Week 3. Mastering Classes via Object Sorting

– Techniques for Managing Structured Data and Object Behavior 2

황현석 / 25. 04. 12

Review.. Very Important

Input Parsing about Scanner.next()

0	1	2	3	4	5	6	7	8	9	10	11	12
1	2		H	E	L	L	O	\n	1		W	\0

Input Parsing about Scanner.next()

0	1	2	3	4	5	6	7	8	9	10	11	12
1	2		H	E	L	L	O	\n	1		W	\0

return new String("12")

Input Parsing

about Scanner.next()

0	1	2	3	4	5	6	7	8	9	10	11	12
1	2		H	E	L	L	O	\n	1		W	\0

Call Scanner.nextLine()

return new String(“ HELLO”)

FAST I/O about BufferedReader & BufferedWriter

To solve this Problem, we should use more

faster method in Input / Output Object Class

```java
public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(System.out));

        int T = Integer.parseInt(br.readLine());

        while(T-- >0) {
            StringTokenizer st = new StringTokenizer(br.readLine());

            int a = Integer.parseInt(st.nextToken());
            int b = Integer.parseInt(st.nextToken());

            bw.write( str: a + " " + b + '\n');
        }

        bw.flush();
        bw.close();
    }
}
```

Previous homework..

```java
import java.util.*;

public class Main {
    public static char [][] arr;  1 usage

    public static void main(String[] args) {
        int N = new Scanner(System.in).nextInt();
        arr = new char[3*N][10*N];

        String starTower[] = new String[N];

        for (int i=0;i<N;i++)
            starTower[i] = "*".repeat( count: 1 + 2*i);

        for (int i=0;i<N;i++)
            starTower[N-1-i] = " ".repeat(i) + starTower[N -1 -i] + " ".repeat( count: i + 1);

        for (String s : starTower) System.out.println(" ".repeat(N) + s);
        for (String s : starTower) System.out.println(s + s);
    }

}
```

Table of Contents

Types of Classes in Java

Type	Description
Concrete Class	기본적인 클래스
Abstract Class	일부만 구현된 클래스. 일반적인 클래스에서 상속을 통해 실제로 구현해야함.
Interface	기능의 명세만 가진 형태. 다중 구현이 가능하다.
Final Class	완성된 최종 클래스, 더 이상 상속이 불가능함.
Inner Class	클래스 안에 선언된 클래스.
Static Nested Class	inner 클래스가 static으로 선언된 클래스
Anonymous Class	이름없이 선언과 동시에 구현하는 클래스. 1회성
Record Class	데이터 만을 담고 운송하기 위해, 만들어짐. 간결하게 선언하는 불변 클래스

Class

What is Class…?

```java
class Point {  4 usages
    int x, y;  3 usages

    @Override
    public String toString() {
        return "Point [x=" + x + ", y=" + y + "]";
    }
}

public class Main {
    public static void main(String[] args) throws IOException {
        Point point = new Point();

        point.x = 1;
        point.y = 2;

        Point point2 = new Point();

        point2.x = 3;
        point2.y = 4;

        System.out.println(point);
        System.out.println(point2);
    }
}
```

```
/Users/hyeonseok/Library/Java/JavaVirtualMac
Point [x=1, y=2]
Point [x=3, y=4]

Process finished with exit code 0
```

Class

about Constructor

```
class Point {  4 usages
    int x, y;  2 usages

    Point (int x, int y) {  2 usages
        this.x=x;
        this.y=y;
    }

    @Override
    public String toString() {
        return "Point [x=" + x + ", y=" + y + "]";
    }
}

public class Main {
    public static void main(String[] args) throws IOException {
        Point point = new Point( x: 1,  y: 2);
        Point point2 = new Point( x: 3,  y: 4);

        System.out.println(point);
        System.out.println(point2);
    }
}
```

Class

about Constructor

```
class Point {  4 usages
    int x, y;  2 usages

    Point (int x, int y) {  2 usages
        this.x=x;
        this.y=y;
    }
```

```
Point point3 = new Point();
```

```
public class Main {
    public static void main(String[] args) throws IOException {
        Point point = new Point( x: 1,  y: 2);
        Point point2 = new Point( x: 3,  y: 4);

        System.out.println(point);
        System.out.println(point2);
    }
}
```

