Yaswanth ram

CB.EN. U4CSE18243

MODULE DESCRIPTION:

My application is about Mailing system

Classes:

1.people details

It takes the details of preferred person facing the issue

The class people details consists of

1.Name;

2.college id;

3. Roll No;

4.Dept;

5.PhoneNo;

6.Hostel;

2.Student

3.Faculty

4.Warden

5.Complaint

6.Mailing system test

In this case study, the person who is facing the issue will register the complaint. If it was a day scholar, the complaint was mailed by faculty to the mechanic and if it was a hostler, the complaint was mailed by warden and the students can know the date of completion of the specific problem by contacting the warden office or faculty chamber.

Extend the program with the following concepts:

1. Exception (any two inbuilt exceptions, any two user defined exception)
2. Stack data structure (10M)
3. Queue data structure (10M)
4. Singly linked list (10M)

Coding part:

import java.util.\*

class Mailing\_system

{

public abstract class PeopleDetails

{

String Name;

String ClgId;

int RollNo;

String Dept;

private int PhoneNo;

String Hostel;

public Details(String name, int rollno, int phone\_number)

{

Name=name;

RollNo=rollno;

PhoneNo=phone\_number;

}

public void setname(String name)

{

Name=name;

}

public String getname()

{

return Name;

}

public void setclgid(String college\_id)

{

ClgId=college\_id;

}

public String getclgid()

{

return ClgId;

}

public void setrollno(int rollno)

{

RollNo=rollno;

}

public int getrollno()

{

return RollNo;

}

public void setdepartment(String department)

{

Dept=department;

}

public String getdepartment()

{

return Dept;

}

public void sethostelname(String Hostel\_name)

{

Hostel=Hostel\_name;

}

public String gethostelname()

{

return Hostel;

}

public void setphoneno(int phoneno)

{

PhoneNo=phoneno;

}

public int getphoneno()

{

return PhoneNo;

}

public String toString()

{

{

return String.format("%s\n%s\n%d\n%s\n%d\n%s",getname(),getclgid(),getrollno(),getdepartment(),getphoneno(),gethostelname())class NumberFormat\_Demo {

class NumberFormat {

public static void main(String args[])

{

try {

int num = Integer.parseInt("yash");

System.out.println(num);

}

catch (NumberFormatException e) {

System.out.println("Number format exception");

}

}

}

}

}

public class Student extends PeopleDetails

{

public Student(String name,int rollno,int phone\_number,String dept)

{

super(name,rollno,phone\_number);

super.setdepartment(dept);

}

public String toString()

{

return String.format("%s\n%s\n%d\n%s\n%d\n%s",getname(),getclgid(),getrollno(),getdepartment(),getphoneno(),gethostelname());

}

}

public class Faculty extends PeopleDetails

{

public Faculty(String name,int rollno,int phone\_number,String dept)

{

super(name,rollno,phone\_number);

super.setdepartment(dept);

}

public String toString()

{

return String.format("%s\n%s\n%d\n%s\n%d",getname(),getclgid(),getrollno(),getdepartment(),getphoneno());

}

}

public class Warden extends PeopleDetails

{

public Warden(String name,int rollno,int phone\_number,String hostel)

{

super(name,rollno,phone\_number);

super.sethostelname(hostel);

}

public String toString()

{

return String.format("%s\n%s\n%d\n%d\n%s",getname(),getclgid(),getrollno(),getphoneno(),gethostelname());

}

}

public class Complaint extends Student

{

public Complaint(String name,int rollno,int phone\_number,String dept,String complaint\_raised,String location\_of\_issue,String nature\_of\_issue)

{

super(name,rollno,phone\_number,dept);

setcomplaintdetails(complaint\_raised,location\_of\_issue,nature\_of\_issue);

}

public String setcomplaintdetails(String complaint\_raised,String location\_of\_issue,String nature\_of\_issue)

{

String Compt=complaint\_raised;

String Loc=location\_of\_issue;

String Nature=nature\_of\_issue;

}

public String getcomplaintintensity()

{

return Nature;

}

public String toString()

{

return String.format("%s\n%s\n%s\n%s\n%s\n%s","Student Details:-",super.toString(),"Complain\_Details:-",Compt,Loc,getcomplaintintensity());

}

}

class File\_notFound {

public static void main(String args[])

{

try {

File file = new File("E:// file.txt");

FileReader fr = new FileReader(file);

}

catch (FileNotFoundException e) {

System.out.println("File does not exist");

}

}

}

public class Mailing\_system\_test

{

public static void main(String args[])

{

Student student = new Student("Ramesh",18123,8080901011,"Computer Science");

Faculty faculty = new Faculty("Rajeev",15111,8080909088,"Computer Science");

Warden warden = new Warden("Shanmukh",14545,8080909188,"Valmiki Bhavanam");

Complaint complaint = new Complaint("Ramesh",18123,8080901011,"Computer Science","Water Dispenser not working","Valmiki Bhavanam","High intensity")

PeopleDetails peopledetails[]=new PeopleDetails[4];

peopledetails[0] = student;

peopledetails[1] = faculty;

peopledetails[2] = warden;

peopledetails[3] = complaint;

for (int j = 0; j < peopledetails.length; j++)

{

System.out.printf("Person %d is a %s \n",j, peopledetails[j],getClass().getName());

}

for (PeopleDetails PD : peopledetails)

{

if (PD instanceof Complaint)

{

Complaint peopledetail = (Complaint) PD;

System.out.println("")

System.out.println("")

System.out.println("Complaint raiser's Details and Complaint Details :-")

}

System.out.println(PD);

}

}

}

}

String methods:

String name =”president George Washington “

returns

date. Index(‘p’) 0

date. Index(‘e’) 2

date. Index(‘George’) 10

date. Index(‘e’,3)

string methods – equality

Boolean b= world1.equals(word2);

Returns true if the string word1 is equal to word 2

Boolean b =world1.equalsignorecase(word2);

Returns true if the string word1 matches word 2, case-blind

Int diff = word 1.compareTo(word2);

Returns the “ difference “ word1 – word2

Int diff = word1.compareToignorecase(word2);

Returns the “difference “word1 – word 2,case – blind

String word2 = word1.trim ():

Returns a new string formed from word1 by removing white space at both ends does not affect white space in the middle.

Road map:

Process concept

Thread concept

Thread scheduling

Creating threads

Thread synchronization

Pausing threads

Multi programming , Multi Processing

Threads in java:

Extending Class Thread

Pubic class test driver {

Public static void main (string [] args)

System.out.println(“Main ThreadId:”+

Thread.currentThread().getId());

Processes in the OS:

Two layers for processes

Thread

Independent stream of instructions

Basic unit of CPU utilization