

Week 2. A Study on Complex Input Handling and Object- Oriented Concepts

- Techniques for Managing Structured Data and Object Behavior

Review.. Very Important

Data Type - What is StringBuilder()?

StringBuilder()는 효율적인, String 구축을 위해 사용하는 클래스이다.

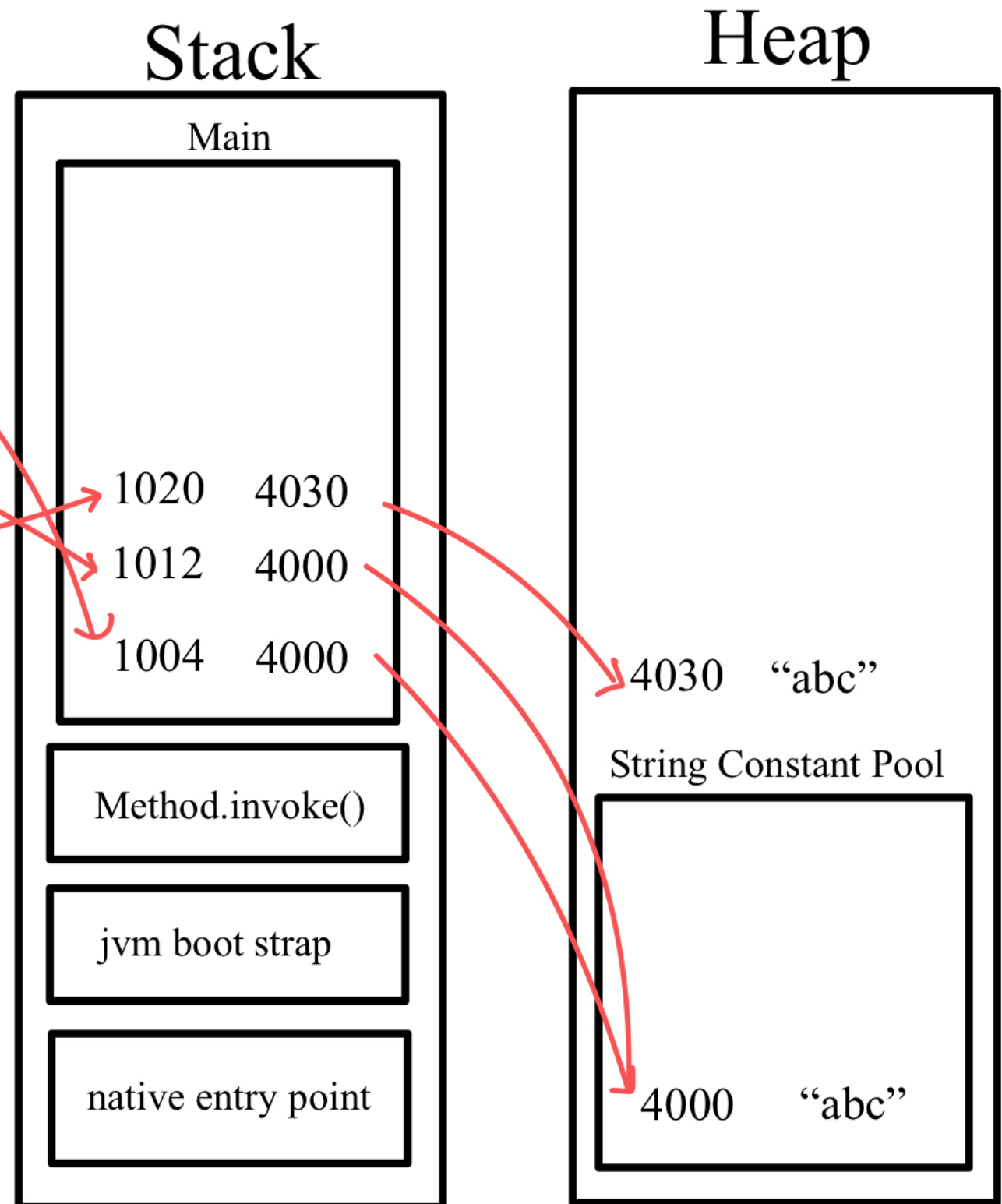
```
public class Main {  
    public static void main(String[] args) throws IOException {  
        String result = "";  
  
        for (int i=0;i<1000;i++) {  
            result += i;  
        }  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        String result;  
        StringBuilder sb = new StringBuilder();  
  
        for (int i=0;i<1000;i++) {  
            sb.append(i);  
        }  
  
        result=sb.toString();  
    }  
}
```