Class

about Constructor

```
Point () {}  1 usage

Point (int x, int y) {  2 usages
    this.x=x;
    this.y=y;
}
```

```
public static void main(String[] args) throws IOException {
    Point point = new Point( x: 1,  y: 2);
    Point point2 = new Point( x: 3,  y: 4);
    Point point3 = new Point();

    System.out.println(point);
    System.out.println(point2);
}
```

Class

about Constructor

```
class Point {  6 usages
    int x, y, z;  3 usages

    Point () {}  1 usage

    Point (int x) {  no usages
        this.x=x;
    }

    Point (int x, int y) {  2 usages
        this.x=x;
        this.y=y;
    }

    Point (int x, int y, int z) {  no usages
        this.x=x;
        this.y=y;
        this.z=z;
    }
}
```

Class

about Constructor

```
class Point {  6 usages
    int x, y, z;  3 usages

    Point () {}  1 usage

    Point (int x) {  no usages
        this.x=x;
    }

    Point (int x, int y) {  2 usages
        this.x=x;
        this.y=y;
    }

    Point (int x, int y, int z) {  no usages
        this.x=x;
        this.y=y;
        this.z=z;
    }
}
```

```
class Point {  6 usages
    int x, y, z;  1 usage

    Point () {}  1 usage

    Point (int x) {  1 usage
        this.x=x;
    }

    Point (int x, int y) {  3 usages
        this(x);
        this.y=y;
    }

    Point (int x, int y, int z) {  no usages
        this(x, y);
        this.z=z;
    }
}
```

Class

about Extends..

Main.java

Run

```java
class Creature {
    void punch () {
        System.out.println("Creature Punch");
    }
}

class Dog extends Creature {
    //아무것도 적지않음.
}

class Main {
    public static void main(String[] args) {
        Dog dog = new Dog();
        dog.punch();
    }
}
```

Output

```
Creature Punch

=== Code Execution Successful ===
```

Class

about Extends.. and Overriding

Main.java

Run

```java
class Creature {
    void punch () {
        System.out.println("Creature Punch");
    }
}

class Dog extends Creature {
    void punch () {
        System.out.println("Dog punch");
    }

}

class Main {
    public static void main(String[] args) {
        Dog dog = new Dog();
        dog.punch();

        Creature dog_pointer = (Creature) dog;
        dog_pointer.punch();
    }
}
```

Output

```
Dog punch
Dog punch

=== Code Execution Successful ===
```

Class

Overriding…

Main.java

Run

```java
class Creature {
    void punch () {
        System.out.println("Creature Punch");
    }
}

class Dog extends Creature {
    void punch () {
        System.out.println("Dog punch");
    }

    void run () {
        System.out.println("run");
    }
}

class Main {
    public static void main(String[] args) {
        Dog dog = new Dog();
        dog.punch();

        Creature dog_pointer = (Creature) dog;
        dog_pointer.punch();
        dog_pointer.run();
    }
}
```

Output

Clear

```
ERROR!
Main.java:24: error: cannot find symbol
        dog_pointer.run();
                    ^
  symbol:   method run()
  location: variable dog_pointer of type Creature
1 error

=== Code Exited With Errors ===
```

Class

create new anonymous Class in StackFrame

```java
class Point {
    int x, y;

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

    @Override
    public String toString() {
        return x + " " + y;
    }
}

public class Main {
    public static void main(String[] args) throws IOException {
        final int Z = 5;

        System.out.println(
        new Point(1, 2) {
            int z = Z;

            public String toString() {
                return x + " " + z + " " + y;
            }
        });

    }
}
```

Real… Hard… to understand

"Creating an anonymous class works like extending an existing class or implementing an interface."

익명 클래스를 만드는것은 새로운 클래스의 기존 클래스나

인터페이스를 extends 하거나 implements 한것처럼 작동한다.

Interface

What is that..? why use that..?

```java
interface Weapon {  4 usages  2 implementations
    void attack();  1 usage  2 implementations
}

class Sword implements Weapon {  no usages
    public void attack() {  1 usage
        System.out.println("검으로 찌른다!");
    }
}

class Bow implements Weapon {  no usages
    public void attack() {  1 usage
        System.out.println("활을 쏜다!");
    }
}

class Warrior {  no usages
    private Weapon weapon;  2 usages

    // 🗡 외부에서 무기를 주입받음 (Injection)
    public Warrior(Weapon weapon) {  no usages
        this.weapon = weapon;
    }

    public void fight() {  no usages
        weapon.attack(); // 어떤 무기든 쓸 수 있음!
    }
}
```

```java
public static void main(String[] args) throws IOException {
    Warrior warrior = new Warrior(new Bow());
    Warrior warrior2 = new Warrior(new Sword());

    Warrior warrior3 = new Warrior(new Weapon() {
        @Override  1 usage
        public void attack() {
            System.out.println("테스트용 무기 주입.");
        }
    });
```

Interface

Example… Sort… so many..

