Національний технічний університет України
«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА №8
З програмування

виконав студент першого курсу
група ІО-91
Нечитайло Олег Андрійович

Тема:Робота з файлами в мові програмування Java.
Мета:Здобуття навичок у створенні власних та використанні існуючих механізмів для роботи з файлами в мові програмування Java.

Варіант 12.

Завдання:Створити засоби для збереження та завантаження колекції з файлу. Передбачити збереження/завантаження колекції як одного об'єкту. Передбачити збереження/завантаження колекції як послідовності об'єктів. Передбачити збереження/завантаження колекції як послідовності об'єктів у вигляді тексту, де кожен рядок відповідає об'єкту, а поля розділені визначеним символом, наприклад, «;» або «:». Для обробки виключних ситуацій необхідно створити власні обробники. Для перевірки необхідно створити клас, що складається з виконавчого методу. Всі дані потрібно вводити з клавіатури. Всі класи повинні бути детально задокументовані з використанням javadoc.

1 CSeaFile

Повне ім'я: `public class CSeaFile`

Наслідується від Object

Клас для роботи з файлами з даними типу ISea

Поля

public File f

Файл для роботи

---

public CSeaConsole cons

об'єкт класу CSeaConsole, через який відбувається читання та запис даних

Конструктори

CSeaFile()

Пустий конструктор

CSeaFile(String filename, boolean create) throws Exception

Параметри	String filename	шлях до файлу без розширення(+".sea"автоматично)
	boolean create	true якщо створити новий файл на місці можливо існуючого, false - дописувати у існуючий.
Виключення	Exception	

Методи

public void writef(ISea o) throws IOException

Запис одного б'єкта до кінця файлу.

Параметри ISea o - об'єкт для запису

Виключення IOException

---

public void writef(CSeaArray o) throws IOException

Запис масиву об'єктів до файлу

Параметри CSeaArray o - об'єкт CSeaArray для запису

Виключення IOException

---

public ISea readf() throws NumberFormatException, IOException, Exception

Прочитати перший елемент файла.

Повертає елемент

Виключення NumberFormatException
IOException
Exception

---

public CSeaArray readfa() throws NumberFormatException, Exception

Повертає увесь зміст файлу у вигляді масиву типу CSeaArray

Виключення NumberFormatException
Exception

---

2 CSeaConsole

Повне ім'я: `public class CSeaConsole`

Наслідується від Object

Клас для організації роботи з **CSeaArray** за допомогою символьних потоків вводу та виводу

Автор Alikont

Поля

InputStream input

Поток вводу

OutputStream out

поток виводу

Конструктори

CSeaConsole()

Встановлює **System.in** та **System.out** як потоки для роботи

CSeaConsole(InputStream inp)

Параметри InputStream inp - потік вводу

CSeaConsole(OutputStream out)

Параметри OutputStream out - потік виводу

CSeaConsole(InputStream inp, OutputStream out)

Параметри InputStream inp - потік вводу
OutputStream out - потік виводу

Методи

public void setComSymb(char s)

Параметри char s встановлюється як символ, що позначає команду

---

public void setDivSymb(char s)

Параметри char s встановлюється як символ, що розділяє параметри

---

public char getComSymb()

Повертає символ для позначення команди

---

public char getSivSymb()

Повертає символ для розділення параметрів

---

public void setFile(CSeaFile file)

Параметри CSeaFile file змінна для роботи з файлом.