Data Type

- How Strings Are Stored at Compile Time

```
String a = "abc";  
String b = "abc";  
String c = new String("abc");
```



Data Type - How Strings Are Stored at Compile Time

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        String a = "abc";  
        String b = "abc";  
  
        System.out.println(a == b);  
    }  
}
```

String Literal은 메모리의 Heap 영역의 String Constant Pool 영역에 String 객체로써 저장된다.

리터널을 변수에 할당 할 때, 이미 String Constant Pool에 동일한 리터널이 있으면, 그 객체를 할당시킨다.

reference type 간의 == 연산자는 주소를 비교한다.

```
/Users/hyeonseok/Library/Java/JavaVirtual  
true  
  
Process finished with exit code 0
```

Ternary operator

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        int score = new Random().nextInt(bound: 100);  
  
        int bin = (64 < score) ? 1 : 0;  
    }  
}
```

(조건식) ? (참일 때) : (거짓 일 때)

단순 알고리즘 구현에서는 사용하는 경우가 없지만,
실제 프로젝트 구현시에는 많이 사용한다.

Bit Operator

Operator	Result
~	Bitwise unary NOT
&	Bitwise AND
	Bitwise OR
^	Bitwise exclusive OR
>>	Shift right
>>>	Shift right zero fill
<<	Shift left
&=	Bitwise AND assignment
=	Bitwise OR assignment
^=	Bitwise exclusive OR assignment
>>=	Shift right assignment
>>>=	Shift right zero fill assignment
<<=	Shift left assignment

신입생때는 비트 연산자를 많이 다룰 기회가 사실 없다.

멘토멘티 때라도, 많은 응용법을 알아 두면 좋을 것 같다.

Bit Operator

실제 개발 환경에서, 상태 관리를 유용하게 하기 위해서, 각 자리 별 비트의 상태를 통해, 다양한 상황이나 상태를 관리하기도 한다.

K번째, 비트의 상태를 잘 알아야 하는데, 어떻게 각 자리의 비트에 쉽게 접근 할 수 있을까?

기본적으로 K번째, 비트를 컨트롤 하기 위해서, $(1 \ll K)$ 의 수를 만든다.

해당 수를 사용해 목표 달성에 필요한 연산을 한다.

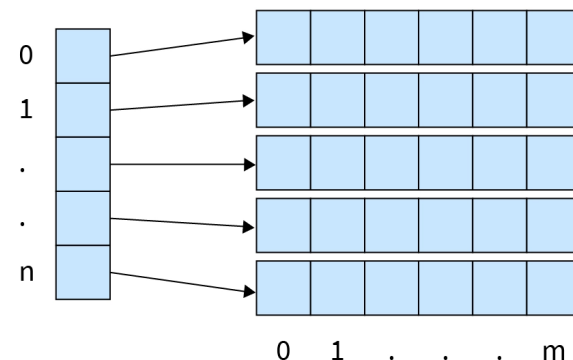
```
int G = 0b0110110101011111110100101101010;  
1 = 0b00000000000000000000000000000001;  
1 << 3 = 0b000000000000000000000000000001000;
```


Array Access Operator []

```
public class Main {  
    public static void main(String[] args) {  
        int N = 10, M = 20;  
        int [][]arr = new int[N][M];  
  
        System.out.println(arr[0].getClass().getTypeName());  
    }  
}
```

```
/Users/hyeonseok/Library/Java/JavaV  
int[]  
  
Process finished with exit code 0
```

놀랍게도...? arr[0] 은 int 배열이다...



Total Elements ($n * m$)

Table of Contents

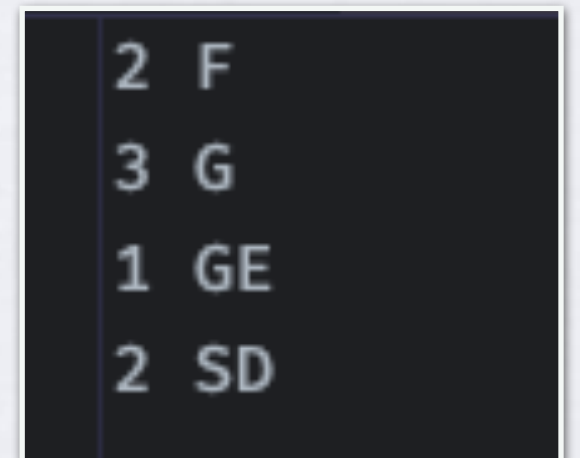
1. 복잡한 입력에 관한 입력처리
 - ▶ Scanner()
 - ▶ split() & StringTokenizer()
2. Speed Boosting..
 - A. BufferedReader() & BufferedWriter
3. Class..
 - ▶ about member Variables..
 - ▶ What is Constructor?
4. ArrayList and Sort Method..
5. Solving Exercise Problems..

