Тема: Об’єктно-орієнтована декомпозиція.

**Мета:** Використання об’єктно-орієнтовного підходу для розробки об’єкта предметної (прикладної) галузі.

**1 ВИМОГИ  
1.1 Розробник**

Інформація про розробника:

- Когутенко Олександр Олексійович;

- КІТ-119Д;

- 11 варіант.

**1.2 Загальне завдання**

* Використовуючи об'єктно-орієнтований аналіз, реалізувати класи для представлення сутностей відповідно [прикладної задачі](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task07/" \l "_4) - domain-об'єктів.
* Забезпечити та продемонструвати коректне введення та відображення кирилиці.
* Продемонструвати можливість управління масивом domain-об'єктів.

1.3 Задача

Магазин:

Запис в каталозі товарів: найменування; одиниця виміру; кількість; ціна одиниці; дата надходження; опис (необмежений набір характеристик у вигляді "властивість, значення").

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Використовуеться наслідування, інтерфейс, поліморфізм.

2.2 Ієрархія та структура класів

Використовую 9 классів: Array, ArrayIterator, HelperClass, InteractiveConsole, Start, SaveArray, Date, Shop, WriteObject.

Array використовую як інтерфейс для класу контейнеру.

ArrayIterator використовую як особисту реалізацію ітератора.

HelperClass допоміжний класс для розрахунків.

InteractiveConsole клас для налагодженого спілкування програми з користувачем.

Start клас який має точку входу у програму.

SaveArray клас контейнер який має все необхідні методи маніпулятори.

Date клас використовуется для збереження дати.

Shop клас приклодної галузі.

WriteObject клас з другою точкою початку де створюється початковий файл для класа контейнера.

2.3 Важливі фрагменти програми

package ua.khpi.oop.kogutenko07;  
  
import java.io.Serializable;  
  
public class Date implements Serializable {  
 private int day;  
 private int month;  
 private int year;  
  
 Date()  
 {  
 this.day = 1;  
 this.year = 1970;  
 this.month=1;  
 }  
  
  
  
 public int getDay() {  
 return day;  
 }  
  
 public void setDay(int day) {  
 this.day = day;  
 }  
  
 public int getMonth() {  
 return month;  
 }  
  
 public void setMonth(int month) {  
 this.month = month;  
 }  
  
 public int getYear() {  
 return year;  
 }  
  
 public void setYear(int year) {  
 this.year = year;  
 }  
  
 public String getDate()  
 {  
 return day + "/" + month + "/" + year;  
 }  
  
 public String GetDate() {  
 String ss = "";  
 Integer day = getDay(), mon = getMonth(), year = getYear();  
 String dayS = day.toString(), monS = mon.toString(), yearS = year.toString();  
 if (day < 10 && mon >= 10) {  
 ss = ss + "0" + dayS + "." + monS + "." + yearS;  
 }  
 else if (mon < 10 && day >= 10) {  
 ss = ss + dayS + "." + "0" + monS + "." + yearS;  
 }  
 else if (mon < 10 && day < 10) {  
 ss = ss + "0" + dayS + "." + "0" + monS + "." + yearS;  
 }  
 else {  
 ss = ss + dayS + "." + monS + "." + yearS;  
 }  
 return ss;  
 }  
  
 public void setDate(int day, int mon, int year)  
 {  
 if(day > 0 && day <= 31)  
 setDay(day);  
 else  
 setDay(1);  
 if(mon > 0 && mon <= 12)  
 setMonth(mon);  
 else  
 setMonth(1);  
 if(year > 1970 && year < 2021)  
 setYear(year);  
 else  
 setYear(2020);  
 }  
  
}

package ua.khpi.oop.kogutenko07;  
  
import java.io.\*;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Scanner;  
  
*/\*\*  
 \* The type Helper class with console.  
 \*/*public class HelperClass extends Object{  
  