---

public CSeaFile getFile()

Повертає файл, з яким ведеться робота

---

public boolean readCommand(CSeaArray arr) throws Exception, NumberFormatException, IOException

читає команду та запускає відповідний метод. Доступні функції та їх синтаксис:
Першим йде символ, що позначає команду(#), потім назва дії, а потім список параметрів через знак розділення(;)
#**madd** - додає елемент на позіцію. Параметри: позиція, індекс класу елемента, список полів елемента.
#**mset** - замінює елемент. Параметри такі самі, як і у #**add**.
#**mget** - виводить елемент на позиції, серед параметрів лише позиція.
#**mdel** - видаляє елемент на позиції, що є єдиним параметром.
#**mall** - виводить весь масив.
#**fopn** - відкриває файл. Параметри: шлях до файлу; чи створювати на тому місці новий.
#**fcls** - закриває файл.
#**fad1** - додає 1 елемент до файлу з вказаної позиції масиву (єдиний параметр).
#**fada** - додає усі елементи масиву до файлу
#**frd1** - читає 1 елемент з файлу
#**frda** - читає усі елементи з файлу і додає їх до колекції
#**exit** - вихід.
Порядок полів при введенні структури: name,ocean,countries,square,salt,volume,temperature.Кожен параметр має закінчуватись на символ розділення(;).

Повертає - **false**, якщо викликано команду #**exit**, інакше **true**.

Параметри CSeaArray arr - масив для роботи

Виключення Exception
NumberFormatException
IOException

---

public ISea readISea() throws Exception, IOException, NumberFormatException

Повертає заповнений елемент типу **ISea** відповідного класу.

Виключення Exception
IOException
NumberFormatException

---

public String readStr() throws Exception

Повертає рядок до першого символу розділення

Виключення Exception

---

public int readInt() throws Exception

читає число від першого символу-цифри до першого символу, який не є цифрою.

Повертає ціле число

Виключення Exception

---

public double readDoub() throws Exception

читає число від першої цифри до першої не цифри, знак розділення цілої і дробової частини - крапка.

Повертає прочитане число

Виключення Exception

---

public void skipToNext() throws Exception

Переходить до найближчого символу розділення

Виключення Exception

---

public void writeISea(ISea obj) throws IOException

виводить об'єкт на заданий потік виводу

Параметри ISea obj - об'єкт для виводу

Виключення IOException

---

3 CSeaArray

Повне ім'я: `public class CSeaArray`

Наслідується від Object

Клас массиву типу ISea

Конструктори

public CSeaArray()

Пустий конструктор, створює масив нульової довжини

---

public CSeaArray(ISea o)

Створює масив довжини 1

Параметри ISea o - елемент масиву

---

public CSeaArray(ArrayList a)

Створює масив з колекції типу ArrayList

Параметри ArrayList a - колекція

Методи

public ISea get(int pos)

Повертає елемент на вказаній позиції

Параметри int pos - позицыя бажаного елемента

---

public int size()

Повертає кількість елементів в масиві

---

public void set(int pos, ISea obj) throws Exception

Змінює значення на заданій позиції, якщо елемемент вже присутній в колекції - не робить нічого

Параметри int pos - позиція
ISea obj - новий об'єкт

Виключення Exception

---

public void add(int pos, ISea obj) throws Exception

Вставити елемент в масив, якщо елемемент вже присутній в колекції - не робить нічого

Параметри int pos - позиція
ISea obj - новий об'єкт

Виключення Exception

---

public void del(int pos) throws ArrayIndexOutOfBoundsException

видалення елементу з вказаної позиції

Параметри int pos - позиція throws ArrayIndexOutOfBoundsException

Виключення ArrayIndexOutOfBoundsException

---

public String toString()

Підготовляє масив до виводу, кожен елемент в новому рядку

---

public int find(ISea obj)

Шукає елемент в масиві, починаючи з початку

Повертає - позицію якщо знайдено, -1 якщо не знайдено

Параметри ISea obj - елемент

---

4 ISea

Повне ім'я: `public abstract interface ISea`

Інтерфейс для класів з данними про море.

Можливо наслідування

Елемент	Наслідувано від

Методи

public String getStrField()

Повертає значення деякого, ключового для даного класу поля з типом String

---

public double getDoubleField()

Повертає значення деякого, ключового для даного класу поля з типом double

---

public void sort1(ISea[] arr)

Метод для сортування.

