IP 203.236.209.135

공유폴더 file

작업폴더 JavaBasic

주말반으로 부팅하세요.

**1주차 - 자바 개요 및 자바 기초문법**

1교시 09:30~10:30 자바의 개념

2교시 10:30~11:30 jdk와 이클립스 다운로드 설치, 자바개발순서

3교시 11:30~12:30 식별자, 예약어, 변수, 상수, 문자열

4교시 12:30~13:20 데이터타입, 형변환

점심시간 1:20~2:30분

5교시 14:30~15:30 연산자

6교시 15:30~16:30 제어문

7교시 16:30~17:30 제어문

8교시 17:30~18:20 배열

===============================================================================

1교시 (9:30~10:30) 자바의 개념 ===============================================================================

자바의 특징

배우기쉽다

플랫폼 독립성

객체지향프로그래밍(OOP)

가비지컬렉션

멀티스레드

자바플랫폼

API - 클래스와 인터페이스들이 묶여져있는 패키지

JVM 자바가상머신 - 명령어와 레지스터의 집합, 스택, 힙, 메서드 영역으로 구성

자바 플랫폼의 종류

SE 일반 데스크탑 PC

EE 서버용 컴퓨터

ME 모바일

===============================================================================

2교시 (10:30~11:30) jdk, 이클립스 설치, api즐겨찾기, 자바개발순서 ===============================================================================

jdk설치

[www.oracle.com](http://www.oracle.com)

JDK(Java Development Kit)자바 개발도구

JDK=JVM+JRE+개발을 위해 필요한 도구(javac, java 명령어)

1. JVM(Java Virtual Machine)자바 가상머신

플랫폼(OS)에 의존적입니다.

2. JRE(Java Runtime Environment)자바 실행환경

자바 프로그램을 동작시킬 때 필요한 라이브러리 파일을 가지고 있습니다.

\*jdk설치시 컴퓨터 운영체제 비트수 확인

환경변수설정

[http://www.java.com/ko/download/help/path.xml](https://www.java.com/ko/download/help/path.xml)

내컴퓨터 --> 고급설정 --> 환경변수 --> PATH

API 즐겨찾기

<http://docs.oracle.com/javase/8/docs/api/>

이클립스설치

<http://www.eclipse.org/>

IDE(Intergrated Development Environment)통합개발환경

자바프로그램 실행순서

코드를 작성=>코드 컴파일=>JVM으로 실행

컴파일이란?

자바 가상머신이 인식할 수 있도록 java코드를 기계어로 바꾸어 주는 것

===============================================================================

3교시 (11:30~12:30) 주석문, 식별자, 예약어, 변수, 상수, 문자열 ===============================================================================

작업폴더명 JavaBasic

프로젝트명 Java0806

패키지명 com.hb.java.week1

주석문

주석은 프로그램의 코드와 실행에 영향을 주지 않는 문장. (컴파일 대상에서 제외가 되는 것)

이클립스 단축키참고

<http://egloos.zum.com/littletrue/v/3987863>

식별자와 예약어

공통적으로 적용

- 공백안됨

- 첫 번째 글자는 문자 이거나 '$', '\_'이어야 함(숫자 안됨)

- 두번재 이후의 글자는 숫자, 문자, '$', '\_' 가능

- '$', '\_(underscore)' 이외의 특수문자 사용 불가능

($,\_는 자바 내장 클래스에서 사용되고 있으므로 충돌이 발생할 수 있으니 가급적 사용을 안하는 것이 좋음)

- 길이 제한이 없다.

- 상수 값을 표현하는 단어 true, false, null은 식별자로 사용할 수 없음

- 자바의 키워드는 식별자로 사용할 수 없다.

클래스 - 첫글자 대문자, 단어 이어질때 첫글자 대문자

상수 - 모두 대문자, 단어 이어질때 \_(underscore)

변수,메서드 - 소문자, 단어 이어질때 첫글자 대문자

예약어 - 자바에 이미 사용하고 있는 등록된 이름

변수, 상수, 문자열, 세미콜론

변수-변하는 값

상수 - 변하지 않는 값(타입 앞에 final 선언)

문자열-문자열 출력(내장클래스)

세미콜론은 문장의 끝맺음(사용 안할시 컴파일 에러가 발생한다. )

세미콜론을 사용시 여러개의 문장을 한줄에 표현가능

int a = 100; double b = 10.1;

===============================================================================

4교시 (12:30~13:20) 데이터타입, 형변환 ===============================================================================

데이터타입

1. 기본형타입

\*표현가능 범위에 따라 나누어 짐.

