dongguk UNIVERSITY

객체지향프로그래밍

실습강의 6주차

실습강의 소개

- **실습 진행 방법**
  - 간단한 이론 복습 및 해당주차 실습과제 설명
  - 실습 후 보고서와 소스코드를 압축하여 일요일 자정(23:59)까지
    꼭!! 이클래스 제출(**이메일 제출 불가, 반드시 이클래스를 통해 제출**)
  - 실습 과제 제출 기한 엄수(제출 기한 이후로는 0점 처리)
  - **보고서는 출력하여 수업 시작 전에 제출(대면으로 실습 수업 진행 시)**
- **Q & A**
  - **이클래스 및 실습조교 이메일을 통해 질의 응답**
  - **이메일 제목 : [객체지향프로그래밍_홍길동] *본인 과목명과 성명 꼭 작성!**
  - **실습조교 메일 주소 : jade1087@dgu.ac.kr**

실습강의 소개

- **실습 보고서**
  - 문제 분석, 프로그램 설계 및 알고리즘, 소스코드 및 주석, 결과 및 결과 분석, 소감
- **제출 방법**
  - 보고서, 소스코드, 실행파일을 1개의 파일로 압축하여 e-class "과제" 메뉴를 통해 제출
    - **"이름학번실습주차.zip" 형태로 제출(e.g. :김동국19919876실습7.zip)**
    - 파일명에 공백, 특수 문자 등 사용 금지
  - 대면 수업일 경우 출력한 보고서를 실습 시간에 제출
- **유의 사항**
  - 보고서의 표지에는 학과, 학번, 이름, 담당 교수님, 제출일자 반드시 작성
  - 정해진 기한내 제출
    - 기한 넘기면 0점 처리
    - e-class가 과제 제출 마지막 날 오류로 동작하지 않을 수 있으므로, 최소 1~2일 전에 제출
    - **당일 e-class 오류로 인한 미제출은 불인정**
  - 소스코드, 보고서를 자신이 작성하지 않은 경우 **실습 전체 점수 0점 처리**
  - **Eclipse, 동국대학교 Shastra**를 사용하여 실습 진행

보고서 작성 방법

- **보고서 양식**
  - 문제 분석: 실습 문제에 대한 요구 사항 파악, 해결 방법 등 기술
  - 프로그램 설계 및 알고리즘
    - 해결 방법에 따라 프로그램 설계 및 알고리즘 등 기술
    - e.g.) 문제 해결 과정 및 핵심 알고리즘 기술
  - 소스코드 및 주석
    - 소스코드와 그에 해당하는 주석 첨부
    - 각각의 함수가 수행하는 작업, 매개변수, 반환 값 등을 명시
    - **소스코드 전체 첨부(소스코드 화면 캡처X, 소스코드는 복사/붙여넣기로 첨부)**
  - 결과 및 결과 분석
    - 결과 화면을 캡처하여 첨부, 해당 결과가 도출된 이유와 타당성 분석
  - 소감
    - 실습 문제를 통해 습득할 수 있었던 지식, 느낀 점 등을 기술

예외 처리

- **예외(Exception)**
  - 실행 중 발생하는 에러는 컴파일러가 알 수 없음
  - 자바에서는 실행 중 발생하는 에러를 예외로 처리
    - 응용 프로그램에서 예외를 처리하지 않으면, 예외가 발생한 경우 프로그램은 강제 종료됨

```
1.  import java.util.Scanner;
2.  public class ExceptionExample1 {
3.      public static void main (String[] args) {
4.          Scanner rd = new Scanner(System.in);
5.          int divisor = 0;
6.          int dividend = 0;
7.
8.          System.out.print("나눔수를 입력하시오:");
9.          dividend = rd.nextInt();
10.         System.out.print("나눗수를 입력하시오:");
11.         divisor = rd.nextInt();
12.         System.out.println(dividend+"를 "+divisor+"로 나누면 몫은 "+dividend/divisor+"입니다.");
13.     }
14. }
```

divisor가 0이므로 예외 발생

```
나눔수를 입력하시오:100
나눗수를 입력하시오:0
Exception in thread "main" java.lang.ArithmeticException: / by zero at
ExceptionExample1.main(ExceptionExample1.java:12)
```

