Q2 B)

Enum:-

**package** com.code;

**public** **enum** Proprty\_Type

{

***ONERK***,***ONEBHK***,***TWOBHK***,***THREEBHK***,***FOURBHK***;

}

Owner.java:-

**package** com.code;

**import** java.io.Serializable;

**import** java.time.LocalDate;

**public** **class** Owner **implements** Serializable

{

**private** String adharno;

**private** String firstname;

**private** String lastname;

**private** String email;

**private** String address;

**private** String mobileno;

**private** Property\_type type;

**private** LocalDate availableDate;

**private** **int** rent;

**private** **boolean** isrentout =**false**;

**public** Owner(String adharno, String firstname, String lastname, String email, String address, String mobileno,

Property\_type type, LocalDate availableDate, **int** rent, **boolean** isrentout) {

**super**();

**this**.adharno = adharno;

**this**.firstname = firstname;

**this**.lastname = lastname;

**this**.email = email;

**this**.address = address;

**this**.mobileno = mobileno;

**this**.type = type;

**this**.availableDate = availableDate;

**this**.rent = rent;

**this**.isrentout = isrentout;

}

@Override

**public** String toString() {

**return** "Owner [adharno=" + adharno + ", firstname=" + firstname + ", lastname=" + lastname + ", email=" + email

+ ", address=" + address + ", mobileno=" + mobileno + ", type=" + type + ", availableDate=" + availableDate

+ ", rent=" + rent + ", isrentout=" + isrentout + "]";

}

**public** **void** setMobileno(String mobileno) {

**this**.mobileno = mobileno;

}

**public** Owner(String adharno) {

**super**();

**this**.adharno = adharno;

}

**public** String getAdharno() {

**return** adharno;

}

**public** String getFirstname() {

**return** firstname;

}

**public** String getLastname() {

**return** lastname;

}

**public** String getEmail() {

**return** email;

}

**public** String getAddress() {

**return** address;

}

**public** String getMobileno() {

**return** mobileno;

}

**public** Property\_type getType() {

**return** type;

}

**public** LocalDate getAvailableDate() {

**return** availableDate;

}

**public** **int** getRent() {

**return** rent;

}

**public** **boolean** isIsrentout() {

**return** isrentout;

}

}

Custom Exception:-

**package** customException;

@SuppressWarnings("serial")

**public** **class** PropertyException **extends** Exception

{

**public** PropertyException(String msg)

{

**super**(msg);

}

}

Q1>>>>>>>>>

Ioutils:-

**package** Utils;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.ObjectInputStream;

**import** java.io.ObjectOutputStream;

**import** java.util.HashSet;

**import** com.code.Owner;

**public** **class** IoUtils

{

**public** **static** **void** store(String filename,HashSet<Owner> set) **throws** IOException

{

**try**(ObjectOutputStream out=**new** ObjectOutputStream(**new** FileOutputStream(filename)))

{

out.writeObject(set);

}

**catch**( IOException e)

{

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

}

}

**public** **static** HashSet<Owner>restore(String filename) **throws** ClassNotFoundException, IOException

{

**try**(ObjectInputStream in=**new** ObjectInputStream(**new** FileInputStream(filename)))

{

**return** (HashSet<Owner>)in.readObject();

}

**catch**(Exception e)

{

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

**return** **new** HashSet<Owner>();

}

}

}

Q2 A)

Validations:-

**package** Utils;

**import** java.text.ParseException;

**import** java.time.LocalDate;

**import** java.util.HashSet;

**import** CustomException.PropertyException;

**import** com.code.Owner;

**import** com.code.Property\_type;

**public** **class** Validationrules

{

@SuppressWarnings("unlikely-arg-type")

**public** **static** **void** checkduplicates(String adhar,HashSet<Owner>own) **throws** PropertyException

{

**if**(own.contains(adhar))

**throw** **new** PropertyException("Duplicate adhar no!!!");

System.***out***.println("No duplicate ");

}

**public** **static** LocalDate parseDate(String date) **throws** PropertyException

{

LocalDate d1=LocalDate.*parse*(date);

**if**(d1.isBefore(LocalDate.*now*()))

**throw** **new** PropertyException("invalid date ! date shoud be after todays date!!!");

**return** d1;

}

**public** **static** Property\_type validatetype(String type)

{

Property\_type t=Property\_type.*valueOf*(type);

**return** t;

}

**public** **static** Owner validateAll(String adharno, String firstname, String lastname, String email, String address, String mobileno,

String type, String availableDate, **int** rent,**boolean** isavailable,HashSet<Owner>own) **throws** PropertyException

{

*checkduplicates*(adharno, own);

LocalDate date=*parseDate*(availableDate);

Property\_type type1=*validatetype*(type);

**return** **new** Owner(adharno,firstname,lastname, email,address, mobileno,type1, date,rent,isavailable);

}

}

Tester:-

**package** Tester;

**import** java.io.IOException;

**import** java.util.HashSet;

**import** java.util.Scanner;

**import** javax.xml.validation.Validator;

**import** Utils.IoUtils;

**import** Utils.IoUtils.\*;

**import** Utils.Validationrules;

**import** com.code.Owner;

**public** **class** TestOwner {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, IOException

{

**try**(Scanner sc=**new** Scanner(System.***in***))

{

System.***out***.println("Enter file Name:");