Input Parsing

about Scanner.next()

다음과 같은 문제를 푸는 것을 목표로 한다.

입력이 다음과 같이 숫자와 문자열이 들어오면,
각 문자열을 들어온 숫자만큼 출력한다.



2	F
3	G
1	GE
2	SD

숫자와 문자열은 공백을 구분하여 라인단위로 주어지고, 들어온 문자
열은 공백없는 하나의 단위라고 가정한다.

Input Parsing

about Scanner.next()

Explain... Why this is incorrect..?

```
public static void main(String[] args) throws IOException {
    Scanner sc = new Scanner(new FileReader( fileName: "test.txt"));

    while(sc.hasNext()) {
        int x = sc.nextInt();
        String y = sc.nextLine();

        for (int i=0;i<x;i++) {
            System.out.println(y);
        }
    }
}
```

Explain... Why this is Correct..?

```
public static void main(String[] args) throws IOException {
    Scanner sc = new Scanner(new FileReader( fileName: "test.txt"));

    while(sc.hasNext()) {
        int x = sc.nextInt();
        String y = sc.next();

        while(x-->0) {
            System.out.println(y);
        }
    }
}
```

Input Parsing

about Scanner.next()

Scanner.next()

선행 delimiter (구분자) 를 모두 건너뛰다.
구분자가 나올 때 까지, 읽어들이는 값을 반환한다.

Scanner.nextLine()

가르키고 있는 커서의 다음부터,
개행 문자가 나올 때 까지, 읽어들이는 값을 반환한다.

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



return new String("HELLO")

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("1")

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("W")

Input Parsing

about Scanner.next()

Explain... Why this is incorrect..?

```
public static void main(String[] args) throws IOException {
    Scanner sc = new Scanner(new FileReader("test.txt"));

    while(sc.hasNext()) {
        int x = sc.nextInt();
        String y = sc.nextLine();

        for (int i=0;i<x;i++) {
            System.out.println(y);
        }
    }
}
```

```
/Users/hyeonseok/Li
A
A
A
B
B
C
D
D
D
D
```

Input Parsing

about Scanner.next()

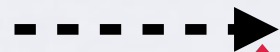
0	1	2	3	4	5	6	7	8	9	10	11	12
1	2		H	E	L	L	O	\n	1		W	\0



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



Call Scanner.next()

