package oop;

import java.util.Scanner;

public class Basla {

public static void main(String[] args) {

int rafsayisi=5;

Kitaplik kitaplik = new Kitaplik(rafsayisi);

Kitap ktp1, ktp2, ktp3;

ktp1 = new Kitap("Pera Palas'ta Gece Yarısı", "Charles King", "978-605-105-148-2",

1, 1);

ktp2 = new Kitap("Ali Emîrî'nin İzinde ", "M. Serhan Tayşi", "978-975-263-956-0", 1, 0);

ktp3 = new Kitap("Bir Türk Ailesinin Öyküsü", "İrfan Orga", "978-975-289-567-6", 2,

2);

kitaplik.kitapEkle(ktp1);

kitaplik.kitapEkle(ktp2);

kitaplik.kitapEkle(ktp3);

kitaplik.gosterRaf(0);

kitaplik.gosterRaf(1);

kitaplik.kitapSil(ktp1);

kitaplik.gosterRaf(1);

kitaplik.gosterRaf(2);

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

int no;

do {

System.*out*.println("Raf numarasi giriniz: ");

no=giris.nextInt();

}while(!kitaplik.rafKontrol(no));

kitaplik.gosterRaf(no);

giris.close();

}

}

package oop;

public class Kitap {

//attributes

private String kitapAd;

private String yazar;

private String ISBN;

private int rafNo;

private int sira;

//constructer yani yapici method olusturuyoruz

//mainin icinde tek tek atamak zorunda kalmayalim diye

//constructer class ismiyle ayni olmali

public Kitap(String kitapAd, String yazar, String iSBN, int rafNo, int sira) {

this.kitapAd=kitapAd;

//soldaki classtaki kitapAd sagdaki senin parameter olarak yollyacagin kitapAd

//karistirmamak icin soldakine this diyerek belirtiyoruz

this.yazar=yazar;

ISBN=iSBN;

//ya da sanirim burada oldugu gibi sagdaki farkli oldugu icin soldakine this demeye gerek kalmadi

this.rafNo=rafNo;

this.sira=sira;

}

//simdi getterlari olusturuyoruz. yukarida yaptigimiz attributes tanimlamalari main icinde o ozelliklere

//erisebilecegimiz anlamina gelmez. okuyabilmek icin mutlaka getter olusturmamiz lazim.

//sadece olusturmak yeterli olmayacaksa o attributelar uzerinde degisiklik de yapmamiz gerekecekse

//setter da olusturuyoruz.

//getterlar return olarak istedigimiz seyi dondururken setterlar void oluyor cunku degisikligi

//yapip geciyoruz.

//set yaparken parametrenin icine koydugun isim sana kalmis

//ister direkt attributeslarla ayni koy ister farkli koy.

//burada mesela setKitapAd'da isim demisiz ama yazarda direkt yazar demisiz.

public String getKitapAd(){

return kitapAd;

}

public void setKitapAd(String isim) {

this.kitapAd = isim;

}

public String getYazar(){

return yazar;

}

public void setYazar(String yazar) {

this.yazar = yazar;

}

public String getISBN(){

return ISBN;

}

public void setISBN(String iSBN) {

ISBN = iSBN;

}

public int getRafNo(){

return rafNo;

}

public void setRafNo(int rafNo) {

this.rafNo = rafNo;

}

public int getSira(){

return sira;

}

public void setSira(int sira) {

this.sira = sira;

}

}

package oop;

public class Kitaplik {

Raf raf[];

//once Raf Classından raf isimli bir obje tanimliyoruz.

//int dizi[]gibi dusunebilirsin

//simdi bu constructor icinde onemli noktalardan biri su, diyelim 5 rafli bir kitaplik var

//once komplesini newle olusturuyosun

//sonra her rafı tek tek

//bu 5 rafı bence yanyana gibi dusun raf[] dedigimizde burada raf dizinin adı ya o yuzden..

//asagidaki satirla sanki o buyuk uzunlamasına olan seyin jelatinini aciyosun gibi

//sonra raf[i] diyerek tek tek kısımlarına gecicez

public Kitaplik(int rafSayi) {

raf = new Raf[rafSayi];

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

raf[i]=new Raf();

}

}

public void kitapEkle(Kitap ktp) {

raf[ktp.getRafNo()].kitapEkle(ktp);

}

public Kitap kitapSil(Kitap ktp2) {

return raf[ktp2.getRafNo()].kitapSil(raf[ktp2.getRafNo()].kitaplar.indexOf(ktp2));

//silmek istedigin kitaba ulasmanin bir suru yolu vardir muhtemelen iste onlardan biri bu

}

public Kitap kitapAra(String kitapAdi) {

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

for(Kitap kitap:raf[i].kitaplar)

if(kitap.getKitapAd().compareTo(kitapAdi)==0)

return kitap;

return null;

}

//for each icin su siteye bak

//:operatorunun solunda istedigimiz turden gecici bi eleman var, sagında ise o elemanın ait oldugu array list

// o gecici elemana bizim dizideki bahsedilen turdeki elemani koyuyor ve iste icerde..

//hangi islem gerceklesitirilecekse onu yapiyor.

// https://www.youtube.com/watch?v=Zbyg\_0RXx6s

public boolean rafKontrol(int rafno) {

if(raf[rafno]==null) {

return false;

}

else {

return true;

}

}

public void gosterRaf(int rafno) {

System.***out***.println(rafno + ". Raftaki Kitaplar...\nKitap Adı\t\t\tYazar\t\tISBN");

for(int i=0;i<raf[rafno].kitaplar.size();i++) {

System.***out***.println(raf[rafno].kitaplar.get(i).getKitapAd()+"\t"+raf[rafno].kitaplar.get(i).getYazar()+"\t"+raf[rafno].kitaplar.get(i).getISBN());

}

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

}

}

package oop;

import java.util.LinkedList;

public class Raf {

//LinkedList bir obje

//14.satirdaki new keywordu de yeni bir obje uretilirken kullaniliyor.

//bu yuzden Kitap classında attributes kisminda new yok ama burada var

LinkedList<Kitap> kitaplar;

//Elemanlari Kitap objesi olan kitaplar isimli bir linked list olusturduk

//simdi constructerını kuruyoruz, class ismi ile aynı olmasi gerektigini unutma

public Raf() {

this.kitaplar=new LinkedList<Kitap>();

}

public void kitapEkle(Kitap ktp) {

this.kitaplar.add(ktp);

}

public Kitap kitapSil(int index) {

Kitap kitap;

kitap=this.kitaplar.remove(index);

return kitap;

}

}