Параметри ISea[] arr - масив з елементами для сортування

---

public void sort2(ISea[] arr)

Метод для сортування.

Параметри ISea[] arr - масив з елементами для сортування

---

public void setName(String Name)

Параметри String Name

---

public void setOcean(String Ocean)

Параметри String Ocean

---

public void setCountries(String Countries)

Параметри String Countries

---

public void setSalt(double Salt)

Параметри double Salt

---

public void setSquare(double Square)

Параметри double Square

---

public void setVolume(double Volume)

Параметри double Volume

---

public void setTemperature(double Temperature)

Параметри double Temperature

---

public int getClassCode()

---

5 CSea1

Повне ім'я: `public class CSea1`

Наслідується від Object

Реалізує інтерфейс ISea

Клас данних про море №1
{Поля:}
name - Назва моря
ocean - Назва басейну океану
countries - Країни, що мають кордони з морем
square - Площа моря
salt - Соленість води
volume - Об'єм води в морі
temperature -Середня температура води в морі

Конструктори

CSea1(String Name, String Ocean, String Countries, double Square, double Salt, double Volume, double Temperature)

Параметри	String Name	значення поля name
	String Ocean	значення поля ocean
	String Countries	значення поля countries
	double Square	значення поля square
	double Salt	значення поля salt
	double Volume	значення поля volume
	double Temperature	значення поля temperature

Методи

public double getDoubleField()

Перевизначення абстрактного методу інтерфейсу ISea.

Повертає значення поля square

---

public String getStrField()

Перевизначення абстрактного методу інтерфейсу ISea.

Повертає значення поля name

---

public void sort1(ISea[] arr)

Сортування за зростанням за полем name. Використовується CSortByStrComp як компаратор

Параметри ISea[] arr

---

public void sort2(ISea[] arr)

Сортування за зростанням за полем square. Використовується CSortByDoubleComp як компаратор

Параметри ISea[] arr

---

public String toString()

Переводить данні об'єкту в рядок, готовий для виводу.

Повертає рядок з данними.

---

6 CSea2

Повне ім'я: `public class CSea2`

Наслідується від Object

Реалізує інтерфейс ISea

Клас данних про море №1
{Поля:}
name - Назва моря
ocean - Назва басейну океану
countries - Країни, що мають кордони з морем
square - Площа моря
salt - Соленість води
volume - Об'єм води в морі
temperature -Середня температура води в морі

Конструктори

CSea2(String Name, String Ocean, String Countries, double Square, double Salt, double Volume, double Temperature)

Параметри	String Name	значення поля name
	String Ocean	значення поля ocean
	String Countries	значення поля countries
	double Square	значення поля square
	double Salt	значення поля salt
	double Volume	значення поля volume
	double Temperature	значення поля temperature

Методи

public double getDoubleField()

Перевизначення абстрактного методу інтерфейсу ISea.

Повертає значення поля salt

---

public String getStrField()

Перевизначення абстрактного методу інтерфейсу ISea.

Повертає значення поля ocean

---

public void sort1(ISea[] arr)

Сортування за зростанням за полем ocean. Використовується CSortByStrComp як компаратор

Параметри ISea[] arr

---

public void sort2(ISea[] arr)

Сортування за зростанням за полем salt. Використовується CSortByDoubleComp як компаратор

Параметри ISea[] arr

---

public String toString()

Переводить данні об'єкту в рядок, готовий для виводу.

Повертає рядок з данними для виводу.

---

7 CSea3

Повне ім'я: `public class CSea3`

Наслідується від Object

Реалізує інтерфейс ISea

Клас данних про море №1
{Поля:}
name - Назва моря
ocean - Назва басейну океану
countries - Країни, що мають кордони з морем
square - Площа моря
salt - Соленість води
volume - Об'єм води в морі
temperature -Середня температура води в морі

Конструктори

CSea3(String Name, String Ocean, String Countries, double Square, double Salt, double Volume, double Temperature)

Параметри	String Name	значення поля name
	String Ocean	значення поля ocean
	String Countries	значення поля countries
	double Square	значення поля square
	double Salt	значення поля salt
	double Volume	значення поля volume
	double Temperature	значення поля temperature

Методи

public double getDoubleField()

Перевизначення абстрактного методу інтерфейсу ISea.

