

2024. 04. 07

선언부 / 구현부

선언부

접근 제한자
private
public
→ 주로 사용

빈 객체
void
< object

이름
동사로 작성

파라미터

개수 제한 X.
순서대로 들어감
변수 범위 (지역변수)

↑
구현부 작성

구현부

↗
Σ 3

<선언부에서 중요한 것>

1. 네이밍

↳ 메서드 : 동사

↳ 변수 : 형식

parameters (파라미터) = 변수

(메서드 분리 하기 위해 선언부 중요)

파라미터 / 오버로드

파라미터
↓
개수
↓
타입

```
public static void main (String .. args) {  
    SimpleDateFormat sdf = new SimpleDateFormat  
    println(sdf)  
}  
  
public static void println (SimpleDateFormat sdf) {
```

3

주소값 참조나 값 참조 : 설명의 예를 들

call by Reference

call by Value

ex) 체크 카드

ex) 돈 (현금)

계좌 정보 (주소)

메모리 영역

클래스에는 게티터, 셋터 붙여 있다

클래스, 인스턴스 → 클래스가 있어야 인스턴스 만들 수 있다

↳ 인스턴스라 할 수 있다

↓

ex) math

가산기 가 = new 가산기();

인스턴스라

3

< 클래스는 객체로 존재 가능 (메모리 등해가되지 X)

클래스 ≠ 인스턴스

new 하는 순간에 메모리 할당이 되는 = 인스턴스라

(스테틱) 영역 - static 스테틱 영역의 특징 (리소스/

(스테틱) " - 메소드의 특징

↳ 공유 : 계산 결과 등 (한번)

전역변수 / 후위 / 중위

94x527
4+3
43+

43

메시지는 처리하는 부분을 가지고 스태틱.
(구현 구조는 스태틱 클래스로 되어 있음)

(하) 명칭 : 객체로서 놓이기

인스턴스화 할 것

상급은 레전 사용이 : 같은 개념을 객체 합남바 - 메모리 부족해함
↑
객체 여러개 생성
↑
객체 관리 쓰기

상속 : 왜? 코드를 재사용을 위한 부분을 나누기
↓
관계에 의해 나옴

is-a / has-a 포함
상속

is-a

ex) 사냥 extends 동물

(사냥은 동물이야)

(사냥은 is 동물) - 직접적인 상관관계가 없어야 함

확장해 쓰려고 하는 애의 관계 생각

⊕

코드의 재사용

has-a 포함

사냥은 도구이다 (X)

사냥을 도구는 가지고 있다 (O)

ex) class 사냥 {

```

class Animal {
    호기 tool;
}

```

코드로 표현할 수 있는가?

```

class 동물 {
    먹기()
    자기()
    abstract 싸기()
}

```

```

class 시냇물 extends 동물 {
    호기 tool; // has - a
}

```

먹기() ; ← 생략 가능 → 먹는게 다는건

해리"고

바뀔 수 있는 것

추상화

(예)

행위인 움직이기

구현, 파생되어 다는 때

(해위늘 갖추기 위함)

@Override

← 상속 관계에서 재구현 ≠ 추상화하는 구현

먹기() {

← 이건 꼭 같아야 한다 "구현"

, tool.구현();

구현하는 의미(구현)

나를 통해 돈을 주는 것

제어 private 를 준다

private 지갑 Wallet;

↳ 지갑 가져와? 주소 알려주지 X

setter & getter 는 중요

대신 받은 (내가 가진 것)

넣기

setter ← 돈 넣기

가져내기

getter ← 돈 빼기

private, getter, setter

캡슐화

↳ 접근 가능하게 속성기 내는

값을 주고 받는다

↓

여기 getter & setter 는 중요

캡슐화는 변수 이름, (메서드는 X)

↓
가변이기에

↓
(캡슐화 때문)

연산용이기에

인터페이스 interface

상수: (상속 매서드 선언, 구현 리기함)
(특정 리턴 타입)

작업 지시서

interface 퇴근() {

3 집에 가();

추상매서드

↓

구현 방법 없음

코드를 작성하지 않음

컴파일 타임에

알려 주리.