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

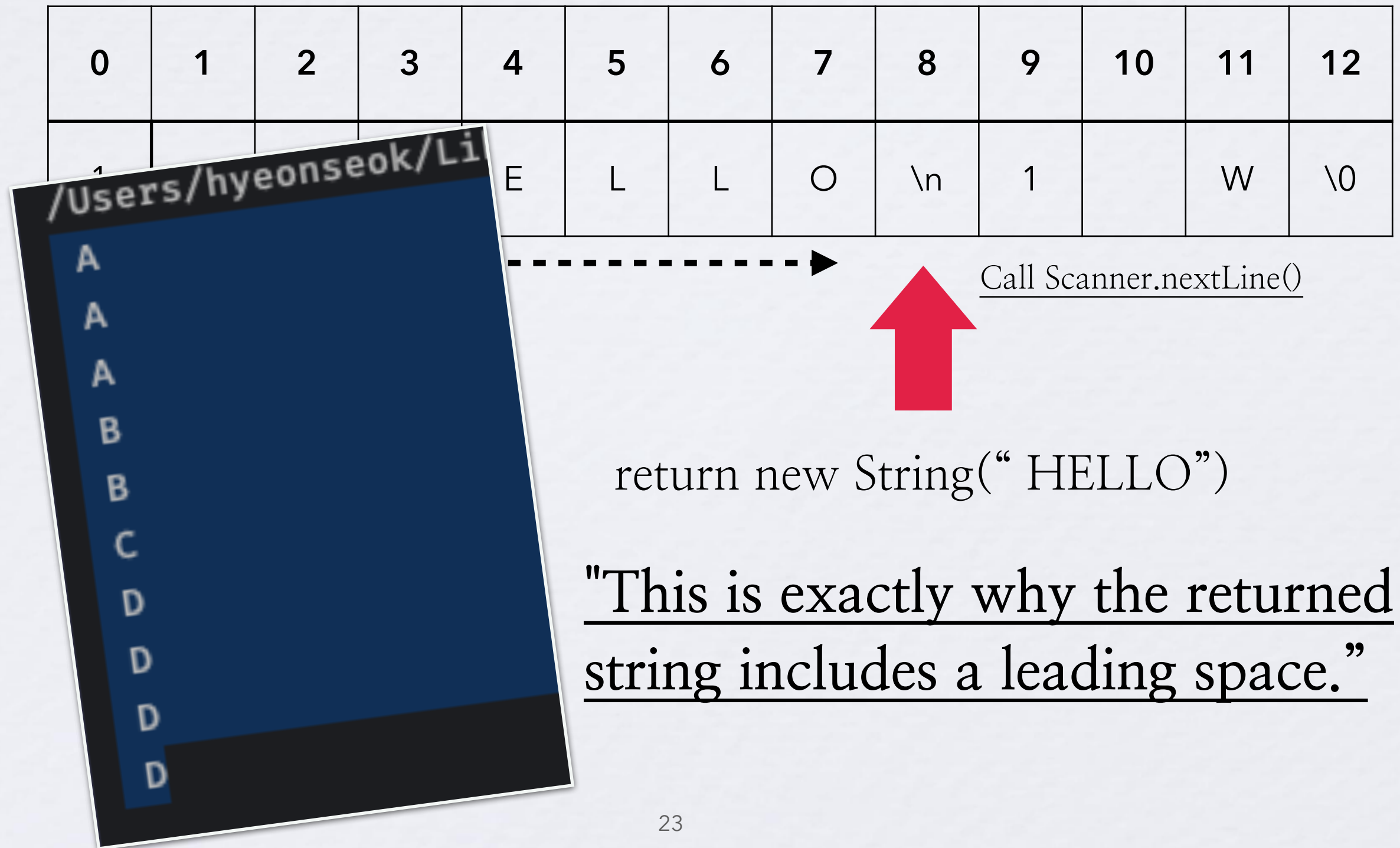


Call Scanner.nextLine()

return new String(" HELLO")

Input Parsing

about Scanner.next()



Input Parsing

about String.split()

Solving with using String.split(String regex, int limit)

```
[jshell> "Hello World 2019".split(" ");  
$34 ==> String[3] { "Hello", "World", "2019" }  
  
[jshell> "A1B2C3D4E5".split("[0-9]");  
$35 ==> String[5] { "A", "B", "C", "D", "E" }  
  
[jshell> "Hello World 2019".split(" ", 2);  
$36 ==> String[2] { "Hello", "World 2019" }  
  
[jshell> "  A1B2C3D ".split("[0-9]");  
$37 ==> String[4] { "  A", "B", "C", "D " }
```

limit의 의미

limit 값	의미
> 0	최대 limit개까지 나눔. 마지막 토큰은 남은 전체 문자열을 포함
= 0	모든 공백 제거, 마지막 빈 문자열도 제거됨 (기본 동작)
< 0	모두 나눔 + 빈 문자열 유지 (특히 끝에 빈 토큰들까지 유지됨)

<https://www.javastring.net/java/string/split-method>

Input Parsing

about String.split()

Solving with using String.split()

```
public static void main(String[] args) throws IOException {
    Scanner sc = new Scanner(new FileReader("test.txt"));

    while(sc.hasNext()) {
        String line = sc.nextLine();

        String[] items = line.split(" ");

        int x = Integer.parseInt(items[0]);
        while(x-->0) System.out.println(items[1]);
    }
}
```

© java.lang.String

```
@NotNull ↗
@Contract(pure = true) ↗
public @NotNull ↗ String[] split(
    @NotNull ↗ String regex
)
```

Splits this string around matches of the given [regular expression](#).

This method works as if by invoking the two-argument [split](#) method with the given expression and a limit argument of zero. Trailing empty strings are therefore not included in the resulting array.

