

모든 단축키 보기 **Ctrl** + **Shift** + **L**

자동완성 **Ctrl** + **Space Bar**

선언된 변수확인 **F3**

메소드 리스트 보기 **Ctrl** + **O**

사용된 모든 패키지를 자동으로 import
Ctrl + **Shift** + **O**

클래스, 메소드, 또는 멤버를 검색 **Ctrl** +
마우스커서(또는 **F3**)

abstract

추상 클래스 또는 추상 메소드에 사용함

boolean

논리형 참(true) 과 거짓(false)
default 값은 false

break

반복문 또는 Switch 문을 벗어나거나 멈출때 사용함

byte

데이터 정수형 타입 1byte = 8 bit

정수형: byte <short <int <long

case

switch ~ case ~ default 를 많이 봤을것이다.

switch 조건문에서 사용함.

case 문에서는 리터럴과 상수만 허용

변수는 안됩니다~

break 를 만나면 switch 문을 빠져나옴

catch

try~catch~finally문에서 사용됨

예외처리 할때 사용됩니다.

catch(){}

() 안에는 예외에 해당하는 인스턴스를 집어넣음

ex) catch(예외클래스 변수) // catch(Exception e)

{ } 안에는

예외 처리 내용을 적어넣음

예외처리 순서

try 문 -> 예외발생 -> catch -> 예외확인/에러처리 -> finally 예외발생 유무상관없이 실행

char

데이터 문자형 (변수) 타입 2byte

ex) 'A' 로 입력, 하나의 문자만 저장할수있음

class

모든 코드는 클래스 안에 존재해야함

클래스 작성방법은 매우 간단함

class 클래스명{

클래스 이름 첫글자는 '대문자'로 시작하는것이 좋음

ex) Pyh class

continue

반복문, 루프돌때 현재 루프를 끝낸뒤,

뒤에 내용 무시하고
앞으로가서 다음 루프를 실행함

default

접근제어자 및 switch 문 에서 쓰임
접근제어자는 클래스, 메소드, 멤버변수 앞에 아무것도 선언을 안하면 default 가 됨.
switch 문에서 default 는 case 문을 돌다가
해당하는 case 가 없으면 default 로 와서 실행함.

do

do~while 문에서 쓰임
하지만 잘 쓰이지 않는 do~while 문
do { 반복 실행 내용 } while (조건문)

double

데이터 실수형 (변수) 타입 8byte
실수값을 넣을때 쓰임

else

'사전적내용 - 또 다른 , 다른'
if~else 문에 쓰임
총 3가지 형식이 있음

1. if(조건) {실행내용}
2. if(조건) {실행내용} else {실행내용}
3. if(조건){실행내용} else if(조건){실행내용} else if(조건){실행내용} else{실행내용}

extends

- 클래스 상속에 쓰임
상속은 2가지 형태가 있음
1. extends - 부모의 모든권한을 가져오는것 (상속+내용까지모두)
클래스 한개만 상속가능.
 2. implements - 인터페이스 (상속은 하지만, 내용은 비어있음, 추상)
다중 상속가능.

false

사전적의미 - 틀린, 사실이아닌

boolean 에서도 쓰임
그외 어디 쓰이는거 같은데 잘모르겠음.

finally

try~catch~finally 예외처리에 쓰임
switch 문의 default 처럼 쓰임
try { 예외가 발생할만한 코드 }
catch(Exception e){ 예외처리 방법 }
예외가 발생할만한 코드를 try {} 안에 넣고
예외가 발생하면 실행을 멈추고 catch{}안의 내용을 실행함
finally{ try~catch 끝난후 처리할내용 }
예외처리 순서
try 문 -> 예외발생 -> catch -> 예외확인/에러처리 -> finally 예외발생 유무상관없이 실행

float

데이터 실수형 (변수) 타입 4byte
실수값 넣을때 사용됨

for

반복문 중 하나
for(초기값;조건식;증가값){내용}
ex) for(int i = 0 ; i <= 9; i++)
{
System.out.println(i);
}
0부터9까지 화면에 뿌려줌

if

일반적으로 사용되는 조건문
1. if(조건) {실행내용}
2. if(조건) {실행내용} else {실행내용}
3. if(조건){실행내용} else if(조건){실행내용} else if(조건){실행내용} else{실행내용}

implements

인터페이스 구현 할때 쓰임, 다중상속 가능
interface 는 메소드 선언만 되어있고

빈 껍데기 (내용이없음)

import

import 패키지명.클래스명;
import 패키지명.*; 을 사용하면
그 패키지 내에 모든 클래스를 가져올수있음.
java api, 외부클래스를 사용하고싶을때 쓰입니다.
단축키 : ctrl + shift + o

instanceof

객체타입 검사시 사용됨
형변환이 가능한지 알고싶을때
true / false로 답을 얻을수있음

int

데이터 정수형 4byte
제일 많이 쓰이는 기본 정수형이 아닐까 싶습니다.
정수형: byte <short <int <long

interface

일종의 추상클래스, 상속을 목적으로 만들어지며 다중상속가능
상수와 추상메소드만 가짐 (내용선언안함)
interface 인터페이스이름 { public static final 타입 이름 = 값 ;
public abstract 메소드이름(매개변수); }
상속은 implements 로 한다.