인터페이스 등록

(매서드 전부 추상
구현 리기함)

구현부 없음

↓

구현 안해서

↓

컴파일 타임에

① Override 불요구 구현 (이유: 선언부 같기때)

다행

구현부도 생략 가능

interface 자국어 {

자국어();

}

interface 음향어 {

자국어();

}

class 고양이 implements 자국어 {

④ Overriding

자국어() {

print("야옹");

}

class 개 implements 자국어 {

④ Overriding

자국어() {

print("멍멍");

}

class 사람 implements 자국어, 음향어 {

④ Over-

자국어() {

print("ㅋㅋㅋ");

}

④ Override

자국어() {

print("스피치 ㅋㅋ");

}

코자

인터페이스 상속

수호전 수호

(보/가/다/오/리)

인터페이스: 계층적 상속 (수호전 수호)

상속: 관계적 상속

다형성: 상속의 is a 관계에 따른

class 예시 {

사람 human = new 동물();

// 동물은 사람이다 (X) (오류)

동물 animal = new 사람();

// 사람은 동물이다 (O)

↑

animal.
 먹인() 동종종교나 알이 동종이 죽어버리면
 죽은()
 새인()

Object obj = new 새인(); 동종은 동종
 관계는 코드로 표시할 수 있다

A instance of B → A가 B가 맞나?
 X
 Object와 instance of 사용. (관계는 동종종교)
 interface는 ~~사용~~
 인터페이스

instance of → 네가 받은 예제의 instance of  해서 찾는다.

스프링

· 애플리케이션은 스프링 보단 커널로

쓰는 것을 권장함

· 테스트 코드 1백프로 자동화

Controller나 다른 rest Controller 하일때

1회 한 예제

신입기준 (관심있는 분야)

↳ 웹: 컴퓨션 (자바로 처리)

SI기준 spring 2개짜 사용

Spring web

API } → 이걸 알아야 됨

← 이걸 통해 테스트

소문자 12

Swagger

메거. API 테스트, 스킴 ?? 정리해 보여줄

연장 : 자기소개서 기본

전반적으로 작성해라.

(짧은게는 적어라.

(물어볼지?
위에서 완성되나?
어떻게 써나갔나

~~성실하게 X~~

~~성실하게 X~~

★ (오늘은 제발이 관계하지
가능해라)

프로그래밍 관습의 법

가위 바니 보 구현

↳ 입력 받기 : 2개, 1개당 3개의 정수

출력 : 승 / 차 / 비

비교 : 네 소문자

1번 : 상수에게도 입력받을

2번 : 입력받은 값은 변수에 저장

2-1) 가위 = 1, 바니 = 3, 승 = 5

3번 : PC의 값을 하나로 선택해 해볼

3-1) 값은 배열에 써서 저장해 줄것

인도

4번; 2번과 3번의 값은 0으로

4-1) 4번의 값은 등/제/비값으로 정리

5번: 정리는 아니지만

→ 34번은 정렬하기 안고, 양쪽은 중복 생각

이런 비특하게 자바 약하기

조건: S/B/O

사용자에게 1~35사이의 값을 받음

PC도 숫자 / 23 중 1개 값을 선택

숫자 0으로 - 같은지 S

- 다른지 B

- 같은 숫자 3번 내기 21번 0 증가

이제

java 11로

spring 5.2에 맞춰

메모리 16G 이상 권장

Java, JS, MySQL, GitHub, REST, API.

JWT.

개발자

컨테이너 또는 dev /

openjdk → 8 또는 11

tomcat.apache.org → 9.0.87

maven.apache.org → 3.4.6

스프링
3

1 ~ 10 분 회선 4개

코드로 표현

(5번은 못함)

코어 : 책보리유기