예외 처리

- **예외 처리문**
  - try-catch-finally문 사용
  - finally 블록은 생략 가능

생략 가능

```
try {
        예외가 발생할 가능성이 있는 실행문(try 블록)
}
catch (처리할 예외 타입 선언) {
        예외 처리문(catch 블록)
}
finally {
        예외 발생 여부와 상관없이 무조건 실행되는 문장(finally 블록)
}
```

예외 처리

- try블록에서 예외가 발생하지 않은 정상적인 경우

```
try {
   ....
   실행문
   ....
}
catch (처리할 예외 타입 선언) {
   예외 처리문
}
finally {
   finally 블록 문
}
```

- try블록에서 예외가 발생한 경우

```
try {
   에외발생
   실행문
   ....
}
catch (처리할 예외 타입 선언) {
   예외 처리문
}
finally {
   finally 블록 문
}
```

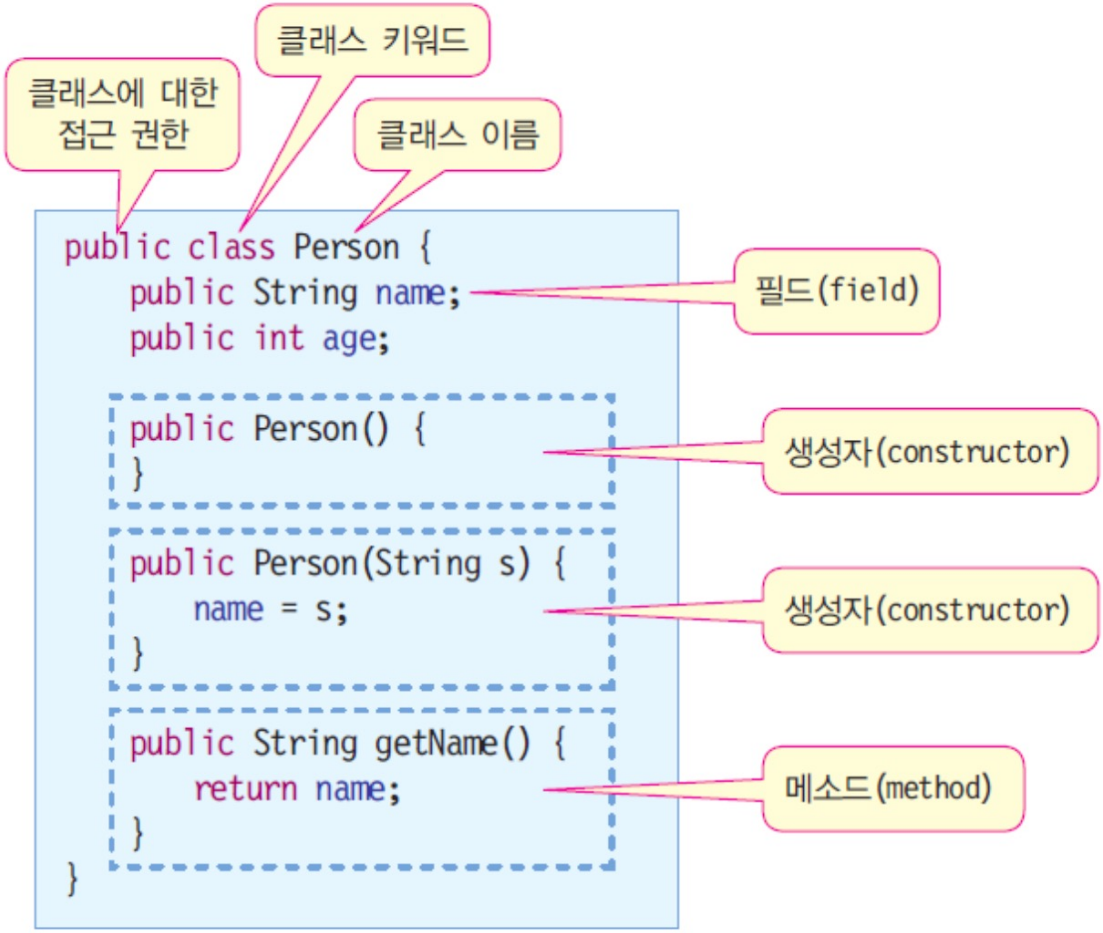
예외 처리

예외 종류	예외 발생 경우
ArithmeticException	정수를 0으로 나눌 때 발생
NullPointerException	null 레퍼런스를 참조할 때 발생
ClassCastException	변환할 수 없는 타입으로 객체를 변환할 때 발생
OutOfMemoryError	메모리가 부족한 경우 발생
ArrayIndexOutOfBoundsException	배열의 범위를 벗어난 접근 시 발생
IllegalArgumentException	잘못된 인자 전달 시 발생
IOException	입출력 동작 실패 또는 인터럽트 시 발생
NumberFormatException	문자열이 나타내는 숫자와 일치하지 않는 타입의 숫자로 변환 시 발생

클래스와 객체

- **클래스(Class)**
  - 객체의 속성과 행위 선언
  - 객체의 설계도 혹은 틀
- **객체(Object, Instance)**
  - 클래스의 틀로 찍어낸 실체
    - 메모리 공간을 갖는 구체적인 실체
    - 클래스를 구체화한 객체를 인스턴스(instance)라고 부름
    - 객체와 인스턴스는 같은 뜻으로 사용
- **사례**
  - 클래스: 소나타 자동차, 객체: 출고된 실제 소나타 100대
  - 클래스: 벽시계, 객체: 우리집 벽에 걸린 벽시계들
  - 클래스: 책상, 객체: 우리가 사용중인 실제 책상들

클래스와 객체



클래스와 객체

- **클래스 접근 권한, public**
  - 다른 클래스들에서 이 클래스를 사용하거나 접근할 수 있음을 선언
- **Class Circle**