String file=sc.next();

HashSet<Owner> owner=IoUtils.*restore*(file);

**boolean** exit=**false**;

**while**(!exit)

{

System.***out***.println("1.Add new Owner "

+ "2. Display All Owners"

+ "3.Remove owner based on adhar no"

+ "4.Update contact no"

+ "5.Sort owner As per Property type and rent"

+ "Change status of owner");

**try**

{

System.***out***.println("Enter your choice:");

**switch**(sc.nextInt())

{

**case** 1:

System.***out***.println("Enter Details to add Owner:");

System.***out***.println("adharno,firstname,lastname, email,address, mobileno,type1, date,rent,isavailable");

Owner own=Validationrules.*validateAll*(sc.next(), sc.next(), sc.next(), sc.next(), sc.next(), sc.next(), sc.next(), sc.next(), sc.nextInt(), sc.nextBoolean(), owner);

owner.add(own);

**break**;

**case** 2:

System.***out***.println("List of owners:");

owner.stream()

.forEach(o->System.***out***.println(o));

**break**;

**case** 3:

// System.out.println("Adhar no:");

// String adhar=sc.next();

// owner.stream()

// .removeIf(p->p.getadharno()==(adhar));

// //.forEach(o1->System.out.println(o1));

**break**;

**case** 4:

// System.out.println("Enter Email no to enter:");

// String email=sc.next();

// // Owner o=new Owner(email);

// if(owner.contains(email))

// {

// System.out.println("Enter contact no to update");

// String contact=sc.next();

// owner. setMobileno(contact);

// }

**break**;

**case** 5:

**break**;

**case** 0:

exit=**true**;

IoUtils.*store*(file, owner);

**break**;

**default**:

System.***out***.println("Invalid choice!!!Try again");

}

}

**catch**(Exception e)

{

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

sc.nextLine();