Повертає значення поля temperature

---

public String getStrField()

Перевизначення абстрактного методу інтерфейсу ISea.

Повертає значення поля countries

---

public void sort1(ISea[] arr)

Сортування за зростанням за полем countries. Використовується CSortByStrComp як компаратор

Параметри ISea[] arr

---

public void sort2(ISea[] arr)

Сортування за зростанням за полем temperature. Використовується CSortByDoubleComp як компаратор

Параметри ISea[] arr

---

public String toString()

Переводить данні об'єкту в рядок, готовий для виводу.

Повертає рядок з данними.

---

8 CSortByStrComp

Повне ім'я: `class CSortByStrComp`

Наслідується від Object

Реалізує інтерфейс Comparator

Клас компаратора, який порівнює 2 об'єкти типу ISea, використовуючи абстрактний метод ISea.getStrField()

Конструктори

CSortByStrComp()

Методи

public int compare(ISea o1, ISea o2)

Параметри ISea o1
ISea o2

---

9 CSortByDoubleComp

Повне ім'я: `class CSortByDoubleComp`

Наслідується від Object

Реалізує інтерфейс Comparator

Клас компаратора, який порівнює 2 об'єкти типу ISea, використовуючи абстрактний метод ISea.getDoubleField()

Конструктори

CSortByDoubleComp()

Методи

public int compare(ISea o1, ISea o2)