 ......  
 public void serialization(String savefile)  
 {  
 File file = new File(savefile);///pathname  
 try {  
 FileOutputStream fos = new FileOutputStream(file);  
 ObjectOutputStream oos = new ObjectOutputStream(fos);  
  
 oos.writeObject(save.size());  
 System.*out*.println("size :" + save.size());  
 for (Shop el : save)  
 {  
 oos.writeObject(el);  
 }  
 }  
 catch(FileNotFoundException e)  
 {  
 e.printStackTrace();  
 }  
 catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void deserializtion(String savefile) {  
 //Array<Shop> container = new SaveArray<>();  
 File file = new File(savefile);///pathname  
 try {  
 FileInputStream fis = new FileInputStream(file);///pathname  
 ObjectInputStream ois = new ObjectInputStream(fis);  
 Integer count = ois.readInt();  
 for(int i = 0; i < count; i++)  
 {  
 save.add((Shop)ois.readObject());  
 }  
  
 }  
 catch(FileNotFoundException e) {e.printStackTrace();}  
 catch (IOException e) {e.printStackTrace(); }  
 catch (ClassNotFoundException e) {e.printStackTrace(); }  
 }  
}

package ua.khpi.oop.kogutenko07;  
  
  
import java.io.Serializable;  
import java.util.Scanner;  
  
public class Shop implements Serializable {  
 private int id;  
 private String name;  
 private String unit;  
 private int count;  
 private Date date = new Date();  
 private Map<String,String> description = new HashMap<String, String>();  
  
 public String getDescription() {  
 String str = "";  
 for(Map.Entry<String, String> entry: description.entrySet())  
 str += entry.getKey() + " - " + entry.getValue() + ", ";  
 return str;  
 }  
  
 public void setDescription(String key, String val) {  
 this.description.put(key,val);  
 }  
  
 public Shop()  
 {  
 id = 0;  
 name = "";  
 unit = "";  
 count = 0;  
 //description = new HashMap<String, String>();  
 }  
  
 public Shop(int id, String name, String unit, int count, Date date, String keyD, String valD) {  
 this.id = id;  
 this.name = name;  
 this.unit = unit;  
 this.count = count;  
 this.date.setDate(date.getDay(),date.getMonth(),date.getYear());  
 setDescription(keyD,valD);  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getUnit() {  
 return unit;  
 }  
  
 public void setUnit(String unit) {  
 this.unit = unit;  
 }  
  
 public int getCount() {  
 return count;  
 }  
  
 public void setCount(int count) {  
 this.count = count;  
 }  
  
 public Date getDate() {  
 return date;  
 }  
  
 public void setDate(Date date) {  
 this.date = date;  
 }  
  
 public String toString()  
 {  
 String info = "id: " + id + " | name: " + name + " | unit: " + unit + " | count: " + count + " | date: " + date.GetDate() + " | description: " + getDescription() + "\n";  
 return info;  
 }  
  
 public void add()  
 {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter info:\n");  
 System.*out*.println("Enter id: ");  
 setId(sc.nextInt());  
 //this.id = sc.nextInt();  
 System.*out*.println("Enter name: ");  
 sc.nextLine();  
 setName(sc.nextLine());  
 //this.name = sc.nextLine();  
 System.*out*.println("Enter unit: ");  
 //sc.nextLine();  
 setUnit(sc.nextLine());  
 //this.unit = sc.nextLine();  
 System.*out*.println("Enter count: ");  
 setCount(sc.nextInt());  
 //this.count = sc.nextInt();  
  
 System.*out*.println("Enter date: \n day - ");  
 int day = sc.nextInt();  
 System.*out*.println("\nmon - ");  
 int mon = sc.nextInt();  
 System.*out*.println("\nyear - ");  
 int year = sc.nextInt();  
 date.setDate(day, mon, year);  
  
 System.*out*.println("\nEnter some description: ");  
 boolean check = true;  
 while (check)  
 {  
 System.*out*.println("\nEnter key: ");  
 sc.nextLine();  
 String key = sc.nextLine();  
 System.*out*.println("\nEnter val: ");  
 //sc.nextLine();  
 String val = sc.nextLine();  
 this.description.put(key,val);  
 System.*out*.println("Do you want to add mor one description? (0 - no, 1 - yes)\n> ");  
 int answer = sc.nextInt();  
 if (answer == 0)  
 {  
 check = false;  
 }  
 }  
 }  
  