}

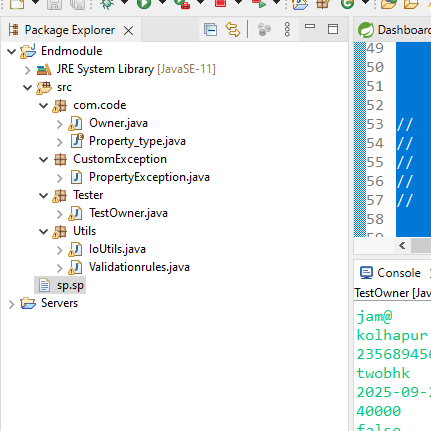
}

}

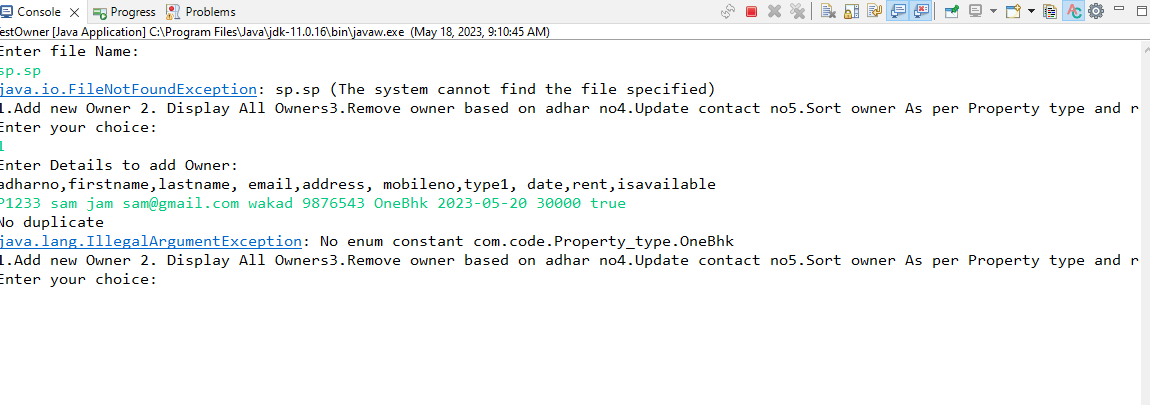
}

}

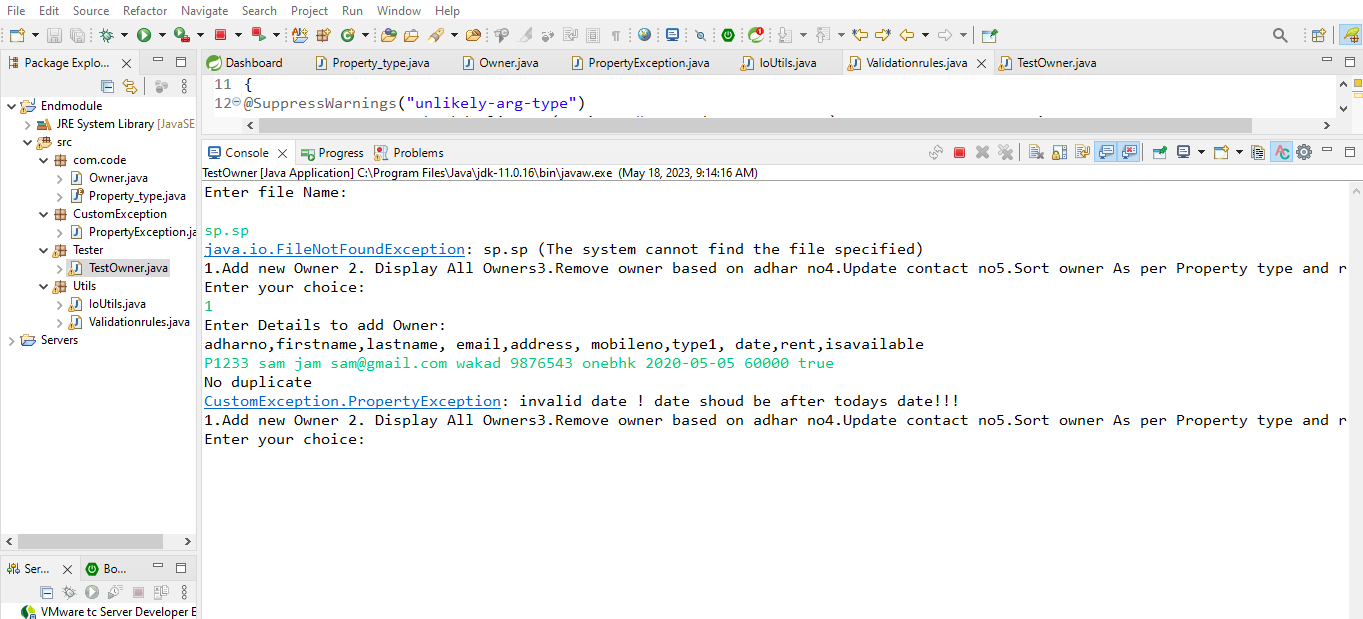
Packages And classes and binary file:-



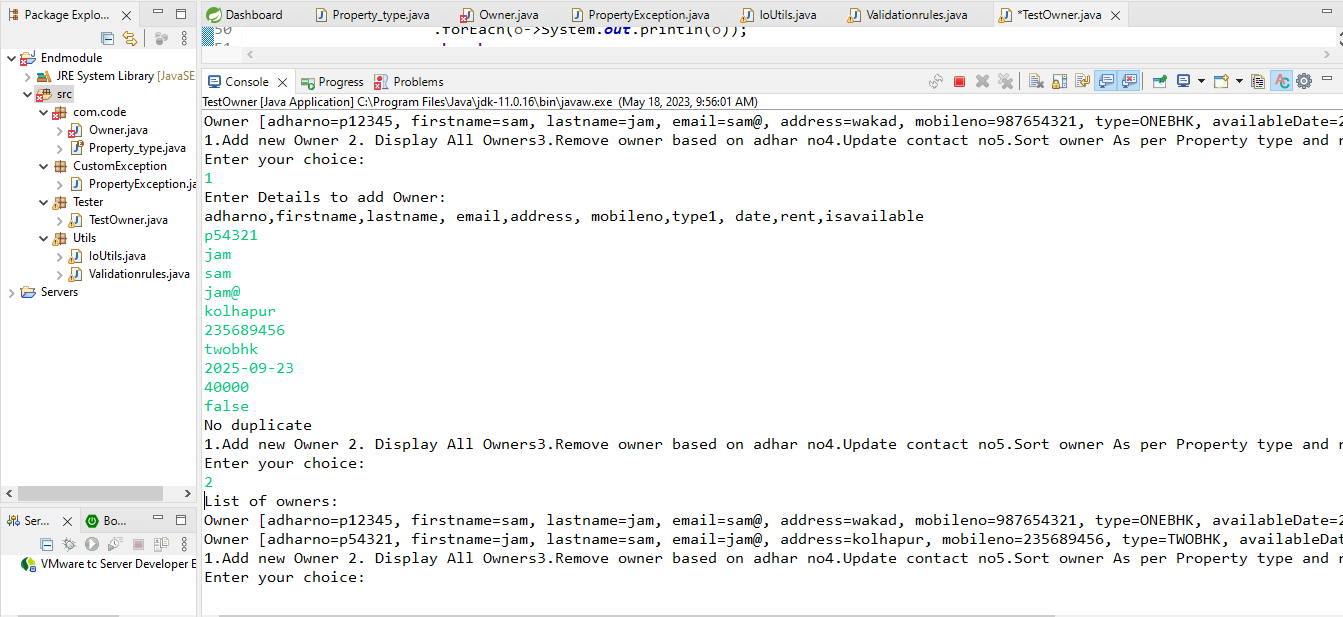
Validating duplicates:-



Validating date:-



Inserting and displaying hashset:-



Binary file:-