Параметри ISea o1
ISea o2

```
//CSeaFile.java
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class CSeaFile {
        public File f;
        public CSeaConsole cons = new CSeaConsole();
        CSeaFile(){
                f=null;
        }
        CSeaFile(String filename, boolean create)throws Exception{
                f=new File(filename+".sea");
                if(create){
                        f.createNewFile();
                        FileOutputStream fos=new FileOutputStream(f.getPath());
                        fos.close();
                }
                else{
                        Exception e;
                        if(!f.canRead())
                                throw e = new Exception("File cannot be read");
                        if(!f.canWrite())
                                throw e = new Exception("File cannot be written");
```

```
                        if(!f.isFile())
                                throw e = new Exception("File is not a file :)");
                }
        }
        public void writef(ISea o) throws IOException{
                cons.out = new FileOutputStream(f.getPath(), true);
                cons.out.write(new String(Integer.toString(o.getClassCode())+';'
                                                                +o.toString()).getBytes());
                cons.out.close();
        }
        public void writef(CSeaArray o) throws IOException{
                cons.out = new FileOutputStream(f.getPath(), true);
                String str;
                for(int i=0;i<o.size();i++){
                        str=String.format("%d;%s",
                                                o.get(i).getClassCode(), o.get(i).toString());
                        cons.out.write(str.getBytes());
                }
                cons.out.close();
        }
        public ISea readf() throws NumberFormatException, IOException, Exception{
                cons.input = new FileInputStream(f.getPath());
                ISea tmp=cons.readISea();
                cons.input.close();
                return tmp;
        }
        public CSeaArray readfa() throws NumberFormatException, Exception{
                cons.input = new FileInputStream(f.getPath());
                CSeaArray arr = new CSeaArray();

                ISea t=null;
                while(cons.input.available()>0){
                        try{t=cons.readISea();}
                        catch(Exception e){
                                if(e.getMessage().equals("End of stream")){
                                        cons.input.close();
                                        return arr;
                                }
                                else
                                        throw e;
                        }
                        arr.add(arr.size(), t);
                }
                cons.input.close();
                return arr;
        }
}
//CSeaConsole.java
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
public class CSeaConsole {
        private int buf;
        InputStream input;
        OutputStream out;
        private CSeaFile file=null;
        private char com='#', div=';';
        CSeaConsole(){
                super();
                input=System.in;
                out=System.out;
        }
        CSeaConsole(InputStream inp){
                super();
                input=inp;
                out=System.out;
        }
        CSeaConsole(OutputStream out){
                super();
                input=System.in;
                this.out=out;
        }
        CSeaConsole(InputStream inp, OutputStream out){
                super();
                input=inp;
                this.out=out;
        }
        public void setComSymb(char s){com=s;}
        public void setDivSymb(char s){div=s;}
        public char getComSymb(){return com;}
```

```
public char getSivSymb(){return div;}
public void setFile(CSeaFile file){
        this.file=file;
}

public CSeaFile getFile(){
        return file;
}
public boolean readCommand(CSeaArray arr)
                        throws Exception, NumberFormatException, IOException{
        int pos;
        buf=input.read();
        while(buf!='#')
                buf=input.read();
        String s=String.format("%c%c%c%c",
                        input.read(), input.read(),input.read(),input.read());
        if(s.equals("madd")){
                pos=readInt();
                skipToNext();
                arr.add(pos, readISea());
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
        if(s.equals("mset")){
                pos=readInt();
                skipToNext();
                arr.set(pos, readISea());
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
        if(s.equals("mdel")){
                pos=readInt();
                skipToNext();
                arr.del(pos);
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
        if(s.equals("mget")){
                pos=readInt();
                skipToNext();
                out.write('\n');
                writeISea(arr.get(pos));
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
        if(s.equals("mall")){
                out.write(arr.toString().getBytes(), 0, arr.toString().length());
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
        if(s.equals("fopn")){
                if(file!=null){
                        Exception e =
                         new Exception("File is already opened. One file at at time.");
                        throw e;
                }
                else{
                        input.read();
                        String fn=readStr();
                        int cr=readInt();
                        file = new CSeaFile(fn, cr!=0);
                }
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
