14-3. 입출력 관련 API

혼자 공부하는 자바 (신용권 저)

목차

시작하기 전에

[핵심 키워드] : System.in, System.out, Scanner, File

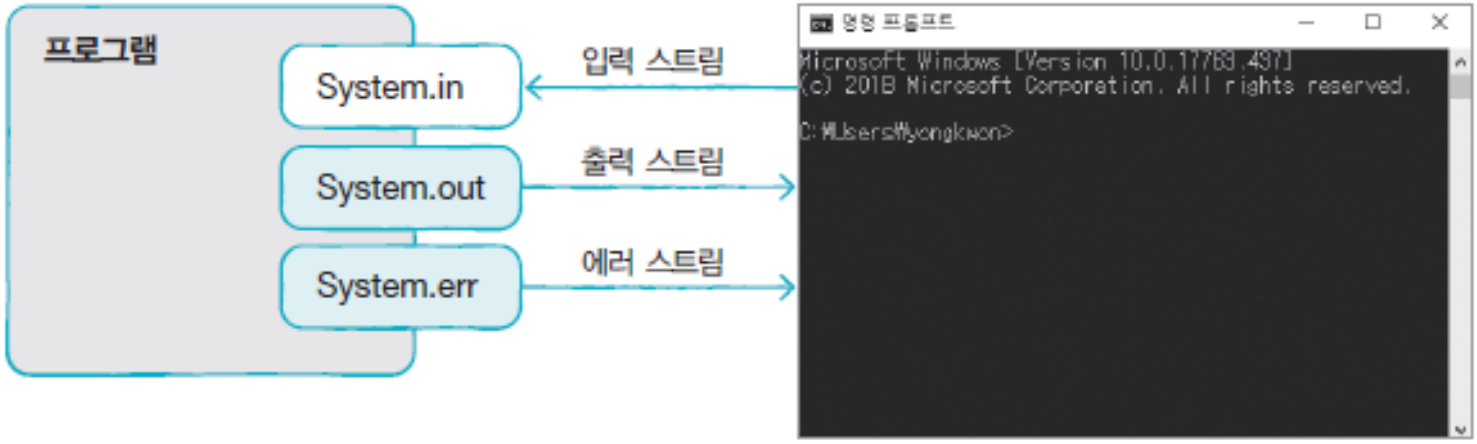
[핵심 포인트]

자바 표준 API에서는 스트림을 이용해 다양한 기능을 제공한다. 대표적으로 콘솔에서 입출력에 사용되는 System.in은 InputStream 타입이고, System.out은 PrintStream 타입이다. 콘솔 입출력 시 보조 스트림을 사용하는 방법과 파일 입출력 시 부가적인 정보를 제공하는 File 클래스에 대해 알아본다.

시작하기 전에

❖ 콘솔 (Console)

- 시스템 사용하기 위해 키보드로 입력받고 모니터로 출력하는 소프트웨어
- 자바는 콘솔로부터 데이터 입력받을 때 System.in 사용하고, 콘솔에 데이터 출력할 때 System.out 사용
- 에러 출력 시에는 System.err 사용



System.in 필드

- 자바는 콘솔에서 키보드 데이터 입력받을 수 있도록 System 클래스의 in 정적 필드 제공
  - System.in은 InputStream 타입의 필드이므로 다음과 같이 InputStream 변수로 참조 가능

```
InputStream is = System.in;
```

  - 키보드로부터 어떤 키가 입력되었는지 확인하려면 InputStream의 read() 메소드로 1byte 읽음

```
int keyCode = is.read();
```

  - [Enter]키 입력 후 라인 단위로 전체 문자열 읽는 방식으로 변경할 경우

```
InputStream is = System.in;
Reader reader = new InputStreamReader(is);
BufferedReader br = new BufferedReader(reader);
```

    - InputStreamReader 보조스트림 연결하여 Reader로 변환 후 라인 단위로 읽기 위해 BufferedReader 보조 스트림 추가 연결

```
String lineStr = br.readLine();
```

System.in 필드

- 예시 – 키보드로부터 라인 단위 문자열 얻기

```
01  package sec03.exam01;
02
03  import java.io.*;
04
05  public class GetLineStringFromKeyboard {
06    public static void main(String[] args) throws Exception {
07      InputStream is = System.in;
08      Reader reader = new InputStreamReader(is);
09      BufferedReader br = new BufferedReader(reader);
10
11      while(true) {
12        System.out.print("입력하세요: ");
13        String lineStr = br.readLine();
14        if(lineStr.equals("q") || lineStr.equals("quit")) break;
15        System.out.print("입력된내용: " + lineStr);
16        System.out.println();
17      }
18
19      br.close();
20    }
21  }
```

