

1. 텍스트 파일의 내용을 읽어 콘솔에 출력하는 코드를 작성한다.

- ① 바이트 스트림 사용 1: `FileInputStream`과 `BufferedInputStream`을 파이프라인으로 구성한 후 `read()` 메서드를 사용한다.

<입력 파일>

```
sample.txt
1 FileInputStream
2 BufferedInputStream
3 파일인풋스트림
4 버퍼드인풋스트림
```

<실행 결과>

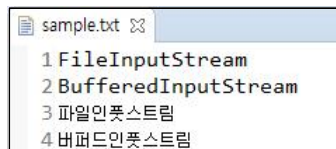
```
FileInputStream
BufferedInputStream
Æ?????2½Æ@. 2
1?Æ?????2½Æ@. 2
```

<코드 예시>

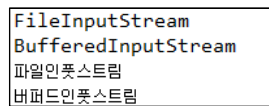
```
1 import java.io.BufferedInputStream;
2 import java.io.FileInputStream;
3 import java.io.IOException;
4
5 public class ByteStreamTest1 {
6
7     public static void main(String[] args) {
8         String fileName = null;
9         int b;
10        fileName = "sample.txt";
11
12        try(FileInputStream fis = new FileInputStream(fileName);
13            BufferedInputStream bis = new BufferedInputStream(fis)){
14            while(bis.available() > 0) {
15                b = bis.read();
16                System.out.printf("%c", b);
17            }
18
19        } catch(IOException e) {
20            System.out.println(e.getMessage());
21        }
22    }
23 }
24
25 }
```

- ② 바이트 스트림 사용 2: 텍스트 파일을 읽어 행단위로 출력한다. FileInputStream과 BufferedInputStream을 파이프라인으로 구성한 후 read(byte[] b) 메서드를 사용한다.

<입력 파일>



<실행 결과>



※ 입력 파일의 인코딩 형식은 운영체제의 기본 형식으로 지정(윈도우의 경우 MS949)

<코드 예시>

```
1 import java.io.BufferedInputStream;
2 import java.io.FileInputStream;
3 import java.io.IOException;
4
5 public class ByteStreamTest2 {
6
7     public static void main(String[] args) {
8         String fileName = "sample.txt";
9         byte b[] = new byte[100];
10
11         try(FileInputStream fis = new FileInputStream(fileName);
12             BufferedInputStream bis = new BufferedInputStream(fis)){
13             while(bis.read(b) != -1) {
14                 System.out.print(new String(b));
15             }
16         } catch(IOException e) {
17             System.out.println(e.getMessage());
18         }
19     }
20
21 }
22
23 }
```

- ③ 문자 스트림 사용: `FileReader`와 `BufferedReader`를 파이프라인으로 구성한 후 `readline()` 메서드를 이용한다. 단, 입력 파일과 출력 결과 및 입력 파일의 인코딩 형식은 ②와 동일하다.

<코드 예시>

```
1 import java.io.BufferedReader;
2 import java.io.FileReader;
3 import java.io.IOException;
4
5 public class ReaderTest {
6
7     public static void main(String[] args) {
8         String fileName = "sample.txt";
9         String str = null;
10
11         try(FileReader fr = new FileReader(fileName);
12             BufferedReader br = new BufferedReader(fr)){
13             while( (str = br.readLine()) != null ) {
14                 System.out.println(str);
15             }
16         }catch(IOException e) {
17             System.out.println(e.getMessage());
18         }
19     }
20
21 }
22
23 }
```

- ④ 스트림 사용: Files 클래스의 lines()메서드와 File 클래스의 toPath() 메서드를 이용해 Stream을 생성한 후 출력한다. 단, 입력 파일의 인코딩 방식은 UTF-8로 지정한다.

<코드 예시>

```
1 import java.io.File;
2 import java.io.IOException;
3 import java.nio.file.Files;
4 import java.util.stream.Stream;
5
6 public class FilesTest {
7
8     public static void main(String[] args) {
9         String fileName = "sample2.txt";
10        File file = new File(fileName);
11
12        /* 기본 스타일 */
13        try(Stream<String> s = Files.lines(file.toPath());){
14            s.forEach(x -> System.out.println(x));
15        }catch(IOException e) {
16            System.out.println(e.getMessage());
17        }
18
19        /* 축약 스타일 */
20        try{
21            Files.lines(new File(fileName).toPath()).forEach(System.out::println);
22        }catch(IOException e) {
23            System.out.println(e.getMessage());
24        }
25    }
26 }
27
28 }
```