        if(s.equals("fcls")){
                file=null;
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
        if(s.equals("fad1")){
                if(file==null){
                        Exception e = new Exception("file is not opened");
                        throw e;
                }
                input.read();
                file.writef(arr.get(readInt()));
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
```

```
        if(s.equals("fada")){
                if(file==null){
                        Exception e = new Exception("file is not opened");
                        throw e;
                }
                file.writef(arr);
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
        if(s.equals("frd1")){
                if(file==null){
                        Exception e = new Exception("file is not opened");
                        throw e;
                }
                arr.add(arr.size(), file.readf());
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
        if(s.equals("frda")){
                if(file==null){
                        Exception e = new Exception("file is not opened");
                        throw e;
                }
                CSeaArray tmp=file.readfa();
                for(int i=0;i<tmp.size();i++){
                        arr.add(arr.size(),tmp.get(i));
                }
                tmp=null;
                out.write(new String(s+" completed"+'\n').getBytes());
                return true;
        }
        if(s.equals("exit")){
                file=null;
                return false;
        }
        out.write("There is no such command".getBytes());
        return true;
}
public ISea readISea() throws Exception, IOException, NumberFormatException{
        if(input.available()==0){
                Exception e = new Exception("End of stream");
                throw e;
        }
        int cc=readInt();
        ISea obj=null;
        if(cc>3 || cc<1)
        {
                Exception e=
                        Exception(String.format("Wrong class index:%d", cc));
                throw e;
        }
        else{
        switch (cc){
                case 1:{ obj=new CSea1(); break;}
                case 2:{ obj=new CSea2(); break;}
                case 3:{ obj=new CSea3(); break;}
        }
        skipToNext();
        obj.setName(readStr());
        skipToNext();
        obj.setOcean(readStr());
        skipToNext();
        obj.setCountries(readStr());
        skipToNext();
        obj.setSquare(readDoub());
        skipToNext();
        obj.setSalt(readDoub());
        skipToNext();
        obj.setVolume(readDoub());
        skipToNext();
        obj.setTemperature(readDoub());
        skipToNext();
        }
        return obj;
}
public String readStr()throws Exception{
        String s="";
        try{
                if(input.available()==0){
                        Exception e = new Exception("End of stream");
```

```
                        throw e;
                    }
                    buf=input.read();
                }
                catch(IOException e){
                    throw e;
                }
                while(buf!=div){
                    s=String.format("%s%c", s, buf);
                    buf=input.read();
                }
                return s;
        }
        public int readInt() throws Exception{
                if(input.available()==0){
                    Exception e = new Exception("End of stream");
                    throw e;
                }
                buf=input.read();
                String s="0", nbs="1234567890";
                while((nbs.indexOf(buf)==-1)&&(buf!=div)){
                    if(input.available()==0){
                        Exception e = new Exception("End of stream");
                        throw e;
                    }
                    buf=input.read();
                }
                while(nbs.indexOf(buf)!=-1){
                    s=String.format("%s%c", s, buf);
                    if(input.available()==0){
                        Exception e = new Exception("End of stream");
                        throw e;
                    }
                    buf=input.read();
                }
                return Integer.parseInt(s);
        }
        public double readDoub() throws Exception {
                String s="0", nbs="1234567890.";
                if(input.available()==0){
                    Exception e = new Exception("End of stream");
                    throw e;
                }
                buf=input.read();
                while((nbs.indexOf(buf)==-1)&&(buf!=div)){
                    if(input.available()==0){
                        Exception e = new Exception("End of stream");
                        throw e;
                    }
                    buf=input.read();
                }
                while(nbs.indexOf(buf)!=-1){
                    s=String.format("%s%c", s, buf);
                    if(input.available()==0){
                        Exception e = new Exception("End of stream");
                        throw e;
                    }
                    buf=input.read();
                }
                return Double.parseDouble(s);
        }
        public void skipToNext() throws Exception{
                while(buf!=div){
                    if(input.available()==0){
                        Exception e = new Exception("End of stream");
                        throw e;
                    }
                    buf=input.read();
                }
        }
        public void writeISea(ISea obj) throws IOException{
                out.write(obj.toString().getBytes(), 0, obj.toString().length());
        }
}

//CSeaArray.java
import java.util.ArrayList;
public class CSeaArray {
        private ISea[] arr;
        private int n;
```

