

Національний технічний університет України  
«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА №5  
З програмування

виконав студент першого курсу  
група ІО-91  
Нечитайло Олег Андрійович

**Тема:**Робота з абстрактними класами та інтерфейсами в мові програмування Java.

**Мета:**Здобуття навичок у створенні та використанні абстрактних класів та інтерфейсів в мові програмування Java.

**Варіант 12.**

**Завдання:**Використовуючи варіант попередньої роботи створити інтерфейс не менше ніж для 3 класів, які складаються не менше ніж з 7 полів та методів доступу до цих полів. В класі необхідно описати, використовуючи стандартні засоби мови програмування, не менше 2 методів, за допомогою яких можна було б сортувати масив об'єктів класу за одним із полів. Для перевірки необхідно створити клас, що складається з виконавчого методу. Всі початкові дані задаються у виконавчому методі. Всі класи повинні бути задокументовані з використанням javadoc.

## 1 ISea

**Повне ім'я:** `public abstract interface ISea`

Інтерфейс для класів з даними про море.

### 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

---

## 2 CSea1

Повне ім'я: `public class CSea1`

Наслідується від `Object`

Реалізує інтерфейс `ISea`

Клас даних про море №1

{Поля:}

`name` - Назва моря

`ocean` - Назва басейну океану

`countries` - Країни, що мають кордони з морем

`square` - Площа моря

`salt` - Солоність 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	<code>String Name</code>	значення поля <code>name</code>
	<code>String Ocean</code>	значення поля <code>ocean</code>
	<code>String Countries</code>	значення поля <code>countries</code>
	<code>double Square</code>	значення поля <code>square</code>
	<code>double Salt</code>	значення поля <code>salt</code>
	<code>double Volume</code>	значення поля <code>volume</code>
	<code>double Temperature</code>	значення поля <code>temperature</code>

---

### 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 даними.

---

### 3 CSea2

Повне ім'я: `public class CSea2`

Наслідується від `Object`

Реалізує інтерфейс `ISea`

Клас даних про море №1

{Поля:}

`name` - Назва моря

`ocean` - Назва басейну океану

`countries` - Країни, що мають кордони з морем

`square` - Площа моря

`salt` - Солоність 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	<code>String Name</code>	значення поля <code>name</code>
	<code>String Ocean</code>	значення поля <code>ocean</code>
	<code>String Countries</code>	значення поля <code>countries</code>
	<code>double Square</code>	значення поля <code>square</code>
	<code>double Salt</code>	значення поля <code>salt</code>
	<code>double Volume</code>	значення поля <code>volume</code>
	<code>double Temperature</code>	значення поля <code>temperature</code>

---

#### 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 даними для виводу.

---

## 4 CSea3

Повне ім'я: `public class CSea3`

Наслідується від `Object`

Реалізує інтерфейс `ISea`

Клас даних про море №1

{Поля:}

`name` - Назва моря

`ocean` - Назва басейну океану

`countries` - Країни, що мають кордони з морем

`square` - Площа моря

`salt` - Солоність 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	<code>String Name</code>	значення поля <code>name</code>
	<code>String Ocean</code>	значення поля <code>ocean</code>
	<code>String Countries</code>	значення поля <code>countries</code>
	<code>double Square</code>	значення поля <code>square</code>
	<code>double Salt</code>	значення поля <code>salt</code>
	<code>double Volume</code>	значення поля <code>volume</code>
	<code>double Temperature</code>	значення поля <code>temperature</code>

---

### 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 даними.

---

## 5 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`

---

## 6 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`

---

```
//CMain.java
public class CMain {
    public static void main(String[] args) {
        ISea iarr[] = { new CSea1("a", "b", "c", 5.3, 12d, 4d, 2d),
                        new CSea1("f", "a", "d", 2.3, 6d, 0.01, 2.3),
                        new CSea1("c", "d", "f", 3.4, 6.5, 0.5, 8.3)};
        System.out.println("+");
        iarr[0].sort1(iarr);
        for(int i=0; i<iarr.length; i++){
            System.out.print(iarr[i]);
        }
        System.out.println("~~~~~+");
        iarr[0].sort2(iarr);
        for(int i=0; i<iarr.length; i++){
            System.out.print(iarr[i]);
        }
        iarr[0] = new CSea2("a", "b", "c", 5.3, 12d, 4d, 2d);
        iarr[1] = new CSea2("f", "a", "d", 2.3, 6d, 0.01, 2.3);
        iarr[2] = new CSea2("c", "d", "f", 3.4, 6.5, 0.5, 8.3);
        System.out.println("~~~+");
        iarr[0].sort1(iarr);
        for(int i=0; i<iarr.length; i++){
```

```

        System.out.print(iarr[i]);
    }
    System.out.println("++++++");
    iarr[0].sort2(iarr);
    for(int i=0; i<iarr.length; i++){
        System.out.print(iarr[i]);
    }
    iarr[0] = new CSea3("a", "b", "c", 5.3, 12d, 4d, 2d);
    iarr[1] = new CSea3("f", "a", "d", 2.3, 6d, 0.01, 2.3);
    iarr[2] = new CSea3("c", "d", "f", 3.4, 6.5, 0.5, 8.3);
    System.out.println("++++");
    iarr[0].sort1(iarr);
    for(int i=0; i<iarr.length; i++){
        System.out.print(iarr[i]);
    }
    System.out.println("++++++");
    iarr[0].sort2(iarr);
    for(int i=0; i<iarr.length; i++){
        System.out.print(iarr[i]);
    }
    System.out.print("\n");
}
}
//ISea.java
public interface ISea {
    public String getStrField();
    public double getDoubleField();
    public void sort1(ISea[] arr);
    public void sort2(ISea[] arr);
}

import java.util.Comparator;
class CSortByStrComp implements Comparator<ISea>{
    public int compare(ISea o1, ISea o2){
        return o1.getStrField().compareTo(o2.getStrField());
    }
}
//CSortByDoubleComp.java
import java.util.Comparator;
class CSortByDoubleComp implements Comparator<ISea>{
    public int compare(ISea o1, ISea o2){
        double r=o1.getDoubleField()- o2.getDoubleField();
        if (r>0) return 1;
        if (r<0) return -1;
        return 0;
    }
}
//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(String Name,String Ocean,String Countries ,
        double Square ,double Salt ,double Volume ,double Temperature){
        name=Name;
        ocean=Ocean;
        square=Square;
        salt=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());
    }
    public String toString(){
        return String.format("%s %s %s %s %.2f %.4f %.1f\n",
            name, ocean, name, 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(String Name,String Ocean,String Countries ,
        double Square ,double Salt ,double Volume ,double Temperature){
        name=Name;
        ocean=Ocean;
        square=Square;
        salt=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());
    }
    public String toString(){
        return String.format("%s_%s_%s_%s_%.2f_%.4f_%.1f\n",
            name, ocean, name, 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(String Name,String Ocean,String Countries ,
        double Square,double Salt ,double Volume,double Temperature){
        name=Name;
        ocean=Ocean;
        square=Square;
        salt=Salt;
        countries=Countries;
        volume=Volume;
        temperature=Temperature;
    }
    @Override
    public double getDoubleField() {return temperature;}
    @Override
    public String getStrField() {return countries;}
    @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());
    }
    public String toString(){
        return String.format("%s_%s_%s_%s_%.2f_%.4f_%.1f\n",
            name, ocean, countries, square, salt, volume, temperature);
    }
}

```

**Висновок:** під час виконання цієї роботи я навчився створювати, реалізовувати та наслідування класів на прикладі інтерфейсів.