numbers 35 and 21 in it.";
        System.out.println(str);
        // Find 1 or more numbers.
        Pattern p = Pattern.compile("\\d+");
        Matcher m = p.matcher(str);
        while (m.find()) {
            String found = str.substring(m.start(), m.end());
            Integer i = Integer.valueOf(found);
            System.out.println("Number found is: " + i);
        }
    }
}
```

```

        System.out.println("\n \t -- VNLIVES.NET --");
    }
}

Bài 2: Đổi ngày trong java
import java.util.Date;
import java.text.SimpleDateFormat;
import java.text.ParseException;
public class ConvertStringToDate {
    public static void main(String[] args) {
        SimpleDateFormat formatter = new SimpleDateFormat("ddMMyyy");
        String dateString = "12042014";
        try {
            Date date = formatter.parse(dateString);
            System.out.println("Date string: " + dateString);
            System.out.println("String convert to date: " + date);
            System.out.println("Date with formatter: " + formatter.format(date));
        } catch (ParseException e) {
            e.printStackTrace();
        }
    }
}

```

3.2 Bài thực hành cơ bản

Bài 1:

Xây dựng lớp PhanSo với hai thuộc tính riêng xác định tử số và mẫu số của phân số, đồng thời có các phương thức sau:

- + Các toán tử tạo lập
 - + Các phép toán cộng, trừ, nhân, chia hai phân số
 - + Phép kiểm tra một phân số có phải tối giản hay không
 - + Phép tìm dạng tối giản của phân số
2. Viết một chương trình ứng dụng thực hiện việc nhập vào một dãy các phân số
- + In ra màn hình dạng tối giản của các phân số đó
 - + Sắp xếp các phân số theo thứ tự tăng dần của tử số của mỗi phân số

Bài 2:

Để quản lý các hộ dân trong một khu phố, người ta quản lý các thông tin như sau:

- Với mỗi hộ dân, có các thuộc tính:
 - + Số thành viên trong hộ (số người)
 - + Số nhà của hộ dân đó (Số nhà được gán cho mỗi hộ dân)
 - + Thông tin về mỗi cá nhân trong hộ gia đình.
- Với mỗi cá nhân, người ta quản lý các thông tin như: họ và tên, tuổi, năm sinh, nghề nghiệp.

1. Hãy xây dựng các lớp: NhanSu để quản lý thông tin về mỗi cá nhân; lớp HoDan để quản lý thông tin về các hộ gia đình. Viết các phương thức để nhập, hiển thị thông tin cho mỗi cá nhân.

2. Cài đặt chương trình thực hiện các công việc sau:

- + Nhập vào một dãy gồm n hộ dân (n - nhập từ bàn phím).
- + Hiển thị ra màn hình thông tin về các hộ trong khu phố.

Bài 3:

Một công ty được giao nhiệm vụ quản lý các phương tiện giao thông gồm các loại: ô tô, xe máy, xe tải.

- + Mỗi loại phương tiện giao thông cần quản lý: Số máy, hãng sản xuất, năm sản xuất, giá bán.
- + Các ô tô cần quản lý: số chỗ ngồi, kiểu động cơ
- + Xe máy cần quản lý: công suất
- + Xe tải cần quản lý: trọng tải.

1. Xây dựng các lớp XeTai, XeMay, OTo kế thừa từ lớp PTGT; đồng thời có các hàm để truy nhập, hiển thị và kiểm tra các thuộc tính của các lớp.

2. Xây dựng lớp QLPTGT cài đặt các phương thức thực hiện các chức năng sau:

- + Nhập thông tin đăng ký cho một danh sách gồm m phương tiện
- + Nhập vào số máy của một phương tiện, cho biết phương tiện đó thuộc loại gì? Và hiển thị thông tin quản lý của phương tiện đó.

3.3 Bài thực hành nâng cao

Bài 1:

Cho hai đa thức $P_n(x)$ và $Q_m(x)$. Hãy viết chương trình thực hiện những thao tác sau:

Tạo lập hai đa thức (nhập hệ số cho đa thức từ bàn phím hoặc file)

Tính $P_n(x_0)$ và $Q_m(x_0)$

Tìm đạo hàm cấp $1 \leq n$ của đa thức.

Tìm $P_n(x) + Q_m(x)$

Tìm $P_n(x) - Q_m(x)$

Tìm $P_n(x) / Q_m(x)$ và đa thức dư