A page of a book

Description automatically generated with low confidence

**package** q1;

**public** **abstract** **class** Doctor {

**static** **int** *ID*;

**static** String *name*;

**static** String *specialization*;

Doctor(**int** ID,String name,String specialization)

{

**this**.*ID*=ID;

**this**.*name*=name;

**this**.*specialization*=specialization;

}

**abstract** **double** computeSalary();

}

**package** q1;

**public** **class** SalariedDoctors **extends** Doctor {

**int** basicPay;

**int** noOfOperations;

**public** SalariedDoctors(**int** ID,String name,String specialization,**int** basicPay,**int** noOfOperations)

{

**super**(ID,name,specialization);

**this**.basicPay=basicPay;

**this**.noOfOperations=noOfOperations;

}

**double** computeSalary() {

**int** sal=basicPay + noOfOperations\*2000;

System.***out***.println("Salaried doctor salary is "+sal);

**return** sal;

}

**public** **void** display()

{

System.***out***.println("Basic pay="+basicPay+"No of operation="+noOfOperations+" Salary="+computeSalary());

}

}

**package** q1;

**public** **class** VisitingDoctor **extends** Doctor {

**int** noOfVisits;

**int** travelFare;

**int** accommodationFare;

VisitingDoctor(**int** ID,String name,String specialization,**int** v,**int** t,**int** a)

{

**super**(ID,name,specialization);

**this**.noOfVisits=v;

**this**.travelFare=t;

**this**.accommodationFare=a;

}

**double** computeSalary() {

**int** vsal=noOfVisits\*(travelFare+accommodationFare);

System.***out***.println("Visiting doctor salary is "+vsal);

**return** vsal;

}

**public** **void** display()

{

System.***out***.println("Travel fare="+travelFare+"No of Visits="+noOfVisits+"Accommodation ="+accommodationFare+" Salary="+computeSalary());

}

}

**package** q1;

**import** java.util.Scanner;

**public** **class** Clinic {

**public** **static** Scanner *s*=**new** Scanner(System.***in***);

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

**int** i;

System.***out***.println("Select Any One: ");

Scanner r=**new** Scanner (System.***in***);

System.***out***.println("1.Salaried Doctor ");

System.***out***.println("2.Visiting Doctor");

**int** ch=r.nextInt();

**switch**(ch){

**case** 1:

System.***out***.println("Enter the number of Salaried Doctor: ");

**int** n=r.nextInt();

SalariedDoctors[] l=**new** SalariedDoctors[n];

**for**(i=0;i<n;i++){

System.***out***.println("Enter the ID,Name,basic,allowance");

l[i]=**new** SalariedDoctors(*s*.nextInt(),*s*.next(),*s*.next(),*s*.nextInt(),*s*.nextInt());

}

**for**(i=0;i<n;i++){

l[i].display();

System.***out***.println(l[i].computeSalary());

}

**break**;

**case** 2:

System.***out***.println("Enter the number of Visiting Doctor: ");

**int** m= r.nextInt();

VisitingDoctor[] h=**new** VisitingDoctor[m];

**for**(i=0;i<m;i++){

System.***out***.println("Enter the ID,Name,hour,wage");

h[i]=**new** VisitingDoctor(*s*.nextInt(),*s*.next(),*s*.next(),*s*.nextInt(),*s*.nextInt());

}

**for**(i=0;i<m;i++){

h[i].display();

System.***out***.println(h[i].computeSalary());

}

**break**;

}

}

}

A close-up of a document

Description automatically generated with medium confidence

**package** q2s;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.Scanner;

**public** **class** Employee {

**public** **static** ArrayList<Employee>*emp*=**new** ArrayList<Employee>();

**public** **static** Scanner *s*=**new** Scanner(System.***in***);

**private** **int** ID,salary;

**private** String gender,name;

Employee(**int** ID,String name,String gender,**int** salary)

{

**this**.ID=ID;

**this**.name=name;

**this**.gender=gender;

**this**.salary=salary;

}

**public** **int** getSalary()

{

**return** salary;

}

**public** String toString()

{

**return** String.*format*("ID=%d Name=%s gender=%s Salary=%d",ID,name,gender,salary);

}

**public** **static** **int** menuOption()

{

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

System.***out***.println("1.Add new Employee");

System.***out***.println("2.Print all");

System.***out***.println("3.Sort By salary");

System.***out***.println("4.Exit");

**return** *s*.nextInt();

}

**public** **static** **void** main(String[] args)

{

**int** ch;

**while**(**true**)

{

ch=*menuOption*();

**switch**(ch)

{

**case** 1:*emp*.add(*addNewEmployee*());

**break**;

**case** 2:

System.***out***.println("All Employee Detail");

**for**(Employee ds:*emp*)

{

System.***out***.println(ds);

}

**break**;

**case** 3:

System.***out***.println("Sort by Salary");

Collections.*sort*(*emp*,**new** SortBySalary());

**for**(Employee a:*emp*)

{

System.***out***.println(a);