```
public CSeaArray(){
        super();
        arr=new ISea[0];
        n=0;
}
public CSeaArray(ISea o){
        super();
        arr=new ISea[1];
        arr[0]=o;
        n=1;
}
public CSeaArray(ArrayList<ISea> a){
        super();
        n=a.size();
        arr=new ISea[n];
        for(int i=0;i<n;i++){
                arr[i]=a.get(i);
        }
}
public ISea get(int pos){
        return arr[pos];
}
public int size(){
        return n;
}
public void set(int pos, ISea obj)throws Exception{
        if((0<=pos)&&(pos<=n)){
                if(find(obj)==-1){
                        arr[pos]=obj;
                }
                else{
                        Exception e=new Exception("There is such element in array");
                        throw e;
                }
        }
        else{
                Exception
                e=new Exception(String.format("Position is out of bounds:%d", pos));
                throw e;
        }

}
public void add(int pos, ISea obj)throws Exception{
        if((pos>n)||(pos<0)){
                Exception
                e=new Exception(String.format("Position is out of bounds:%d", pos));
                throw e;
        }
        else{
                if (find(obj)==-1){
                        ISea tmp[]=arr;
                        arr=new ISea[++n];
                        for(int i=0; i<pos; i++){
                                arr[i]=tmp[i];
                        }
                        for(int i=pos;i<n-1; i++){
                                arr[i+1]=tmp[i];
                        }
                        arr[pos]=obj;
                        tmp=null;
                }
                else{
                        Exception e=new Exception("There is such element in array");
                        throw e;
                }
        }
}
public void del(int pos)throws ArrayIndexOutOfBoundsException {
        ArrayIndexOutOfBoundsException
        e=new ArrayIndexOutOfBoundsException(String.format(
                        "Position is out of bounds:%d", pos));
        if(0<=pos && pos<n){
                for(int i=pos;i<n-1;i++){
                        arr[i]=arr[i+1];
                }
                arr[--n]=null;
                return;
        }
        else throw e;
}
```

```
        public String toString(){
                String s="";
                for(int i=0;i<n;i++){
                        s+=arr[i].toString();
                }
                return s;
        }

        public int find(ISea obj){
                for(int i=0;i<n;i++){
                        if(arr[i]==obj) return i;
                }
                return -1;
        }
}
//ISea.java
public interface ISea {
        public String getStrField();
        public double getDoubleField();
        public void sort1(ISea[] arr);
        public void sort2(ISea[] arr);

        public void setName(String Name);
        public void setOcean(String Ocean);
        public void setCountries(String Countries);
        public void setSalt(double Salt);
        public void setSquare(double Square);
        public void setVolume(double Volume);
        public void setTemperature(double Temperature);

        public int getClassCode();
}

//CSea1.java
public class CSea1 implements ISea {
        private String name;
        private String ocean;
        private String countries;
        private double square, salt, volume, temperature;

        public String getName()          {return name;}
        public String getOcean()         {return ocean;}
        public String getCountries() {return countries;}
        public double getSalt()          {return salt;}
        public double getSquare()        {return square;}
        public double getVolume()        {return volume;}
        public double getTemperature(){return temperature;}

        public void setName(String Name)                              {name=Name;}
        public void setOcean(String Ocean)                            {ocean=Ocean;}
        public void setCountries(String Countries)        {countries=Countries;}
        public void setSalt(double Salt)                              {salt=Salt;}
        public void setSquare(double Square)              {square=Square;}
        public void setVolume(double Volume)              {volume=Volume;}
        public void setTemperature(double Temperature){temperature=Temperature;}

        CSea1(){
                super();
        }
        CSea1(String Name, String Ocean,double Square, double Salt){
                this();
                name=Name;
                ocean=Ocean;
                square=Square;
                salt=Salt;
        }
         CSea1(String Name,String Ocean,String Countries,
                                double Square,double Salt,double Volume,double Temperature){
                this(Name, Ocean, Square, Salt);
                countries=Countries;
                volume=Volume;
                temperature=Temperature;
        }

        @Override
        public double getDoubleField() {return square;}
         @Override
        public String getStrField() {return name;}
         @Override
        public void sort1(ISea[] arr) {
```

```
                java.util.Arrays.sort(arr, new CSortByStrComp());
        }
        @Override
        public void sort2(ISea[] arr) {
                java.util.Arrays.sort(arr, new CSortByDoubleComp());
        }
        @Override
        public int getClassCode(){
                return 1;
        }
        public String toString(){
                return String.format("%s;%s;%s;%s;%.2f;%.4f;%.1f;\n",
                                name, ocean, countries, square, salt, volume, temperature);
        }
}
//CSea2.java
public class CSea2 implements ISea {
        private String name;
        private String ocean;