InputStream을 Reader로 변환하고 다시 BufferedReader를 연결

라인 단위로 문자열을 읽음

q 또는 quit를 읽었을 때 while 반복문 종료

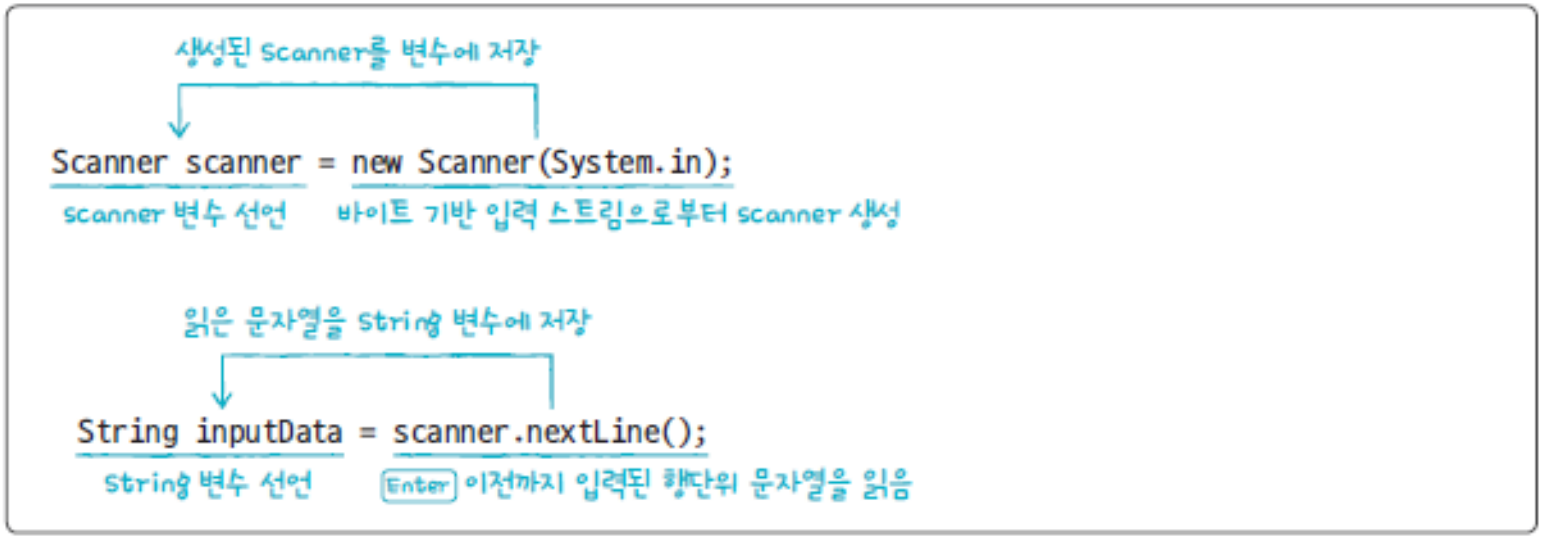
입력 스트림 닫기

실행결과

```
입력하세요: System.in에
입력된내용: System.in에
입력하세요: 보조 스트림을 연결해서
입력된내용: 보조 스트림을 연결해서
입력하세요: 편리하게 라인 단위로 입력 받을 수 있습니다.
입력된내용: 편리하게 라인 단위로 입력 받을 수 있습니다.
입력하세요: q
```

System.out 필드

- 콘솔에서 모니터로 데이터 출력하기 위해 System 클래스의 out 정적 필드 사용
  - PrintStream이 제공하는 print(), println(), printif() 등 메소드 사용



System.out 필드

- 예시 – Scanner로 입력된 문자열 얻기

```
01  package sec03.exam02;
02
03  public class Product {
04    private int pno;
05    private String name;
06    private int price;
07    private int stock;
08
09    public int getPno() { return pno; }
10    public void setPno(int pno) { this.pno = pno; }
11    public String getName() { return name; }
12    public void setName(String name) { this.name = name; }
13    public int getPrice() { return price; }
14    public void setPrice(int price) { this.price = price; }
15    public int getStock() { return stock; }
16    public void setStock(int stock) { this.stock = stock; }
17  }
```