}

}

}

}

**private** **static** Employee addNewEmployee() {

System.***out***.println("Enter the ID,Name,Gender,Salary");

**return** **new** Employee(*s*.nextInt(),*s*.next(),*s*.next(),*s*.nextInt());

}

}

**class** SortBySalary **implements** Comparator<Employee>{

@Override

**public** **int** compare(Employee o1, Employee o2) {

**if**(o1.getSalary()==o2.getSalary())

**return** 0;

**else** **if**(o1.getSalary()>o2.getSalary())

**return** 1;

**else**

**return** -1;

}

}

Text

Description automatically generated

**package** q2;

**import** java.util.Scanner;

**public** **class** Conversion {

**private** **int** feet;

**public** **void** setFeet(**int** feet)

{

**try** {

**if**(feet<0)

{

**throw** **new** InvalidInputException(feet);

}

**else**

{

**this**.feet=feet;

}

}

**catch**(InvalidInputException e)

{

System.***out***.println(e);

}

}

**public** **int** Convert()

{

**return** feet\*12;

}

**public** String toString()

{

**return** "Length in inch="+Convert();

}

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

Scanner s=**new** Scanner(System.***in***);

Conversion c=**new** Conversion();

c.setFeet(s.nextInt());

System.***out***.println(c);

}

}

**class** InvalidInputException **extends** Exception{

**int** feet;

InvalidInputException(**int** feet)

{

**this**.feet=feet;

}

**public** String toString()

{

**return** "InvalidInputException";

}

}

A close-up of a document

Description automatically generated with medium confidence

**package** q5;

**public** **abstract** **class** Bank {

**abstract** String getBalance();

}

**class** BankA **extends** Bank{

String getBalance()

{

**return** "$ "+150;

}

}

**class** BankB **extends** Bank{

String getBalance()

{

**return** "$ "+200;

}

}

**class** BankC **extends** Bank{

String getBalance()

{

**return** "$ "+250;

}

}

**package** q5;

**public** **class** Main {

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

Bank b;

b=**new** BankA();

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

System.***out***.println("Balance in Bank A is "+b.getBalance());

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

b=**new** BankB();

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

System.***out***.println("Balance in Bank B is "+b.getBalance());

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

b=**new** BankC();

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

System.***out***.println("Balance in Bank B is "+b.getBalance());

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

}

}

Text, letter

Description automatically generated

**package** q3;

**public** **class** Person {

**protected** **static** String *fname*;

**protected** **static** String *lname*;

}

**class** Votor **extends** Person{

**private** **int** ID,age;

Person p;

Votor(**int** ID,**int** age,String fname,String lname)

{

**this**.ID=ID;

**this**.age=age;

**this**.*fname*=fname;

**this**.*lname*=lname;

**this**.p=p;

}

**public** **void** setAge(**int** age)

{

**try** {

**if**(age>18 && age<110)

{

**this**.age=age;

}

**else**

{

**throw** **new** InvalidAgeException(age);

}

}

**catch**(InvalidAgeException e)

{

System.***out***.println(e);

}

}

**public** **int** getAge()

{

**return** age;

}

**public** String toString()

{

**return** String.*format*("First Name=%s%n Last Name=%s%n ID=%d%n Age=%d",*fname*,*lname*,ID,getAge());

}

}

**class** InvalidAgeException **extends** Exception{

**int** age;

InvalidAgeException(**int** age)

{

**this**.age=age;

}

**public** String toString()

{

**return** "InvalidAge";

}

}

**package** q3;

**public** **class** VotorDemo {

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

Votor v=**new** Votor(90187,19,"Niranjan","sah");

System.***out***.println(v);

}

}

An E-Commerce website sells Mobiles. Every Mobilehas brand, model, price as attributes. Store the data inan ArrayList and sort them by

a) price and

b) brand

**package** q4;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**public** **class** Mobile {

**private** String brand,model;

**private** **int** price;

Mobile(String brand,String model,**int** price)

{

**this**.brand=brand;

**this**.model=model;

**this**.price=price;

}

**public** String getBrand()

{

**return** brand;

}

**public** String getModel()

{

**return** model;

}

**public** String toString()

{

**return** String.*format*("Brand=%s Model=%s Price=%d",brand,model,price);

}

**public** **static** **void** main(String[] args)

{

ArrayList<Mobile>m=**new** ArrayList<Mobile>();

m.add(**new** Mobile("Samsung","S 20 Ultra",80000));

m.add(**new** Mobile("Xiomi","redmi 10",20000));

m.add(**new** Mobile("Oppo","F 20 Ultra",30000));

m.add(**new** Mobile("Vivo","V 20",15000));

Collections.*sort*(m,**new** SortbyBrand());

System.***out***.println("Sort by Brand\n");

**for**(Mobile mo:m)

{

System.***out***.println(mo);

}

Collections.*sort*(m,**new** sortBymodel());

System.***out***.println("\nSort by Model\n");

**for**(Mobile s:m)

{

System.***out***.println(s);