}

package ua.khpi.oop.kogutenko07;  
  
import java.io.\*;  
  
public class WriteObject {  
 public static void main (String args[])  
 {  
 Date date = new Date();  
 date.setDate(1,2,2002);  
 Shop shop1 = new Shop(1 , "qwerty", "kg", 12, date, "color", "Blue");  
 Shop shop2 = new Shop(2 , "asdfdg", "l", 13, date, "size", "Big");  
 Shop shop3 = new Shop(3 , "zxvcxv", "m", 123, date, "color", "White");  
 Shop shop4 = new Shop(4 , "fdgffg", "H", 112, date, "size", "Small");  
 Shop shop5 = new Shop(5 , "qwweee", "E", 1, date, "material", "Wood");  
 Shop[] shops = {shop1,shop2,shop3,shop4,shop5};  
 try{  
 FileOutputStream fos = new FileOutputStream("shopsin7.bin");  
 ObjectOutputStream oos = new ObjectOutputStream(fos);  
  
  
 oos.writeInt(shops.length);  
  
 oos.writeObject(shop1);  
 oos.writeObject(shop2);  
 oos.writeObject(shop3);  
 oos.writeObject(shop4);  
 oos.writeObject(shop5);  
  
 /\*for (Shop shop : shops)  
 {  
 oos.writeObject(shop);  
 }\*/  
 oos.close();  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
}

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

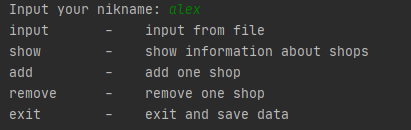


Рисунок 7.1 - початок роботи

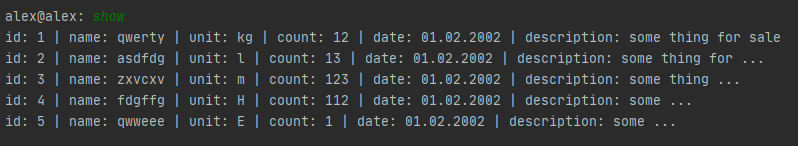
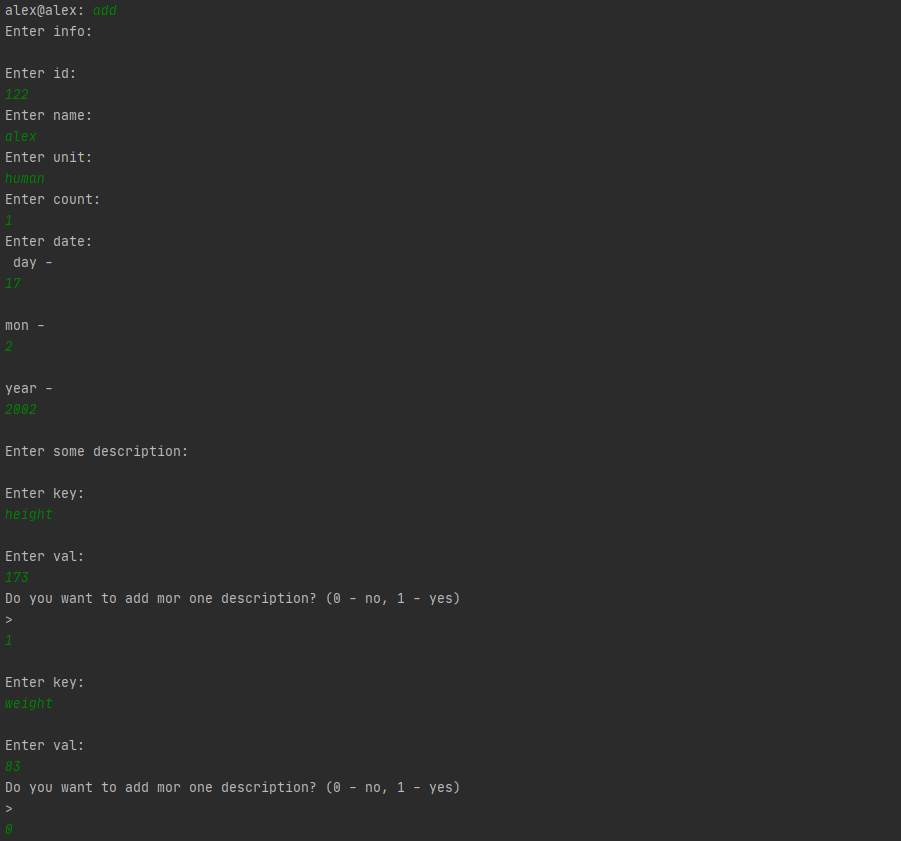


