

다양한 예제들

scanner

- 스캐너

- 정의 : cmd 창에서 사용자의 입력을 받는데 사용되는 클래스
- 사용하려면 : `java.util.Scanner` 클래스 импорт
=> `import java.util.Scanner`

- 객체생성

- 스캐너 클래스로 객체를 생성할 때는 매개변수로 `System.in`을 받음(키보드로 입력받겠다는 뜻)
- `Scanner sc = new Scanner(System.in);`

scanner

- 메소드

- 스캐너 클래스는 제대로 입력이 되고 있는지 입력값을 보여줄 필요가 있음
- 입력되는 데이터의 종류에 따라 다른 메소드를 사용함

메소드	설명
nextBoolean()	사용자가 입력한 불린 값을 보여 줌
nextByte()	사용자가 입력한 바이트값을 보여 줌
nextDouble()	사용자가 입력한 더블형 숫자를 보여 줌
nextFloat()	사용자가 입력한 플로트형 숫자를 보여 줌
nextInt()	사용자가 입력한 정수값을 보여 줌
nextLine()	사용자가 입력한 문자열을 보여 줌

예제1

```
import java.util.Scanner;

class ScannerTest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("이름을 입력하세요");
        String name = sc.nextLine(); // 문자열 입력

        System.out.println("거주지를 입력하세요");
        String area = sc.nextLine();

        System.out.println("나이를 입력하세요");
        int age = sc.nextInt(); // 정수 입력

        System.out.println("");
        System.out.println("입력정보");

        System.out.println("이름: " + name);
        System.out.println("거주지: " + area);
        System.out.println("나이: " + age);
    }
}
```

이름을 입력하세요
홍길동
거주지를 입력하세요
서울
나이를 입력하세요
3333

입력정보
이름: 홍길동
거주지: 서울
나이: 3333

예제2

실행화면=====→
오른쪽과 같이 프로그램이 실행
되고 있는 중에는 계속
1~4의 메뉴와 선택이 나오고
선택이 1~4로 선택되는 것에 따라
1 : "예금액>" 글씨 출력
입력되는 숫자를 sum에 더해줌
2 : "출금액>" 글씨 출력
입력되는 sum에서 숫자를 빼줌
3 : "잔고>" 글씨 출력
sum 변수의 내용 출력
4 : "프로그램 종료"를 출력 후 더 이상
1~4까지의 선택 내용이 나오지 않음.

switch~case 문 이용

```
-----  
1.예금 | 2.출금 | 3.잔고 | 4.종료  
-----
```

```
선택> 1  
예금액> 10000
```

```
-----  
1.예금 | 2.출금 | 3.잔고 | 4.종료  
-----
```

```
선택> 2  
출금액> 4000
```

```
-----  
1.예금 | 2.출금 | 3.잔고 | 4.종료  
-----
```

```
선택> 3  
잔고> 6000
```

```
-----  
1.예금 | 2.출금 | 3.잔고 | 4.종료  
-----
```

```
선택> 4  
|  
프로그램 종료
```

예제2

```
1 package sec02.verify.exam07;
2
3 import java.util.Scanner;
4
5 public class Exam07 {
6     public static void main(String[] args) {
7         boolean run = true;
8
9         int balance = 0;
10
11         Scanner scanner = new Scanner(System.in);
12
13         while (run) {
14             System.out.println("-----");
15             System.out.println("1.예금 | 2.출금 | 3.잔고 | 4.종료");
16             System.out.println("-----");
17             System.out.print("선택> ");
18
19             int menuNum = Integer.parseInt(scanner.nextLine());
```

예제 2

```
21         switch (menuNum) {
22             case 1:
23                 System.out.print("예금액> ");
24                 balance += Integer.parseInt(scanner.nextLine());
25                 break;
26             case 2:
27                 System.out.print("출금액> ");
28                 balance -= Integer.parseInt(scanner.nextLine());
29                 break;
30             case 3:
31                 System.out.print("잔고> ");
32                 System.out.println(balance);
33                 break;
34             case 4:
35                 run = false;
36                 break;
37         }
38
39         System.out.println();
40     }
41
42     System.out.println("프로그램 종료");
43 }
44 }
```

예제3 – 파일생성

```
1 package exercise;
2 import java.io.*;
3
4 public class CreateFile {
5     public static void main(String[] args) {
6         try {
7             File f = new File("test.txt");
8             if (f.createNewFile()) {
9                 System.out.println("파일 생성: " + f.getName());
10            } else {
11                System.out.println("같은 이름의 파일이 이미 존재합니다.");
12            }
13        } catch (IOException e) {
14            System.out.println("오류가 발생했습니다.");
15            e.printStackTrace();
16        }
17    }
18 }
```

- 자바에서 파일생성은 File 클래스를 통해 이루어짐
- File 클래스는 java.io 패키지를 임포트해 사용가능
- File 클래스는 파일이름을 매개변수로하는 생성자로 객체 생성

File 변수 = new File("파일명");

- createNewFile() 메소드는 빈 파일을 만들어 낼 때 사용 하며 Boolean 값 반환
- f.getName() 메소드는 파일 이름 반환

예제4 – 파일에 내용 쓰기

```
1 package exercise;
2 import java.io.*;
3
4 public class WriteTest {
5     public static void main(String[] args) {
6         try {
7             FileWriter fr = new FileWriter("test.txt");
8             fr.write("안녕하세요? 저는 자바를 배웁니다.\n"
9                     + "만나서 반갑습니다.\n"
10                    + "파일쓰기 잘 되지요?");
11             fr.close();
12             System.out.println("파일 쓰기에 성공했습니다^^.");
13         } catch (IOException e) {
14             System.out.println("에러가 발생했습니다.");
15             e.printStackTrace();
16         }
17     }
18 }
```

- 자바에서 파일쓰기는 FileWriter 클래스의 write()메소드를 통해 이루어짐
- FileWriter 클래스는 java.io 패키지를 임포트해 사용가능
- write() 메소드의 매개변수로 쓰기 내용 씬.

파일객체.write(쓰기 내용);

- close() 메소드로 파일을 꼭 닫아줘야 함.
- try..... catch 구문을 통해 꼭 오류처리를 해줘야 함.