논리형 boolean(1bit) true, false

문자형 char(2byte) 문자 캐릭터

정수형

byte(1byte)

short(2byte)

int(4byte)

long(8byte)

실수형

float(4byte)

double(8byte)

2. 참조형타입

객체는 기본형 타입이 아닌 참조형타입

형변환

**byte < short, char < int < long < float < double**

프로모션-작은타입->큰타입(자동형변환, 자료손실없음)

디모션-큰타입->작은타입(형변환필요, 자료손실있음)

\*리터럴=상수

정수리터럴=50,-10,100

실수리터럴=10.2, 1.2

문자리터럴=A,B,\n

논리리터럴=true, false

===============================================================================

5교시 (14:30~15:30) 연산자 ===============================================================================

산술연산자

+, -, \*, /, %

대입연산자

단순 대입 연산자 =

복합 대입 연산자

e1+=e2; //e1=e1+e2

e1-=e2; //e1=e1-e2

e1\*=e2; //e1=e1\*e2

e1/=e2; //e1=e1/e2

e1%=e2; //e1=e1%e2

단항연산자

부호 +, -

증감 ++, --

비교연산자(관계연산자)

==같다

!=다르다

<, >, <=, >=

논리연산자

&& 둘다만족 and 논리곱

|| 하나만 만족 or 논리합

! 부정(not) not

연산의 우선순위

산술 > 비교 > 논리 > 대입

단항 > 이항 > 삼항

문자열비교

equal()

===============================================================================

6~7교시 (15:30~17:30) 제어문 ===============================================================================

if 조건문

조건문에는 true 또는 false값을 반환하는 것이라면 무엇이든 올 수 있습니다.

if(조건문){

}else if(조건문){

}else{

}

삼항연산자

(조건문)? true일때 실행 : false일때 실행

Switch문

int num=2;

switch(num){

case 1:

System.out.println("one");

break;// break가 없으면 다음 문장 실행

case 2:

System.out.println("two");

break;

default:

System.out.println("default");

break;

}

switch 문을 사용할 때 한가지 주의 할 것

switch의 조건으로는 몇가지 제한된 데이터 타입만을 사용할 수 있다는 것입니다.

=>byte, short, char, int, enum, String, Character, Byte, Short, Integer

7버전 이전에는 정수타입만 받을 수 있었지만 이후 버전부터는 위의 타입을 다 받을 수 있게 되었습니다.

While 반복문

반복문의 조건 : 반복의 횟수를 세기 위한 변수, 반복의 조건문, 조건을 끝내기위한 연산

while(조건){

반복 실행 영역

}

검사-실행-검사-실행

조건문이 거짓이 될 때까지 반복문안의 영역을 실행하게 됩니다.

거짓이 되는 경우를 실행해주어야 무한루프를 방지할 수 있습니다.

처음부터 조건문이 거짓일 경우는 실행되지 않습니다.

Ctrl+C(dnls 나 Cmd+. 단축키를 이용해서 무한 반복을 중지할 수 있습니다.

do-While 반복문

실행-검사-실행

조건이 만족하지 않더라도 무조건 한번은 실행되게 하는 반복문입니다.

do{

실행문;

}while(조건문);

for 반복문

for(초기화; 조건식(종료조건); 증감식){

반복적으로 실행될 구문

}

break, continue;

반복문에서 사용되는 키워드입니다.

break

break문을 만나면 이후는 실행되지 않고 반복문을 빠져나갑니다.

무한루프에서 break를 쓰면 좋습니다.

ex)6의 배수이면서14의 배수인 가장 작은 정수찾기

break레이블

둘이상의 반복문을 빠져나갈때 break레이블을 사용한다.

outerLoop:

for(){//1번

for(){2번

break outerLoop;//2번 반복문을 빠져나감

}

}

continue

실행하다가 continue를 만나면 그이후의 실행은 건너뛰고 다시 반복문으로 이동하여 계속 실행.

continue레이블

둘이상의 반복문으로 돌아갈때 사용

outerLoop:

for(){//1번

for(){2번

continue outerLoop;//1번 반복문으로 돌아감

}

}

중첩반복문

중첩반복문은 반복문 안에 반복문을 넣는 것입니다.(섞어서 사용가능)

=>for문안에 while

=>for문안에 for

=>while문안에 for

=>while문안에 while

===============================================================================

8교시 (17:30~18:20) 배열 ===============================================================================

배열은 연관된 정보를 그룹핑하는 것입니다.

