Java 프로그래밍

<ArrayList>

```
package com.day07;

import java.util.ArrayList;
import java.util.Scanner;

public class TeacherMain {
	static Scanner sc = new Scanner(System.in);
	ArrayList<Student> arr = new ArrayList<>();


	public static void showMenu() {
		System.out.println("선택하세요>>");
		System.out.println("1. 입력  2. 전체보기/종료");
		System.out.println("선택>>");
	}

	public void inputData() {
		System.out.println("--- 학생 성적 입력 ----");
		System.out.println("이름>>");
		String name = sc.next();
		System.out.println("국어 영어 수학>>");
		int kor = sc.nextInt();
		int eng = sc.nextInt();
		int math = sc.nextInt();
		arr.add(new Student(name, kor, eng, math));
	}

	public void viewData() {
		// for-each
		for (Student s : arr) {
			System.out.println("이름 : " + s.getName());
			System.out.println("국어 : " + s.getKor());
			System.out.println("영어 : " + s.getEng());
			System.out.println("수학 : " + s.getMath());
			System.out.println("총점 : " + s.getTotal());
			System.out.println("평균 : " + s.getAvg());
			System.out.println();
		}
//		for(int i=0; i<arr.size(); i++) {
//		System.out.println("이름 : " + arr.get(i).getName());
//		System.out.println("국어 : " + arr.get(i).getKor());
//		System.out.println("영어 : " + arr.get(i).getEng());
//		System.out.println("수학 : " + arr.get(i).getMath());
//		System.out.println("총점 : " + arr.get(i).getTotal());
//		System.out.println("평균 : " + arr.get(i).getAvg());
//		System.out.println();
//		}
```

```
}

	public static void main(String[] args) {
		TeacherMain t = new TeacherMain();
		while (true) {
			TeacherMain.showMenu();
			int num = sc.nextInt();
			switch (num) {
			case 1:
				t.inputData();
				break;
			case 2:
				t.viewData();
				System.exit(0); // 종료
			default:
				System.out.println("입력오류");

			}// switch
		} // while
	}// main

} // class

package com.day08;

class TV1{
	private int size;
	public TV1(int size) {
		this.size=size;
	}

	//getter
	public int getSize() {
		return size;
	}
}


class ColorTV {
	private int size;
	private int color;

	public ColorTV(int size, int color) {
		this.size = size;
		this.color = color;
	}

	public void print() {
		System.out.println(size +"인치 "+ color+"컬러 입니다.");
	}
```

```
}

class IPTV extends ColorTV {
	private String ip;

	public IPTV(String ip, int size, int color) {
		super(size, color);
		this.ip = ip;
	}

	@Override
	public void print() {
		System.out.print("나의TV는 "+ip+"주소의 ");
		super.print();

	}

}

public class TVMain {
	public static void main(String[] args) {
		ColorTV mytv = new ColorTV(32, 1024);
		mytv.print(); // 32 inch, 1024 Color

		ColorTV iptv = new IPTV("192.1.1.2", 32, 2048);
		iptv.print(); // 나의TV 는 192.1.1.2 주소의 32인치 2048 Color
	}
}
```

<추상 클래스 와 인터페이스 (Abstract Class & Interface)>
구현이 덜 된 클래스 - 추상 클래스

인스턴스를 직접 생성할 수 없는 클래스로, 다른 클래스들의 공통된 특성을 추상화하여 정의하는 클래스인 것이 추상클래스의 특징이다.