Main.java Share Run

```java
class MyIntegerList {
    ArrayList<Integer> list = new ArrayList<>();

    //get(index) 하면, index번째 값을 리턴해줌.
    public int get (int index) {
        return list.get(index);
    }

    //add(item) 하면, item을 리스트 맨 뒤에 넣음.
    public void add (int item) {
        list.add(item);
    }

    // 그때 그때, 정렬 기준이 다른데... 모든 상황을 미리 적어 놓을 수가 없다.
    public void sort () {
        //정렬은 하면 돼는데, 기준이 너무 많아...
    }


    public void sort2 () {
        //반대로 정렬...
    }

    public void sort3 () {
        //절대값 순으로 정렬..
    }

    public void sort4 () {
        //이진수로 변환후, 문자열 사전순으로 정렬..
    }
}
```

Interface

Can I get a Parameter of Sorting Standard?

Main.java Share Run

```java
class MyIntegerList {
    ArrayList<Integer> list = new ArrayList<>();

    //get(index) 하면, index번째 값을 리턴해줌.
    public int get (int index) {
        return list.get(index);
    }

    //add(item) 하면, item을 리스트 맨 뒤에 넣음.
    public void add (int item) {
        list.add(item);
```

```java
    public void sort (/* 정렬 기준을 매개변수로 받아볼 순 없나..?*/) {

    }
```

```java

    public void sort2 () {
        //반대로 정렬...
    }

    public void sort3 () {
        //절대값 순으로 정렬..
    }

    public void sort4 () {
        //이진수로 변환후, 문자열 사전순으로 정렬..
    }
}
```

Interface

Obviously Possible.

Main.java

Share Run

```java

interface MyComparator {
    //a 와 b를 비교 했을 때, 누가 더 우선인지 알려주는 함수.
    int compare (int a, int b);
}

class MyIntegerList {
    ArrayList<Integer> list = new ArrayList<>();

    //set(index, item) index번째에, item을 저장함.
    public void set (int index, int item) {
        list.set(index, item);
    }

    public void sort (MyComparator comparator) {
        final int SIZE = list.size();

        //받아온 정렬 기준으로... 버블정렬을 시도함.
        for (int i=0;i<SIZE;i++) {
            for (int j=i+1;j<SIZE;j++) {
                if (comparator.compare(get(i), get(j)) > 0) {
                    int temp = get(i);
                    set(i, get(j));
                    set(j, temp);

                }
            }
        }
    }
```

Interface

So how can I use sort..?

Main.java

Share Run

```java
class SortStandard implements MyComparator {
    @Override
    public int compare(int a, int b) {
        return Integer.compare(Math.abs(a), Math.abs(b));
    }
}

public class Main {
    public static void main(String[] args) {
        MyIntegerList list = new MyIntegerList();

        //랜덤으로 100개의 수 넣기.
        for (int i=0;i<10;i++) list.add(new Random().nextInt(100));

        MyComparator comparator = new SortStandard();

        list.sort(comparator);
        System.out.println(list.list);
    }
}
```

Output

```
[1, 10, 25, 35, 49, 62, 66, 70, 79, 91]

=== Code Execution Successful ===
```

Interface

That's why We should use Anonymous Class

```
class SortStandard implements MyComparator {
    @Override
    public int compare(int a, int b) {
        return Integer.compare(Math.abs(a), Math.abs(b));
    }
}

class Sort2 implements MyComparator {
    @Override
    public int compare(int a, int b) {
        return a-b;
    }
}

class Sort3 implements MyComparator {
    @Override
    public int compare(int a, int b) {
        return b-a;
    }
}
```

정렬 할 때마다 클래스를 만들어야 한다..

파일이 지저분해지고, 관리가 힘들어 진다.