}

}

}

**class** SortbyBrand **implements** Comparator<Mobile>

{

@Override

**public** **int** compare(Mobile o1, Mobile o2) {

**if**(o1.getBrand()==o2.getBrand())

**return** 0;

**else** **if**(o1.getBrand()==o2.getBrand())

**return** 1;

**else**

**return** -1;

}

}

**class** sortBymodel **implements** Comparator<Mobile>

{

@Override

**public** **int** compare(Mobile o1, Mobile o2) {

**if**(o1.getModel()==o2.getModel())

**return** 0;

**else** **if**(o1.getModel()==o2.getModel())

**return** 1;

**else**

**return** -1;

}

}

Draw the class Diagram and develop logic for thefollowing:Consider a static methodcomputeTotal():int. The method reads marks of 3subjects secured by a student, computes and returnstotal. Marks must be within the range of 0-100,otherwise throw InvalidMarksException. (Create auser defined Exception).

**package** q6;

**public** **class** Student {

**private** **static** **int** *mark1*,*mark2*,*mark3*;

**public** **void** setMark(**int** mark1,**int** mark2,**int** mark3)

{

**try** {

**if**((mark1<100) && (mark2<100) &&(mark3<100))

{

**this**.*mark1*=mark1;

**this**.*mark2*=mark2;

**this**.*mark3*=mark3;

}

**else**

{

**throw** **new** InvalidMarkException(mark1,mark2,mark3);

}

}

**catch**(InvalidMarkException e)

{

System.***out***.println(e);

}

}

**public** **static** **int** computeTotal()

{

**return** *mark1*+*mark2*+*mark3*;

}

**public** String toString()

{

**return** String.*format*("Total Makr=%d",*computeTotal*());

}

}

**class** InvalidMarkException **extends** Exception{

**int** mark1,mark2,mark3;

InvalidMarkException(**int** mark1,**int** mark2,**int** mark3)

{

**this**.mark1=mark1;

**this**.mark2=mark2;

**this**.mark3=mark3;

}

**public** String toString()

{

**return** "InvalidMark";

}

}

**package** q6;

**public** **class** MarkDemo {

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

Student s=**new** Student();

s.setMark(70, 80, 90);

System.***out***.println(s);

}

}

The administration of a company decides to record the following data of its Employees – ID and PAN (String). The PAN is a ten-character long alpha-numeric unique identifier. The PAN structure is as follows: AAAPL1234C: The first five characters are letters, followed by four numerals, and the last (tenth) character is a letter. If invalid the setter must throw an Exception. Handle the exception in main() method of EmployeeTest class.

**package** q7;

**public** **class** Employee {

**private** String ID;

**private** String PAN;

**public** **void** setID(String ID)

{

**this**.ID=ID;

}

**public** **void** setPAN(String PAN)

{

**try** {

**if**(PAN.matches("[A-Z]{5}[0-9]{4}[A-Z]{1}+"))

{

**this**.PAN=PAN;

}

**else**

{

**throw** **new** InvalidPAN(PAN);

}

}

**catch**(InvalidPAN e)

{

System.***out***.println(e);

}

}

**public** String getID()

{

**return** ID;

}

**public** String getPAN()

{

**return** PAN;

}

**public** String toString()

{

**return** String.*format*("ID=%s%n PAN=%s%n",getID(),getPAN());

}

}

**class** InvalidPAN **extends** Exception {

String type;

InvalidPAN(String type)

{

**this**.type=type;

}

**public** String toString()

{

**return** "InvalidPAN";

}

}

**package** q7;

**import** java.util.Scanner;

**public** **class** EmployeeTest {

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

Scanner s=**new** Scanner(System.***in***);

Employee e=**new** Employee();

System.***out***.println("Enter the ID");

e.setID(s.next());

System.***out***.println("Enter the PAN");

e.setPAN(s.next());

System.***out***.println(e);

}

}

A page of a book

Description automatically generated with low confidence

**package** q8;

**abstract** **class** Shape {

**abstract** **int** edges();

**abstract** **int** vertices();

}

**class** Rectangle **extends** Shape{

**double** length=5;

**double** breadth=2;

@Override

**int** edges() {

**return** 4;

}

@Override

**int** vertices() {

**return** 4;

}

**double** area()

{

**return** length\*breadth;

}

**public** **void** display()

{

System.***out***.println("Edges:"+edges()+" Vertices:"+vertices()+" Length:"+length+" Breadth:"+breadth+" Area:"+area());

}

}

**class** Cuboid **extends** Rectangle{

**double** height=3;

**double** volume()

{

**return** area()\*height;

}

}

**package** q8;

**public** **class** ShapeDemo {

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

Shape s;

Rectangle r=**new** Rectangle();

r.display();

Cuboid c=**new** Cuboid();

System.***out***.println("Edges"+c.edges()+" Vertices:"+c.vertices()+" Length:"+c.length+" Breadth:"+c.breadth+" Height:"+c.height+" Volume:"+c.volume());;

}

}