인터페이스(interface)는 Java에서 다중 상속을 지원하기 위해 사용되는 개념입니다. 인터페이스는 클래스와 달리 추상적인 개념이다.

```
package com.day08;

public class CalcMain extends Calc{

	public static void main(String[] args) {
		Calc c = new CalcMain();
		//Calc c = new Calc(); // 추상클래스에서는 구현이 불가

	}
```

```
	@Override
	public int add(int a, int b) {
		// TODO Auto-generated method stub
		return 0;
	}

	@Override
	public int sub(int a, int b) {
		// TODO Auto-generated method stub
		return 0;
	}

	@Override
	public double average(int[] a) {
		// TODO Auto-generated method stub
		return 0;
	}

}
```

```
package com.day08;

public abstract class Calc {

	public abstract int add(int a, int b);
	public abstract int sub(int a, int b);
	public abstract double average(int[]a);

}
```

<interface>

```
package com.day08;
	// interface에 선언된 변수는 final(final 생략 가능)
	// 메소드는 추상으로 만들어짐(abstract 생략 가능)
public interface Calc2 {
	public int add(int a, int b);
	public int sub(int a, int b);
	public int double average(int a, int b);
}
```

```
package com.day08;

public class Calc2Main implements Calc2 {

	public static void main(String[] args) {
		Calc2 c2 = new Calc2Main();
		System.out.println(c2.value);
```

```
		//c2.value=50; //final로 선언되어 있어서 오류(수정 불가)

	}

	@Override
	public int add(int a, int b) {
		// TODO Auto-generated method stub
		return 0;
	}

	@Override
	public int sub(int a, int b) {
		// TODO Auto-generated method stub
		return 0;
	}

	@Override
	public double average(int[] a) {
		// TODO Auto-generated method stub
		return 0;
	}

}
```