Input Parsing

about StringTokenizer

Solving with using StringTokenizer(String str, String delim)

```
public static void main(String[] args) throws IOException {
    Scanner sc = new Scanner(new FileReader( fileName: "test.txt"));

    while(sc.hasNext()) {
        String line = sc.nextLine();

        StringTokenizer st = new StringTokenizer(line, delim: " ");

        int x = Integer.parseInt(st.nextToken());
        String sentence = st.nextToken();
        while(x-->0) System.out.println(sentence);
    }
}
```

똑같이, delimiter를 기준으로, 분할 해준다.

StringTokenizer.nextToken()으로 계속 다음원소를 호출 할 수 있다.

Input Parsing

about StringTokenizer

If there are no more elements and you call nextToken().....

```
while(sc.hasNext()) {  
    String line = sc.nextLine();  
  
    StringTokenizer st = new StringTokenizer(line, " ");  
  
    int x = Integer.parseInt(st.nextToken());  
    String sentence = st.nextToken();  
    while(x-->0) System.out.println(sentence);  
  
    //if call st.nextToken()...  
    //Exception will be occured.  
    st.nextToken();  
}
```

A
A
A

```
Exception in thread "main" java.util.NoSuchElementException Create breakpoint  
    at java.base/java.util.StringTokenizer.nextToken(StringTokenizer.java:349)  
    at Main.main(Main.java:126)
```

FAST I/O

about BufferedReader & BufferedWriter

4 15552번

제출

맞힌 사람

숏코딩

재채점 결과

채점 현황

내 제출

난이도 기여

질문 게시판

빠른 A+B

성공

☆

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초 (하단 참고)	512 MB	374286	169197	136169	45.423%

<https://www.acmicpc.net/problem/15552>

Before Start Learning BufferedReader...

Let's solve this simple problem

FAST I/O

about BufferedReader & BufferedWriter

```
public static void main(String[] args) throws IOException {  
    Scanner sc = new Scanner (System.in);  
    int T = sc.nextInt();  
  
    while(T-- > 0) {  
        int a = sc.nextInt();  
        int b = sc.nextInt();  
  
        System.out.println(a+b);  
    }  
}
```

Really Easy Problem.. but.

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
93010674	hhs2003	 15552	시간 초과			Java 11 / 수정	349 B	11초 전

FAST I/O

about BufferedReader & BufferedWriter

To solve this Problem, we should use more
faster method in Input / Output Object Class

```
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();  
    }  
}
```

FAST I/O

about System.out.println() vs BufferedWriter.flush()

항목	System.out.println()	BufferedWriter
 용도	콘솔(화면)에 간단한 디버깅, 출력용	더 빠르고 정제된 출력 처리에 적합
 출력 대상	콘솔 (<code>System.out</code>)	콘솔, 파일, 네트워크 등 다양하게 가능
 속도	상대적으로 느림 (한 줄마다 출력됨)	빠름 (출력 내용을 버퍼에 모아 한 번에 씀)
 자동 개행 여부	자동으로 줄 바꿈	<code>write()</code> 는 개행 안 함 → <code>\n</code> 직접 넣어야 함
 출력 형식 조정	문자열만 전달하면 됨	파일 출력, 다국어 문자, 속도 제어 등 유리
 출력 직후 반응	바로 콘솔에 나타남	버퍼에만 저장되어 <code>flush()</code> or <code>close()</code> 해야 진짜 출력됨

System.out.println() 은 매번, BufferedWriter.flush() 을 호출하는 것과 같다.

FAST I/O

about System.out.println() vs BufferedWriter.flush()

```
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();
}
```

Very inefficiently written code...

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
93011820	hhs2003	 15552	시간 초과			Java 11 / 수정	635 B	12초 전