  - Circle이라는 이름의 클래스 선언
  - 클래스는 {로 시작하여 }로 닫으며 이곳에 모든 필드와 메소드 구현
- **필드(filed)**
  - 값을 저장할 멤버 변수
    - 멤버 변수 혹은 필드라고 함
  - 필드의 접근 지정자 public
    - 필드를 다른 클래스의 메소드에서 접근할 수 있도록 공개한다는 의미
- **생성자(constructor)**
  - 클래스의 이름과 동일한 메소드
  - 클래스의 객체가 생성될 때만 호출되는 메소드
- **메소드(method)**
  - 메소드는 함수이며 객체의 행위를 구현
  - 메소드의 접근 지정자 public
    - 메소드를 다른 클래스의 메소드에서 호출할 수 있도록 공개한다는 의미

[연습 1] 클래스와 객체 사용

- Circle 클래스와 myCircle 객체 만들기를 따라 작성하고 실행하시오

```
1. class Circle {
2.   int radius;
3.   void set(int r) { radius = r;}
4.   double getArea(){
5.     return 3.14*radius*radus;
6.   }
7.  }
```

```
1. class Main {
2.   public static void main(String[] args) {
3.     Circle myCircle = new Circle();
4.     myCircle.set(3);
5.     System.out.println(myCircle.getArea());
6.   }
7. }
```

```
28.26
```

[연습 2] 클래스와 객체 사용

- Rectangle 클래스와 myRectangle 객체 만들기를 따라 작성하고 실행하시오

```
1. class Rectangle {
2.   int width;
3.   int height;
4.   void setWidth(int w) { width = w;}
5.   void setheight(int h) { height = h;}
6.   int getArea(){
7.     return width*height;
8.   }
9.  }
```

```
1. class Main {
2.   public static void main(String[] args) {
3.     Circle myRectangle = new Rectangle();
4.     myRectangle.setWidth(5);
5.     myRectangle.setheight(7);
6.     System.out.println(myRectangle.getArea());
7.   }
8. }
```

```
35
```