<Interface 활용 도형 넓이 구하기>

```
package com.day08;


//도형의 넓이와 둘레 구하기
interface ShapeArea {
	double area();
	double circum();
}

	class Rectangle implements ShapeArea {
		private int x;
		private int y;

		public Rectangle(int x, int y) {
			this.x = x;
			this.y = y;
		}

		@Override
		public double area() {
			// TODO Auto-generated method stub
			return x * y;
		}

		@Override
```

```
		public double circum() {
			// TODO Auto-generated method stub
			return (x + y) * 2;
		}

	}

	class SCircle implements ShapeArea {
		private int r;
		public SCircle(int r) {
			this.r=r;
		}

		@Override
		public double area() {
			// TODO Auto-generated method stub
			return r*r*Math.PI;
		}
		@Override
		public double circum() {
			// TODO Auto-generated method stub
			return r*2*Math.PI;
		}
	}


	public class InterfaceTest {

		public static void main(String[] args) {
			Rectangle rec = new Rectangle(5, 7);
			System.out.println("사각형 넓이 : " + rec.area());
			System.out.println("사각형 둘레 : " + rec.circum());
			SCircle cir = new SCircle(7);
			System.out.println("원 넓이 : " + cir.area());
			System.out.println("원 둘레 : " + cir.circum());
		}

	}

package com.day08;

interface PhoneInterface{
	final int TIMEOUT=10000;
	void sendCall();
	void receiveCall();
	default void printLogo() { // Default Method Java 1.8 버전부터 사용 가능
		System.out.println("*** Phone ***");
	}
}
```

```
interface MobileInterface extends PhoneInterface{
	void sendSMS();
	void receiveSMS();
}

interface MP3Interface extends PhoneInterface{
	void play();
	void stop();
}

class PDA{
	public int calculate(int x, int y) {
		return x*y;
	}
}

//class PDA implements PhoneInterface{ // class로 interface를 상속받을때는 implements
//
//}

public class InterfaceExam extends PDA implements MobileInterface, MP3Interface{ //
Add해서 Override 해줘야함. 다중으로도 interface 사용 가능
	public static void main(String[] args) {
	 InterfaceExam ex = new InterfaceExam();
	 System.out.println("3*5=" +ex.calculate(3,5));
	 ex.printLogo();
	 ex.sendCall();
	 ex.receiveCall();
	 ex.sendSMS();
	 ex.receiveSMS();
	 ex.play();
	 ex.stop();
	}

	@Override
	public void sendCall() {
		System.out.println("sendCall");

	}

	@Override
	public void receiveCall() {
		System.out.println("receieveCall");

	}

	@Override
	public void sendSMS() {
		System.out.println("sendSMS");
```

```
	}

	@Override
	public void receiveSMS() {
		System.out.println("receieveSMS");

	}

	@Override
	public void play() {
		System.out.println("play");

	}

	@Override
	public void stop() {
		System.out.println("stop");

	}

}

package com.day08;

import java.util.ArrayList;
import java.util.List;
// p.279
class Shape{
	public void draw() {
		System.out.println("Shape");
	}
}

class Circle extends Shape{
	public void draw() {
		System.out.println("Circle");
	}
}

class Triangle extends Shape{
	public void draw() {
		System.out.println("Triangle");
	}
}

public class ShapeTest {

	public static void main(String[] args) {
		// ArrayList<Shape> list = new ArrayList<>();
		List<Shape> list=new ArrayList<>(); // ArrayList<Shape> list = new
```

```
ArrayList<>();와 같다.
		list.add(new Circle());
		list.add(new Triangle());
		list.add(new Shape());

		// 출력문 => Circle, Triangle, Shape
		for(Shape s : list) {
			s.draw(); // return 값이 없으므로 sysout 사용할 필요 없음.
		}

	}
}

package com.day08;
//p.347
public interface Sort {
	void ascending(int[]arr);
	void descending(int[]arr);
	default void description() {
		System.out.println("숫자 정렬 알고리즘");
	}

}

package com.day08;

public class QuickSort implements Sort {

	@Override
	public void ascending(int[] arr) {
		System.out.println("QuickSort ascending");

	}

	@Override
	public void descending(int[] arr) {
		System.out.println("QuickSort descending");

	}

	@Override
	public void description() {
		// TODO Auto-generated method stub
		Sort.super.description();
		System.out.println("QuickSort 정렬 알고리즘");
	}
}

package com.day08;

public class HeapSort implements Sort {

	@Override
	public void ascending(int[] arr) {
		System.out.println("HeapSort ascending");

	}

	@Override
	public void descending(int[] arr) {
```

```
		System.out.println("HeapSort descending");

	}

	@Override
	public void description() {
		// TODO Auto-generated method stub
		Sort.super.description();
		System.out.println("HeapSort 정렬 알고리즘");
	}
}

package com.day08;

public class BubbleSort implements Sort {

	@Override
	public void ascending(int[] arr) {
		System.out.println("BubbleSort ascending");

	}

	@Override
	public void descending(int[] arr) {
		System.out.println("BubbleSort descending");

	}

	@Override
	public void description() {
		Sort.super.description();
		System.out.println("BubbleSort 정렬 알고리즘");
	}
}

package com.day08;

import java.util.Scanner;

public class SortTest {

	public static void main(String[] args) {
		System.out.println("정렬방식 선택");
		System.out.println("B : BubbleSort");
		System.out.println("H : HeapSort");
		System.out.println("Q : QuickSort");

		Scanner sc = new Scanner(System.in);
		String ch = sc.next();
		Sort sort;

		if (ch.toUpperCase().equals("B")) {
			sort = new BubbleSort();
		} else if (ch.toLowerCase().equals("h")) {
			sort = new HeapSort();
		} else if (ch.equalsIgnoreCase("q")) {
			sort = new QuickSort();
		} else {
			System.out.println("지원되지 않는 기능입니다.");
			return;
		}

//		if (ch.equals("B") || ch.equals("b")) {
//			sort = new BubbleSort();
```

```
//              } else if (ch.equals("H") || ch.equals("h")) {
//                      sort = new HeapSort();
//              } else if (ch.equals("Q") || ch.equals("q")) {
//                      sort = new QuickSort();
//              } else {
//                      System.out.println("지원되지 않는 기능입니다.");
//                      return;
//              }

                int[] arr = new int[10];
                sort.ascending(arr);
                sort.descending(arr);
                sort.description();
        }

}
```