Interface

That's why We should use Anonymous Class

```java
public class Main {
    public static void main(String[] args) {
        MyIntegerList list = new MyIntegerList();

        //랜덤으로 100개의 수 넣기.
        for (int i=0;i<10;i++) list.add(new Random().nextInt(100));

        //익명클래스 작성법을  활용하여, 1회용 클래스를 만듬.
        MyComparator comparator = new MyComparator() {
            @Override
            public int compare(int a, int b) {
                return Integer.compare(Math.abs(a), Math.abs(b));
            }
        };

        list.sort(comparator);
        System.out.println(list);
    }
}
```

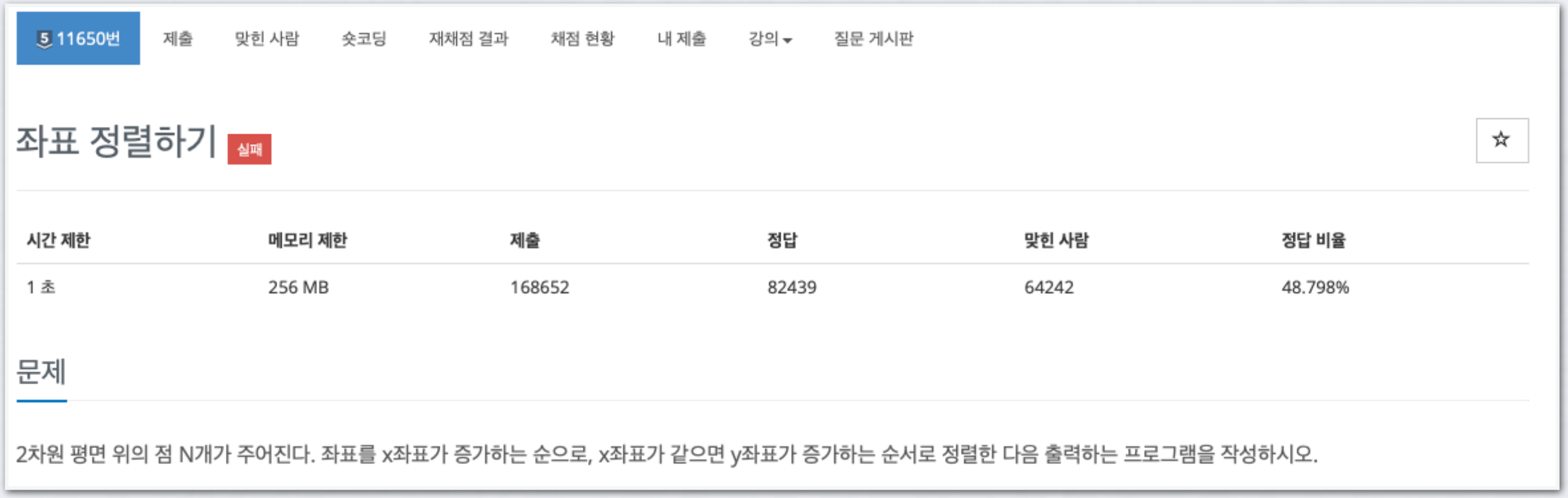
Interface

That's why We should use Anonymous Class

```java
public class Main {
    public static void main(String[] args) {
        MyIntegerList list = new MyIntegerList();

        //랜덤으로 100개의 수 넣기.
        for (int i=0;i<10;i++) list.add(new Random().nextInt(100));

        //익명 클래스를 파라미터를 넘겨주는 공간에서 만들어 버림.
        list.sort(new MyComparator() {
            @Override
            public int compare(int a, int b) {
                return Integer.compare(Math.abs(a), Math.abs(b));
            }
        });

        System.out.println(list);
    }
}
```

ArrayList & Sort

https://www.acmicpc.net/problem/11650



11650번 제출 맞힌 사람 숏코딩 재채점 결과 채점 현황 내 제출 강의 질문 게시판

좌표 정렬하기 실패

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	168652	82439	64242	48.798%

문제

2차원 평면 위의 점 N개가 주어진다. 좌표를 x좌표가 증가하는 순으로, x좌표가 같으면 y좌표가 증가하는 순서로 정렬한 다음 출력하는 프로그램을 작성하시오.

before try this.. we should learn how to Sort.. in Java..