[예제 6-1] 상품 하나를 표현하는 클래스 Goods 만들기

- 상품 하나를 표현하는 클래스 **Goods**를 작성하라. 상품은 **String** 타입의 **name**, **int** 타입의 **price**, **numberOfStock**, **sold** 등 네 개의 필드를 갖는다. **Goods** 클래스 내에 **main()** 메소드를 작성하여 **Goods** 객체를 하나 생성하고 이 객 체에 대한 레퍼런스 변수 명을 **camera**로 하라. 그리고 나서 **camera**의 상품 이름(**name** 필드)을 "**Nikon**", 값(**pr ice**) 을 **400000**, 재고수(**numberOfStock**)를 **30**, 팔린 개수(**sold**)를 **50**으로 설정하라. 그리고 설정된 이들 값을 화 면에 출력하라.

```
1. public class Goods {
2.     String name;
3.     int price;
4.     int numberOfStock;
5.     int sold;
6.
7.     public static void main(String[] args) {
8.         Goods camera = new Goods();
9.
10.        camera.name = "Nikon";
11.        camera.price = 400000;
12.        camera.numberOfStock = 30;
13.        camera.sold = 50;
14.
15.        System.out.println("상품 이름:" + camera.name);
16.        System.out.println("상품 가격:" + camera.price);
17.        System.out.println("재고 수량:" + camera.numberOfStock);
18.        System.out.println("팔린 수량:" + camera.sold);
19.    }
20.}
```

```
상품 이름:Nikon
상품 가격:400000
재고 수량:30  팔
린 수량:50
```

[예제 6-2] 지수 클래스 MyExp 만들기

- 클래스 **MyExp**를 작성하라. **MyExp**는 지수값을 표현하는 클래스로서 두 개의 정수형 멤버 필드 **base**와 **exp**를 가진 다. **23**의 경우 **base**는 **2**이며, **exp**는 **3**이다. **base**와 **exp**는 양의 정수만을 가지는 것으로 가정한다. 또한 **MyExp**는 정수값을 리턴하는 **getValue()**라는 메소드를 제공한다. **getValue()**는 **base**와 **exp** 값으로부터 지수를 계산하여 정 수 값으로 리턴한다. 예를 들어 **MyExp**객체의 **base** 필드가 **2**이고 **exp**가 **3**이라면 **getValue()**는 **8**을 리턴한다.

```
1. public class MyExp {
2.     int base;
3.     int exp;
4.     int getValue() {
5.         int res=1;
6.         for(int i=0; i<exp; i++)
7.             res = res * base;
8.         return res;
9.     }
10.    public static void main(String[] args) {
11.        MyExp number1 = new MyExp(); number1
12.        .base = 2;
13.        number1.exp = 3;
14.
15.        MyExp number2 = new MyExp();
16.        number2.base = 3;
17.        number2.exp = 4;
18.
19.        System.out.println("2의 3승 = " + number1.getValue());
20.        System.out.println("3의 4승 = " + number2.getValue());
21.    }
22.}
```

```
2의 3승 = 8
3의 4승 = 81
```

실습과제 안내 [1/2]

- **실습과제 1**
  - "q"가 입력되기 전까지 정수 값을 입력받아 54를 더하는 프로그램을 작성하시오.
    (단, 숫자가 아닌 문자가 입력될 경우는 예외 처리할 것)

- **처리조건**
  - 예외처리(try ~ catch)
  - 반복문
  - 조건문
  - 입출력

- **실행결과**
  - 숫자 입력 종료(q):9
  - 숫자 + 54 : 63.0
  - 숫자 입력 종료(q):A
  - 문자가 아닌 숫자 입력할 것.
  - 숫자 입력 종료(q):q
  - main 메소드 종료

실습과제 안내 [2/2]

- **실습과제 2**
  - 밑변과 높이 필드를 가지는 삼각형 클래스를 작성하고, 두 삼각형의 밑변과 높이를 입력 받아 넓이를 비교하시오.

- **처리조건**
  - 기본 생성자
  - 조건문
  - 입출력

- **실행결과**
  - 삼각형1 밑변: 5
  - 삼각형1 높이: 7
  - 삼각형2 밑변: 8
  - 삼각형2 높이: 12
  - 삼각형2가 더 넓습니다!