System.out 필드

```
01  package sec03.exam02;
02
03  import java.util.*;
04
05  public class ProductStorage {
06    private List<Product> list = new ArrayList<>();  <-- List 컬렉션 생성
07    private Scanner scanner = new Scanner(System.in);  <-- 키보드 입력 Scanner 생성
08    private int counter;  <-- pno 제공 카운터
09
10    public void showMenu() {
11      while(true) {
12        System.out.println("--------------------------------");
13        System.out.println("1.등록  |  2.목록  |  3.종료");
14        System.out.println("--------------------------------");
15
16        System.out.print("선택: ");
17        String selectNo = scanner.nextLine();  <-- 번호 읽기
18        switch(selectNo) {
19          case "1": registerProduct(); break;  <-- Product 등록
20          case "2": showProducts(); break;  <-- 등록된 모든 Product 정보 보기
21          case "3": return;  <-- 프로그램 종료
22        }
23      }
24    }
25
```

System.out 필드

```
26      public void registerProduct() {
27        try {
28          Product product = new Product();
29          product.setPno(++counter);
30
31          System.out.print("상품명: ");
32          product.setName(scanner.nextLine());
33
34          System.out.print("가격: ");
35          product.setPrice(Integer.parseInt(scanner.nextLine()));
36
37          System.out.print("재고: ");
38          product.setStock(Integer.parseInt(scanner.nextLine()));
39
40          list.add(product);
41        } catch(Exception e) {
42          System.out.println("등록 에러: " + e.getMessage());
43        }
44    }
45
46      public void showProducts() {
47        for(Product p : list) {
48          System.out.println(p.getPno() + "\t" + p.getName() + "\t" +
49                             p.getPrice() + "\t" + p.getStock());
50        }
51      }
52    }
```

pno 세팅

이름을 읽고 세팅

가격을 읽고 세팅

재고를 읽고 세팅

list에 Product 저장

list에 저장된 모든 Product 정보를 모니터에 출력

System.out 필드

```
01  package sec03.exam02;
02
03  public class ProductStorageExample {
04    public static void main(String[] args) {
05      ProductStorage productStoreage = new ProductStorage();
06      productStoreage.showMenu();
07    }
08  }
```

ProductStorage 객체를 생성하고 showMenu() 메소드 호출

실행결과

```
---------------------------------
1.등록  |  2.목록  |  3.종료
---------------------------------
선택: 1
상품명: 마우스
가격: 10000
재고: 5
---------------------------------
1.등록  |  2.목록  |  3.종료
---------------------------------
선택: 1
상품명: 키보드
가격: 15000
재고: 7
---------------------------------
1.등록  |  2.목록  |  3.종료
---------------------------------
선택: 2
1      마우스 10000       5
2      키보드 15000       7
---------------------------------
1.등록  |  2.목록  |  3.종료
---------------------------------
선택: 3
```

File 클래스

❖ java.io 패키지에서 제공하는 File 클래스는 파일 및 폴더 정보 제공 역할 함

```
File file = new File("C:/Temp/file.txt");
File file = new File("C:\\Temp\\file.txt");
```

- 경로 구분자는 운영체제마다 조금씩 다름
  - 윈도우에서는 \ 및 / 모두 사용 가능

- 해당 경로에 실제로 파일이나 폴더 있는지 확인하려면 File 객체 생성후 exists() 메소드 호출

```
boolean isExist = file.exists();
```

- exists() 메소드의 리턴값이 false일 경우 다음 메소드로 파일 혹은 폴더 생성

리턴 타입	메소드	설명
boolean	createNewFile()	새로운 파일을 생성합니다.
booelan	mkdir()	새로운 폴더를 생성합니다.
boolean	mkdirs()	경로상에 없는 모든 폴더를 생성합니다.

File 클래스

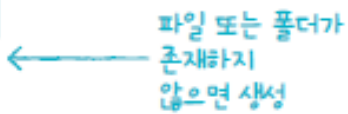
- exists() 메소드의 리턴값이 true라면 다음 메소드 사용 가능

리턴 타입	메소드	설명
boolean	delete()	파일 또는 폴더를 삭제합니다.
boolean	canExecute()	실행할 수 있는 파일인지 여부를 확인합니다.
boolean	canRead()	읽을 수 있는 파일인지 여부를 확인합니다.
boolean	canWrite()	수정 및 저장할 수 있는 파일인지 여부를 확인합니다.
String	getName()	파일의 이름을 리턴합니다.
String	getParent()	부모 폴더를 리턴합니다.
File	getParentFile()	부모 폴더를 File 객체로 생성 후 리턴합니다.
String	getPath()	전체 경로를 리턴합니다.
boolean	isDirectory()	폴더인지 여부를 확인합니다.
boolean	isFile()	파일인지 여부를 확인합니다.
boolean	isHidden()	숨김 파일인지 여부를 확인합니다.
long	lastModified()	마지막 수정 날짜 및 시간을 리턴합니다.
long	length()	파일의 크기를 리턴합니다.
String[]	list()	폴더에 포함된 파일 및 서브 폴더 목록 전부를 String 배열로 리턴합니다.
String[]	list(FilenameFilter filter)	폴더에 포함된 파일 및 서브 폴더 목록 중에 FilenameFilter에 맞는 것만 String 배열로 리턴합니다.
File[]	listFiles()	폴더에 포함된 파일 및 서브 폴더 목록 전부를 File 배열로 리턴합니다.
File[]	listFiles(FilenameFilter filter)	폴더에 포함된 파일 및 서브 폴더 목록 중에 FilenameFilter에 맞는 것만 File 배열로 리턴합니다.