FAST I/O

about BufferedReader & BufferedWriter

Writer 종류

의미

```
new OutputStreamWriter(System.out)
```

콘솔(화면)으로 출력

```
new FileWriter("file.txt")
```

파일로 출력

```
new OutputStreamWriter(socket.getOutputStream())
```

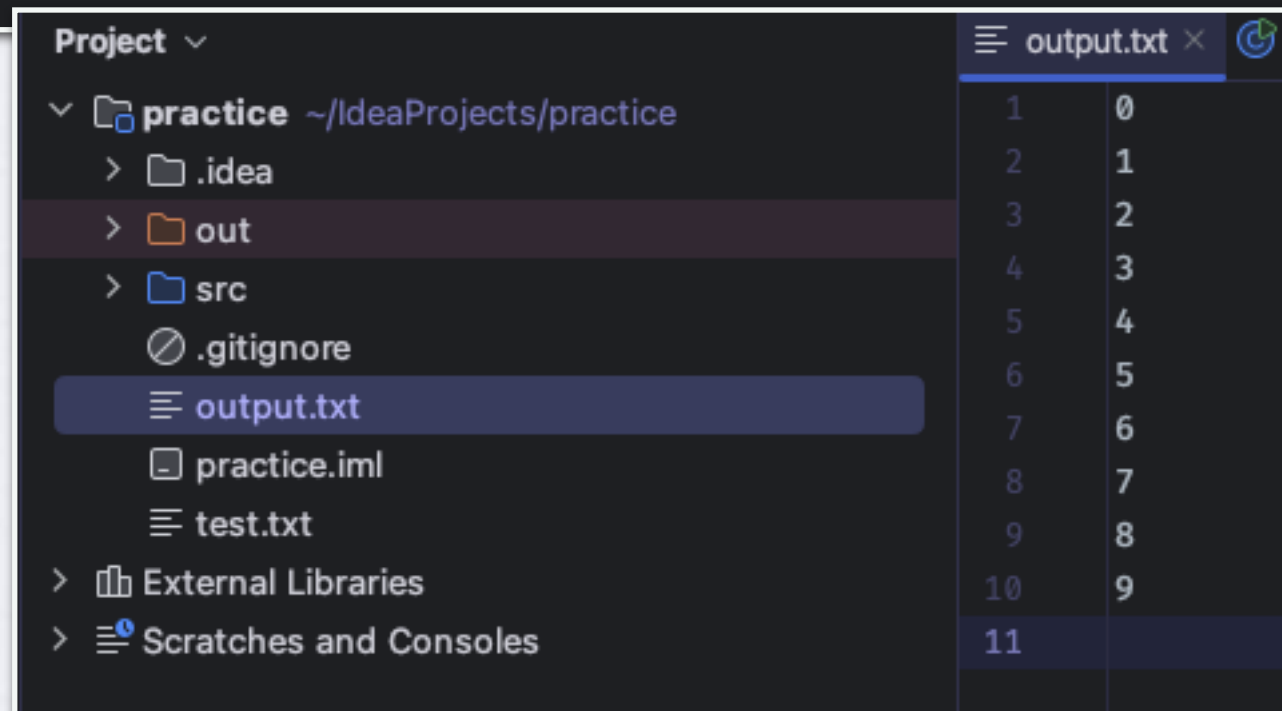
네트워크로 전송

FAST I/O

about BufferedReader & BufferedWriter

If you want to use this BufferedWriter for Writing file....

```
public static void main(String[] args) throws IOException {  
    BufferedWriter bw = new BufferedWriter(new FileWriter(new File( pathname: "output.txt")));  
  
    for (int i=0;i<10;i++) {  
        bw.write( str: i + "\n");  
    }  
    bw.flush();  
    bw.close();  
}
```



Class

What is Class...?

```
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);
    }
}
```

```
C:\Users\nyeonseok/Library/Java/javavirtua...
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

create new anonymous Class in StackFrame

```
4- class Point {  
5    int x, y;  
6  
7-    Point (int x, int y) {  
8        this.x=x;  
9        this.y=y;  
10   }  
11  
12   @Override  
13-   public String toString() {  
14       return x + " " + y;  
15   }  
16 }  
17  
18- public class Main {  
19-     public static void main(String[] args) throws IOException {  
20         final int Z = 5;  
21  
22-         System.out.println(  
23-             new Point(1, 2) {  
24                 int z = Z;  
25  
26-                 public String toString() {  
27                     return x + " " + z + " " + y;  
28                 }  
29             });  
30  
31     }  
32 }  
33  
34
```

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..?

```
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!
    }
}
```

```
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.");
        }
    });
}
```

ArrayList & Sort

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

5 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

```
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);
    }
}
```

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

ArrayList & Sort

```
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

```
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);
    }
}
```

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


ArrayList & Sort

```
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);
    }
}
```

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

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

ArrayList & Sort

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

2 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.

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

ArrayList & Sort

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

5 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.

다음은, 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) Engine을 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, 연결을 전부 DI가 담당

We've completed Week 2