JAVA 방중교육

2. Java 설치와 자료형

방중교육 진행자: 김준성

TO-DO LIST:

- ☑ 조건문
- ☑ 반복문
- ☑ continue와 break
- ☑ 배열
- ☑ 예외 처리

조건문

조건문에는 if와 switch가 있습니다.

괄호 안에 if는 제약이 없는 반면, switch는 정수, 문자, 문자열만 들어올 수 있습니다.

그러나 조건이 많아질수록 if는 switch보다 속도가 느려집니다.

즉, if는 유연성이 좋지만 속도가 느리고, switch는 유연성이 좋지 않지만 속도가 빠릅니다.

조건문

```
String goods = “콜라”

int price = 0;

if(goods == “콜라” || goods == “사이다") {

    price = 1200;

}

else if(goods == “물”) {

        price = 500;

}

else {

        System.out.println(“요청하신 제품은 존재하지 않습니다.”)

}
```

조건문

```
String goods = “콜라”

int price = 0;

switch(goods) {

    case “콜라”:

    case “사이다”:

        price = 1200;

        break;

    case “물”:

        price = 500;

        break;

    default:

        System.out.println(“요청하신 제품은 존재하지 않습니다.”);

}
```

반복문

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

        print(i + " ");

}

while(i<=10) {

        print(i + " ");

        i++

}
```

continue와 break

continue는 자신 이하의 명령을 생략하고 다음 구간으로 넘어가는 명령어

break는 더 이상 자신 이하의 명령을 생략하고 중괄호{} 밖으로 나오는 명령어

```
while(True) {
        if(i % 2 == 0) {
            continue;
        }
        print(i);
        i++;
        if(i>=10){
            break;
        }
}
```

배열

C와 달리, Java에서는 배열 선언 시, 배열에 대한 (레퍼런스) 변수 선언과 배열의 저장 공간 할당이 개별적으로 이루어집니다.

ex)
int arr[] 이라는 변수를 선언하고 new int[3];을 통해 저장 공간 할당을 해야 합니다.

```
int arr[];
arr = new int[3];
```

```
int arr[] = new int[3];
```

만약 C처럼 배열 선언과 저장 공간 할당을 동시에 할 경우 컴파일 에러가 일어납니다.

```
int arr[3]; <- 컴파일 오류, 배열크기 지정 불가.
```

배열

int arr[]이랑 int[] arr, 두 가지 방법으로 변수를 선언할 수 있습니다.
int arr[] = new int[3];　　int[] arr = new int[3];

배열을 초기화 할 때, 선언문에서 중괄호를 사용하면 초기화된 배열을 만들 수 있습니다.
int arr[] = {1, 2, 3, 4, 5}

Java에서는 레퍼런스 변수와 배열 공간이 분리되어서 다수의 변수가 하나의 배열 공간을 가리키는 배열 공유가 쉽게 이루어 집니다.

int arr[] = new int[5];
int testArr[] = arr;
arr[1] = 2;
testArr[1] = 10;

이 경우, arr과 testArr은 배열을 공유하고 있기 때문에, arr[1]과 testArr[1]은 둘 다 100이 됩니다.

그리고 arr.length;를 사용하여 배열의 총 인덱스 수를 쉽게 알 수 있습니다.
System.out.println(arr.length); **-> 5가 나옴**

예외처리

예외처리는 컴파일 오류로 걸러지지 않는 문제들을
ex) 정수를 0으로 나눌 때, 정수를 입력해야되는데 문자열을 입력할 때 등…
처리하기 위한 것입니다.

```
try {
        예외가 발생할 것 같은 명령
}
catch(처리할 예외 타입 선언)  {
        예외처리
}
finally {
        예외 발생 여부와 상관없이 마지막에 (무조건) 실행되는 명령
}
```

예외처리

예외 타입	설명	패키지
ArithmeticException	정수를 0으로 나눌 때	java.lang
NullPointerException	null 레퍼런스를 참조할 때	java.lang
ClassCastException	변환할 수 없는 타입으로 객체를 변환할 때 발생	java.lang
ArrayIndexOutOfBoundException	배열 범위를 벗어난 접근을 할 때	java.lang
InputMismatchException	정수를 입력받으려고 했는데, 사용자가 문자를 입력했을 때	java.util
NumberFormatException	문자열이 나태내는 숫자와 일치하지 않는 타입의 숫자로 변환 시 발생	java.lang

예외처리

```
try {
        int i = scanner.nextInt();
        System.out.println("입력하신 숫자는 " + i + "입니다.");
}
catch(InputMismatchException ex) {
        System.out.println("숫자를 입력해주시기 바랍니다.");
}
finally {
        System.out.println("시스템을 종료합니다);
}
```