

## 31강 Package 11/15

### 1. 패키지

#### (1) 패키지의 정의

① 패키지는 많은 소스들을 개발자 나름대로의 기준을 삼아 관리하기 위한 방법.

② 폴더단위로 구분

③ 같은 패키지에 있는 클래스는 자유롭게 사용가능, 패키지가 다른 클래스일 경우, 패키지를 Import를 통해 소스에 포함시켜주면 사용가능하다.

#### (2) 기본패키지

① 저명도까지는 모든 패키지 중에서 java.lang 패키지는 Import 안해도 사용가능.

## 32강 배열 (Array)

### 1. 배열의 생성

#### (1) 배열의 생성

① `자료형 [] 배열명 = new 자료형[개수];`  
`자료형 배열명 [] = new 자료형[개수];`

② [] 위치는 상관이 없으나 공간에 위치한 걸 주로 사용

### 2. 배열의 초기화

(1) 배열은 선언과 동시에 초기화 가능

(2) 배열에 붙어주는 데이터 만큼 자동으로 배열 생성.

(3) `int [] array = {1, 2, 3, 4, 5};`

### 3. 배열의 접근

(1) 만들어진 배열의 순서대로 인덱스 번호 붙는다.

(2) [] 안의 숫자 : 첨자

## 34강 args의 활용

Date

Page

### 1. args

- (1) 프로그램을 실행할 때 사용자가 프로그램으로 직접 값을 넘겨줄 때 사용.
- (2) Java 클래스명 값이 값2 값3
- (3) `public static void main (String [] args)`
- (4) 값의 개수에 따라 args가 개수에 맞게 배열 생성된다.

## 35강 생성자 (constructor)

### 1. 생성자

#### (1) 생성자

운영체제가 호출함.

- ① Java class에서 객체를 생성할 때 자동으로 호출되는 메서드
- ② return type이 없음
- ③ 메서드의 이름 = 클래스의 이름
- ④ 객체를 생성할 때 반드시 수행되어야 하는 코드가 있거나 초기화의 작업을 하는데 사용됨

#### (2) 사용방식

```
⇒ public class명 (매개변수) {  
    코드  
}
```

### 2. 생성자의 overloading

#### (1) default 생성자

- ① 생성자를 class에 만들지 않으면 매개변수가 없는 생성자가 자동으로 만들어지며 내부에는 소스코드가 없다.
- ② 이러한 매개변수가 없는 생성자를 default 생성자라고 한다.

#### (2) 매개변수가 있는 생성자

- ① 매개변수의 형태를 다르게 하여 여러개의 생성자 생성 가능
- ② 객체 생성 시 선택 가능. ← 생성자의 overloading

## 36강 상속 (Inheritance)

### 1. 상속

#### (1) 상속

- ① 클래스를 만들 때 기존 클래스가 가지고 있는 메서드 혹은 변수를 가져다서 사용하는 방법 중 하나
- ② 참조는 질의, 매서드 변수를 물려받아 자기 자신 것처럼 사용하는 개념.
- ③ 기존에 만든 클래스에 새로운 기능을 추가하거나 이미 만들어진 매서드를 다시 구현 할 수 있음
- ④ 기존의 클래스를 직접 손대지 않고 추가 혹은 수정 가능  
→ 객체지향프로그래밍에서 많이 사용.
- ⑤ 다중상속 지원 X. 단일상속 지원 O
- ⑥ 개발자가 만든 상속은 ~~없음~~ → object라는 클래스 자동상속

#### (2) 부모와 자식 (객체 지향 언어)

- ① 부모 클래스 = 물려주는 쪽
- ② 자식 클래스 = 물려받는 쪽

### 2. 상속 방법

#### (1) extends

⇒ class 클래스이름 extends 부모클래스



## 3월 강 다형성 (Polymorphism) (상속2)

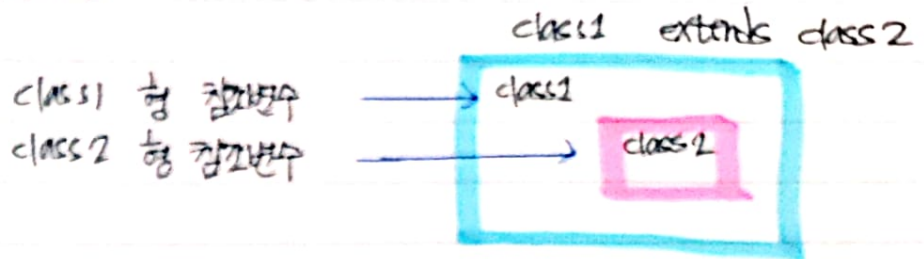
Date

Page

### 1. 다형성

#### (1) 참조변수를 통한 객체 참조

→ 클래스로 부터 객체를 생성하고 생성된 객체의 주소를 참조 변수에 넣을 때 참조변수의 타입에 따라 생성된 객체에 접근할 수 있는 영역이 달라진다.



### 2. 메서드의 overriding

- (1) 부모클래스가 가지고 있는 메서드를 자식클래스에서 다시 구현한 것을 가리킴
- (2) 메서드를 overriding을 한 부모클래스의 메서드를 호출하고자 할 때는 super를 이용한다.

#### (3) 참조변수의 활용

→ 부모클래스형 참조변수를 통해 객체에 접근하면 부모클래스의 영역에 접근하지만 호출하려는 method가 overriding 된 경우에는 자식클래스에 재 정의한 메서드가 호출된다.