        private String countries;
        private double square, salt, volume, temperature;

        public String getName()          {return name;}
        public String getOcean()         {return ocean;}
        public String getCountries() {return countries;}
        public double getSalt()          {return salt;}
        public double getSquare()        {return square;}
        public double getVolume()        {return volume;}
        public double getTemperature(){return temperature;}

        public void setName(String Name)                              {name=Name;}
        public void setOcean(String Ocean)                            {ocean=Ocean;}
        public void setCountries(String Countries)        {countries=Countries;}
        public void setSalt(double Salt)                              {salt=Salt;}
        public void setSquare(double Square)              {square=Square;}
        public void setVolume(double Volume)              {volume=Volume;}
        public void setTemperature(double Temperature){temperature=Temperature;}

        CSea2(){
                super();
        }
        CSea2(String Name, String Ocean,double Square, double Salt){
                this();
                name=Name;
                ocean=Ocean;
                square=Square;
                salt=Salt;
        }
         CSea2(String Name,String Ocean,String Countries,
                                double Square,double Salt,double Volume,double Temperature){
                this(Name, Ocean, Square, Salt);
                countries=Countries;
                volume=Volume;
                temperature=Temperature;
        }

        @Override
        public double getDoubleField() {return salt;}
        @Override
        public String getStrField() {return ocean;}
         @Override
        public void sort1(ISea[] arr) {
                java.util.Arrays.sort(arr, new CSortByStrComp());
        }
        @Override
        public void sort2(ISea[] arr) {
                java.util.Arrays.sort(arr, new CSortByDoubleComp());
        }
        @Override
        public int getClassCode(){
                return 2;
        }
        public String toString(){
                return String.format("%s;%s;%s;%s;%.2f;%.4f;%.1f;\n",
                                        name, ocean, countries, square, salt, volume, temperature);
        }
}
//CSea3.java
public class CSea3 implements ISea {
        private String name;
```

```
        private String ocean;
        private String countries;
        private double square, salt, volume, temperature;

        public String getName()          {return name;}
        public String getOcean()         {return ocean;}
        public String getCountries() {return countries;}
        public double getSalt()          {return salt;}
        public double getSquare()        {return square;}
        public double getVolume()        {return volume;}
        public double getTemperature(){return temperature;}

        public void setName(String Name)                                {name=Name;}
        public void setOcean(String Ocean)                              {ocean=Ocean;}
        public void setCountries(String Countries)          {countries=Countries;}
        public void setSalt(double Salt)                                {salt=Salt;}
        public void setSquare(double Square)                {square=Square;}
        public void setVolume(double Volume)                {volume=Volume;}
        public void setTemperature(double Temperature){temperature=Temperature;}

        CSea3(){
                super();
        }

        CSea3(String Name, String Ocean,double Square, double Salt){
                this();
                name=Name;
                ocean=Ocean;
                square=Square;
                salt=Salt;
        }
         CSea3(String Name,String Ocean,String Countries,
                                double Square,double Salt,double Volume,double Temperature){
                this(Name, Ocean, Square, Salt);
                countries=Countries;
                volume=Volume;
                temperature=Temperature;
        }
        public double getDoubleField() {return temperature;}
        public String getStrField() {return countries;}
        public void sort1(ISea[] arr) {
                java.util.Arrays.sort(arr, new CSortByStrComp());
        }
        public void sort2(ISea[] arr) {
                java.util.Arrays.sort(arr, new CSortByDoubleComp());
        }
        public int getClassCode(){
                return 3;
        }
        public String toString(){
                return String.format("%s;%s;%s;%s;%.2f;%.4f;%.1f;\n",
                                name, ocean, countries, square, salt, volume, temperature);
        }
}
//CMain.java
import java.util.ArrayList;
public class CMain {
        public static void main(String[] args) {
                ArrayList<ISea> iarr = new ArrayList<ISea>();
                iarr.add(new CSea1("a", "b", "c", 5.3, 12d, 4d, 2d));
                iarr.add(new CSea1("f", "a", "d", 2.3, 6d, 0.01, 2.3));
                iarr.add(new CSea1("c", "d", "f", 3.4, 6.5, 0.5, 8.3));
                CSeaArray myarr = new CSeaArray(iarr);
                System.out.println(myarr);
                CSeaConsole cons=new CSeaConsole();
                boolean ex=true;
                while(ex){
                        try {
                                ex=cons.readCommand(myarr);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
                System.out.print("Bye");
        }
}
```

Висновок: під час виконання цієї роботи я оволодів навичками роботи з символьними потоками введення та виведення, роботи з файловими потоками та покращив навички обробки вхідних даних, роботи з механізмом виключень, організації консольного інтерфейсу користувача.