long

데이터 정수형 8byte
정수값을 저장하는 변수타입
정수형: byte <short <int <long

native

자바 외 다른 프로그래밍 언어로 구현되었음을 알려줌

new

인스턴스,객체를 만들때 사용
객체를 생성하여 메모리공간을 만듦

null

ex) Box box1 = null

변수선언 하고 box1 의 값은 null로 초기화 시킨것

메모리에 할당 안된상태

숫자타입의 변수는 null을 넣을수없음

package

클래스들을 편리하게 관리하기 위해 묶어놓은것 (하나의 폴더라고 생각)

package 작성법: package 패키지명;

import 방법: import 패키지명.클래스명;

private

접근제한자 중 가장 높은보안수준(?)

같은 클래스 내에서만 접근 가능한것

순서: public >protected >default (아무것도 안적음) >private

protected

접근제한자 이며, 같은패키지와 상속받은 클래스에서 접근이 가능한것

순서: public >protected >default (아무것도 안적음) >private

public

접근제한자 중 가장 제한이 없는것

접근제한이 없기때문에 같은클래스,패키지,상속에 상관없이

누구나 이용할수있다.

return

메소드를 종료하거나 값을 반환할때 쓰임

short

데이터 정수형 타입 2byte

정수값을 저장하는데 사용됨

정수형: byte <short <int <long

static

정적변수 혹은 클래스변수 라고 부르며

클래스 변수란 클래스 단위로 존재하며
클래스로부터 만들어진 모든 객체가 공유할수있는 변수를 의미
한번만 생성되고, 컴파일할때 만들기때문에 객체생성(인스턴스)이 불필요하다.
static 은 main함수 보다 먼저 실행된다 (단 한번 실행함)
static 에서는 this 와 super 를 사용할 수 없다.

super

상위 클래스의 생성자를 호출할때 또는
상위 클래스의 멤버변수 또는 메소드에 접근할때 사용됨.
super 로 호출하기 위해선 해당 멤버변수,메소드 가 protected,public 이어야함.
호출방법

1. 상위 클래스의 생성자 호출 super(매개변수)
2. 상위 클래스의 멤버 변수 호출 super.변수명
3. 상위 클래스의 메소드 호출 super.메소드명(매개변수)

switch

조건문 중 하나
조건이 수가 많을때는 if문 보다 switch문을 사용하는것이 더 좋다.

```
switch(조건식) {  
    case 값1 : 내용    break;  
    case 값2 : 내용    break;  
    case 값3 : 내용    break;  
    case 값4 : 내용    break;  
    default : 내용  
}
```

synchronized

쓰레드 동기화 를 위해 존재함
쓰레드2가 방(synchronized) 메소드에 들어가기위해선
열쇠(lock)이 필요한데 이 열쇠(lock) 가 딱 하나밖에없다.
쓰레드1이 방(synchronized)에 열쇠(lock)를 들고 들어갔으면
일을 마치고 나와서 열쇠(lock) 를 반환할때까지
쓰레드2는 기다렸다가 쓰레드1이 방에서 일을 마치고 나오면
그 열쇠(lock) 를 받아서 방(synchronized)에 들어갈수가 있는것이다.

this

인스턴스가 '본인(자기자신)' 을 가르킬때 사용할 수 있다.

throw / throws

둘다 예외를 던질때 사용

throw 사전적의미 '던지다'

throw 와 throws 의 차이점

throw는 메소드내에서 상위 블럭 으로 예외를 던짐

throw 는 현재 메소드에서 상위 메소드로 예외를 던짐

throw 는 일부러 예외를 발생시킬때 사용되기도 하지만

현재 메소드 에러를 처리한후 상위 메소드에 에러정보를 줌으로써

상위 메소드에서도 에러를 감지할수있음

true

사전적의미 - 사실,참,진실

boolean 에서도 쓰임

그외 어디 쓰이는지 잘모르겠어요.

try

try~catch~finally문에서 사용됨

예외처리 할때 사용됩니다.

try { 예외가 발생할만한 코드 }

catch(Exception e){ 예외처리 방법 }

예외가 발생할만한 코드를 try {} 안에 넣고

예외가 발생하면 실행을 멈추고 catch{}안의 내용을 실행함

finally{ try~catch 끝난후 처리할내용 }

예외처리 순서

try 문 -> 예외발생 -> catch -> 예외확인/에러처리 -> finally 예외발생 유무상관없이 실행

void

아무것도 리턴할것이 없다.

void 에 속한 메소드는 리턴할게 없어서

처리를 끝낸 후 메소드를 호출한곳으로 넘어가 제어를 함.

while

반복문 for문과 while문은 항상 서로 변환이 가능함.

while(조건식) { 조건식이 맞으면 수행할 내용 }

*주의점 초기값 위치는 while문 밖에해야한다.

while 문 내에 하면 무한루프에 빠진다.