ArrayList & Sort

```java
import java.io.*;
import java.util.*;

public class Main {
    public static void main(String[] args) throws IOException {
        ArrayList<Integer> list = new ArrayList<>();
        for (int i=0;i<10;i++) {
            list.add(new Random().nextInt(100));
        }

        list.sort(new Comparator<Integer>() {
            @Override
            public int compare(Integer o1, Integer o2) {
                return o1 - o2;
            }
        });

        System.out.println(list);
    }
}
```

```java
public void sort(
    java.util.Comparator<? super E> c
)
```

ArrayList & Sort

```java
public class Main {
    public static void main(String[] args) throws IOException {
        ArrayList<Integer> list = new ArrayList<>();
        for (int i=0;i<10;i++) {
            list.add(new Random().nextInt(100));
        }

        Comparator<Integer> cmp = new Comparator<Integer>() {
            @Override
            public int compare(Integer o1, Integer o2) {
                return o2-o1;
            }
        };

        list.sort(cmp);

        System.out.println(list);
    }
}
```

ArrayList & Sort

```java
import java.io.*;
import java.util.*;

public class Main {
    public static void main(String[] args) throws IOException {
        ArrayList<Integer> list = new ArrayList<>();
        for (int i=0;i<10;i++) {
            list.add(new Random().nextInt(100));
        }

        list.sort(new Comparator<Integer>() {
            @Override
            public int compare(Integer o1, Integer o2) {
                return o1 - o2;
            }
        });

        System.out.println(list);
    }
}
```

```java
public void sort(
    java.util.Comparator<? super E> c
)
```

ArrayList & Sort

```java
class ImplementedClass implements Comparator<Integer> {  2 usages
    @Override
    public int compare(Integer o1, Integer o2) {
        return o1 - o2;
    }
}

public class Main {
    public static void main(String[] args) throws IOException {
        ArrayList<Integer> list = new ArrayList<>();
        for (int i=0;i<10;i++) {
            list.add(new Random().nextInt( bound: 100));
        }

        ImplementedClass cmp = new ImplementedClass();
        list.sort(cmp);

        System.out.println(list);
    }
}
```

```java
    }

    list.sort(new ImplementedClass());
    System.out.println(list);
}
```

```java
public void sort(
    java.util.Comparator<? super E> c
)
```

ArrayList & Sort

https://www.acmicpc.net/problem/2750

2750번 | 제출 | 맞힌 사람 | 숏코딩 | 재채점 결과 | 채점 현황 | 내 제출 | 질문 게시판

수 정렬하기

☆

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	235995	135893	93087	58.282%

문제

N개의 수가 주어졌을 때, 이를 오름차순으로 정렬하는 프로그램을 작성하시오.

다음을, BufferedReader, BufferedWriter, ArrayList
ArrayList.sort() 를 활용하여 풀어보자.

ArrayList & Sort

https://www.acmicpc.net/problem/11650

11650번 | 제출 | 맞힌 사람 | 숏코딩 | 재채점 결과 | 채점 현황 | 내 제출 | 강의 | 질문 게시판

좌표 정렬하기 실패

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	168652	82439	64242	48.798%

문제

2차원 평면 위의 점 N개가 주어진다. 좌표를 x좌표가 증가하는 순으로, x좌표가 같으면 y좌표가 증가하는 순서로 정렬한 다음 출력하는 프로그램을 작성하시오.

다음을, Class, BufferedReader, BufferedWrtier
ArrayList, ArrayList.sort(),
anonymous Class implements Comparator interface,
를 사용하여 풀자.

End of this week..

오늘 배운 Weapon의 attack()동작을 바로바로 재정의해서
기존 다른 매서드의 attack()호출에 대해 재정의한 함수를
실행유도 시키는 것을 의존성 주입이라고 한다.

1. 결합도를 낮춰줌

- 객체 간 의존이 약해짐
- 유지보수와 확장성이 엄청 쉬워짐

2. 테스트하기 쉬움

- 테스트할 때 진짜 Engine 대신 **가짜(Mock) Engine**을 넣을 수 있음
- 유닛 테스트가 훨씬 쉬워짐

3. 코드 재사용성 증가

- 다양한 구현체를 바꿔서 쓸 수 있음
- 코드가 유연하고 단단해짐

4. Spring 같은 프레임워크의 핵심 원리

- DI 없이는 스프링을 제대로 이해할 수 없음
- 실제 대규모 애플리케이션에서는 **생명주기, 관리, 연결**을 전부 DI가 담당

We’ve completed Week 3