Рисунок 7.2 - додвємо один запис

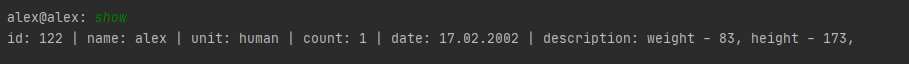


Рисунок 7.2 - вводимо данні

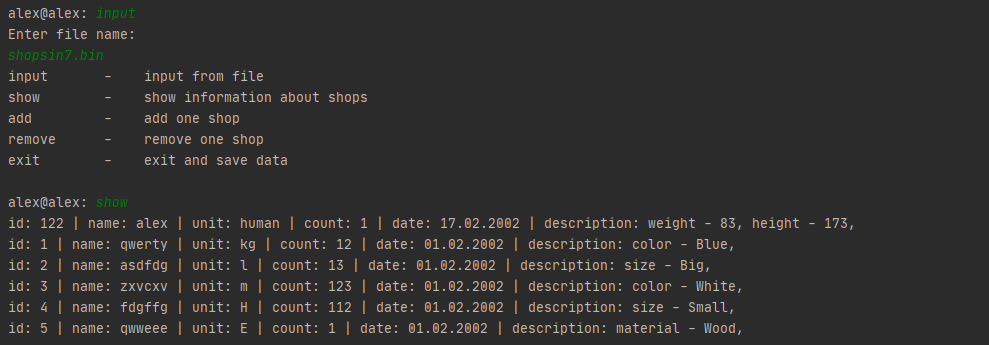


Рисунок 7.3 - запис з файлу

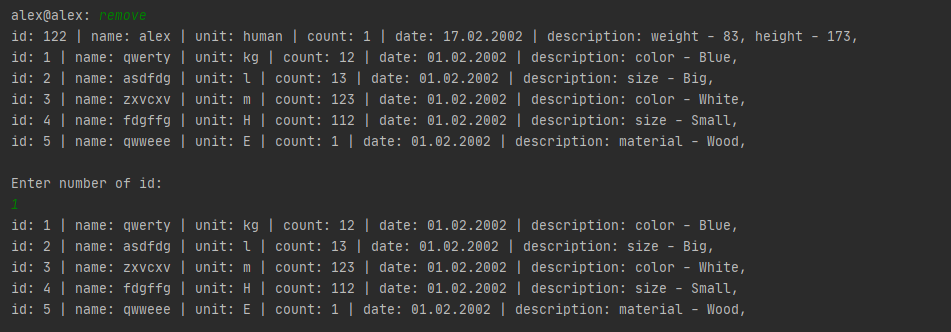


Рисунок 7.4 - видалення 1ого запису

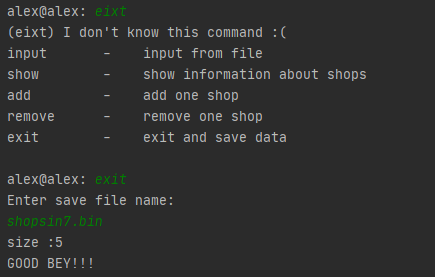


Рисунок 7.5 - збереження у файл після закінчення роботи

**ВИСНОВКИ**

Навчився використовувати об’єктно-орієнтовний підхід для розробки об’єкту предметної (прикладної) галузі.