File 클래스

- 예시 – File 클래스 이용한 파일 및 폴더 정보 출력

```
01  package sec03.exam03;
02
03  import java.io.File;
04  import java.text.SimpleDateFormat;
05  import java.util.Date;
06
07  public class FileExample {
08    public static void main(String[] args) throws Exception {
09      File dir = new File("C:/Temp/images");
10      File file1 = new File("C:/Temp/file1.txt");
11      File file2 = new File("C:/Temp/file2.txt");
12      File file3 = new File("C:/Temp/file3.txt");
13
14      if(dir.exists() == false) {  dir.mkdirs();  }
15      if(file1.exists() == false) {  file1.createNewFile();  }
16      if(file2.exists() == false) {  file2.createNewFile();  }
17      if(file3.exists() == false) {  file3.createNewFile();  }
18
19      File temp = new File("C:/Temp");
20      File[] contents = temp.listFiles();
```



File 클래스

```
21
22        System.out.println("시간\t\t\t형태\t\t크기\t이름");
23        System.out.println("-------------------------------------------------------");
24        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd a HH:mm");
25        for(File file : contents) {
26          System.out.print(sdf.format(new Date(file.lastModified())));
27          if(file.isDirectory()) {
28            System.out.print("\t<DIR>\t\t\t" + file.getName());
29          } else {
30            System.out.print("\t\t\t" + file.length() + "\t" + file.getName());
31          }
32          System.out.println();
33        }
34      }
35    }
```

파일 또는 폴더 정보를 출력

실행결과

```
시간                    형태      크기    이름
-------------------------------------------------------
2019-05-07 PM 14:10             382     board.db
2019-05-07 PM 14:30             0       file1.txt
2019-05-07 PM 14:30             0       file2.txt
2019-05-07 PM 14:30             0       file3.txt
2019-05-07 PM 14:30     <DIR>           images
2019-05-07 PM 13:57             76      printstream.txt
```

File 클래스

- File 객체는 파일 입출력 스트림 객체 생성 시 경로 정보 제공 목적으로도 사용됨
  - 스트림 생성자에 문자열 경로 대신 File 객체를 대입

```
//첫 번째 방법
FileInputStream fis = new FileInputStream("C:/Temp/image.gif");

//두 번째 방법
File file = new File("C:/Temp/image.gif");
FileInputStream fis = new FileInputStream(file);
```

키워드로 끝내는 핵심 포인트

- System.in : 자바는 콘솔에서 키보드의 데이터 입력받을 수 있도록 System 클래스의 in 정적 필드 제공함. 주로 InputStreamReader 보조 스트림과 BufferedReader 보조 스트림을 연결해서 사용하거나 Scanner를 이용해서 입력된 문자열 읽음

- System.out : 콘솔에서 키보드로 입력된 데이터를 System.in로 읽었다면 반대로 콘솔에서 모니터로 데이터 출력하기 위해서는 System 클래스의 out 정적 필드 사용. PrintStream이 제공하는 print(), println(), printf()와 같은 메소드 이용하여 모니터로 출력 가능.

- Scanner : Scanner 클래스는 입출력 스트림도, 보조 스트림도 아님. Scanner는 문자 파일이나 바이트 기반 입출력 스트림에서 라인 단위 문자열을 쉽게 읽도록 하기 위해 java.util 패키지에서 제공하는 클래스.

- File : java.io 패키지에서 제공하는 File 클래스는 파일 및 폴더 정보 제공하는 역할 함.

확인문제

❖ 아래 메뉴를 가지고 있는 콘솔 게시판을 만들어보세요. 예외가 발생하여 프로그램이 종료되지 않도록 예외 처리에 주의하기 바랍니다.

```
------------------------------------------------------------------
1. 목록보기 | 2.상세보기 | 3.수정하기 | 4.삭제하기 | 5.파일저장 | 6.종료
------------------------------------------------------------------
선택:
```

① 추천 소스 파일:

```
Board.java, BoardService.java, BoardServiceExample.java
```

② 추천 API:

```
List, ArrayList, Scanner,
FileOutputStream, FileInputStream, ObjectOutputStream, ObjectInputStream
```

Thank You !

혼자 공부하는 자바 (신용권 저)