

Java 프로그래밍

<지난 시간 개념 복습>

<반복자 - Iterator>

객체(Object)를 반복문 같이 돌려주는 것

Iterator(반복자)를 사용하게 됨으로 객체를 순회하면서 요소에 접근하기 위한 인터페이스입니다.

Iterator를 사용하여 컬렉션의 요소를 순차적으로 접근하고 수정할 수 있습니다.

<Stack & Queue>

Stack - FILO(먼저 들어간 자료가 나중에 나옴)

Queue - FIFO(먼저 들어간 자료가 먼저 나온다.)

<Exception - 예외처리>

```
package com.day10;
```

```
public class ExceptionTest01 {
```

```
    public static void main(String[] args) {
```

```
        String[] message = {"one","two","three"};
```

```
//        for(String m : message) {
```

```
//            System.out.println(m);
```

```
//        }
```

```
//        for(int i=0; i<message.length; i++) {
```

```
//            System.out.println(message[i]);
```

```
//        }
```

```
        String str = null;
```

```
        try {
```

```
            for(int i=0; i<10; i++) {
```

```
                System.out.println(message[i]);
```

```
            }
```

```
        System.out.println(str.length());
```

```
    }catch(NullPointerException n){
```

```
        System.out.println("Null 값"); // 예외 처리(Exception) try~catch로 사용
```

```
    }catch (ArrayIndexOutOfBoundsException a) {
```

```
        System.out.println("배열 범위 초과");
```

```
    }
```

```
    finally { // 반드시 수행되는 구문
```

```
        System.out.println("반드시 수행");
```

```
    }
```

```
}
```

```
}
```

```
package com.day10;
```

```
public class ExceptionTest02 {
```

```
    public static void main(String[] args) throws ArrayIndexOutOfBoundsException {  
        String[] msg = {"one","two","three"};  
        for(int i=0; i<10; i++) {  
            System.out.println(msg[i]);  
        }  
    }  
}
```

```
}
```

<Stream>

Byte 흐름 - Stream

단 방향성

입력과 출력이 존재

Byte 기반, 문자(Character) 기반이 있다.

입력

Byte 기반 - InputStream

문자 기반 - Reader

출력

Byte 기반 - OutputStream

문자 기반 - Writer

<InputStream>

```
package com.day10;
```

```
import java.io.IOException;
```

```
public class InputStreamTest {
```

```
    public static void main(String[] args) {  
        while (true) {  
            try {  
                int i = System.in.read(); // 표준 입력  
                if(i==-1) break;  
                System.out.print((char)i); // 표준 출력  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```
    }
```

```
}
```

```
    }  
}
```

<InputReader>

```
package com.day10;
```

```
import java.io.IOException;
```

```
import java.io.InputStreamReader;
```

```
public class InputReaderTest {  
    public static void main(String[] args) {  
        InputStreamReader isr = new InputStreamReader(System.in);  
        while (true) {  
            try {  
                int i=isr.read();  
                if(i== -1) break;  
                System.out.print((char)i);  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

<InputStream과 OutputStream 활용>

```
package com.day10;
```

```
import java.io.FileInputStream;
```

```
import java.io.FileNotFoundException;
```

```
import java.io.FileOutputStream;
```

```
import java.io.IOException;
```

```
public class FileCopyTest {  
  
    public static void main(String[] args) {  
        long millisecond = 0;  
  
        // p.497 try-with-resources(간결한 표현)  
        try (FileInputStream fis = new FileInputStream("src\\cap2.txt");  
             FileOutputStream fos = new FileOutputStream("src\\copy.zip")) {  
            millisecond = System.currentTimeMillis();  
            int i;  
            while ((i = fis.read()) != -1) {  
                fos.write(i);  
            }  
        }  
    }  
}
```

```

        }
        millisecond = System.currentTimeMillis() - millisecond;
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    System.out.println("복사시간 : " + millisecond + "millisecond 소요");
}

}

```

<BuffedStream>

```
package com.day10;
```

```

import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

```

// p.545

```

public class BuffedStreamTest {

    public static void main(String[] args) {
        long millisecond = 0;
        try(FileInputStream fis = new FileInputStream("src\\cap2.txt");
            FileOutputStream fos = new FileOutputStream("src\\cap4.zip");
            BufferedInputStream bis = new BufferedInputStream(fis);
            BufferedOutputStream bos = new BufferedOutputStream(fos)){
            millisecond = System.currentTimeMillis(); // Copy 전 소요 시각
            int i;
            while ((i=bis.read())!=-1) {
                bos.write(i);
            }
            millisecond = System.currentTimeMillis() - millisecond;
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println("파일 복사하는데 " +millisecond+"millisecond 소요되었습니다.");
    }
}

```

```
package com.day10;
```

```

import java.io.BufferedReader;
import java.io.IOException;

```

```

import java.io.InputStreamReader;
import java.io.PrintStream;

public class BuffedReaderTest {

    public static void main(String[] args) {
        // 한 줄 입력
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        PrintStream ps = null;
        String str;
        try {
            str = br.readLine();
            ps = new PrintStream("text1.txt"); // 파일로 한 줄 출력
            ps.print(str);
            System.out.println(str);

        } catch (IOException e) {
            e.printStackTrace();
        }

    }
}

```

<새로운 파일에 한 줄씩 읽어서 전체 출력하기>

```

package com.day10;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintStream;

// com.day10.Member.java 파일을 읽어서 output.txt로 내보내시오.
// 1 : package com.day10; ... 형식으로 출력
// BufferedReader, PrintStream 사용하기

```

```

public class LineFile {

    public static void main(String[] args) {
        try(BufferedReader br = new BufferedReader(new
FileReader("src\\com\\day10\\Member.java"));
            PrintStream ps = new PrintStream("output.txt")){
            String str;
            int cnt = 1;
            while ((str = br.readLine()) != null) {
                ps.println(cnt++ + " : " +str);
                //cnt++
            }

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e1) {
            e1.printStackTrace();
        }

    }
}

```