같은 데이터 타입을 가진 연속된 메모리 공간으로 이루어진 자료구조입니다.

같은 데이터 타입을 가진 여러개의 변수가 필요할 때 사용합니다.

변수는 하나의 데이터를 저장하는 것이고 배열은 여러개의 데이터를 저장하는 것.

배열도 인스턴스이기 때문에 기본데이터 타입이 아닌 참조형 타입입니다.

배열은 크기를 지정해 주어야하고 한번 정하면 바꿀 수 없습니다.

1차원배열

**배열 사용하기**

배열로 사용하기 위해서는 변수처럼 선언을 먼저 해야합니다.

배열의 순서는 index의 개념으로 1번째가 아닌 0번째부터 시작합니다.

배열의 크기가 2라면 순서는 0번째, 1번째에 값을 넣을 수 있습니다.

String [] str=new String[2];//배열의 크기가 2이며 아직 값은 담지 않은 상태

str[0]=”a”;//index 0번째에 a을 담습니다.

str[1]=”b”;//index 1번째에 b를 담습니다.

String [] str={“a”, “b”};//배열의 크기가 2이며 값a, b를 담은 상태

str[0]; //배열의 값

str.length; //배열의 크기

이때 타입이 String인 배열은 문자열만 담아야 합니다.

그리고 배열은 크기를 정해주어야합니다. 정해주면 고정값이 되구요.

정해진 크기보다 값을 더 담게 되면 에러가 나게 됩니다.

값을 모자라게 담게 되면 남아있는 공간은 null로 표기됩니다.

**배열에 있는 값 순차적으로 출력하기**

그럼 배열에 들어있는 값들을 순차적으로 출력하려면 어떻게 해야할까요?

반복문을 이용해서 값들을 출력할 수 있습니다.

반복문을 사용하기 위해서 필요한 것은 변수를 선언하고 몇번 반복할 것인가?

1. 20명의 학생이 있을경우 배열에 20명의 학생을 담습니다.

String[] arr={“1”,....”20”};

2. 배열의 크기를 알아냅니다.

int count=arr.length;

3. 반복문을 작성해서 출력합니다.

for(int i=0; i<count; i++){

System.out.println(arr[i]);

}

for-each

배열은 반복문과 함께 빈번하게 사용을 합니다.

for문을 좀더 간편하게 사용하기 위해서 나타난것이 for-each문.

for-each문은 for문에서 반복하고 증가시키는 작업을 감추고 간략하게 표기한 것입니다.

String[] members = { "1", "2", "3" };

for (String e : members) {

System.out.println(e + "번 학생");

}

\*자바 5.0부터 도입된 기능이다.

2차원배열

1차원배열이 여러개 있는 것

int[][] array=new int[2][3];

첫번째 대괄호는 1차원배열의 갯수:행(세로로 2개)

두번째 대괄호는 각 1차원배열의 길이:열(가로로 3개)

2행 3열의 다차원 배열이 생성됨.

ㅁㅁㅁ

ㅁㅁㅁ

즉 2차원 배열은 1차원배열을 묶은 형태입니다.

public class Array {

public static void main(String[] args) {

int[][] arr=new int[2][3];

for(int i=0; i<arr.length; i++){//1차원배열의 갯수 2번 반복

for(int j=0; j<arr[i].length; j++){//각 1차원 배열의 길이만큼 반복(여기선 3번)

arr[i][j]=i+j;

}

for(int j=0; j<arr[i].length; j++){

System.out.print(arr[i][j]+" ");

}

System.out.println("");

}

}

}

가변적배열

//각 1차원 배열의 길이를 모두 3로 고정한 상태

int[][] array=new int[2][3];

int[0]={1,2,3};

int[1]={1,2,3};

ㅁㅁㅁ

ㅁㅁㅁ

//각 1차원 배열의 길이를 다르게 주고 싶을 때

int[][] array=new int[2][];//가변적

int[0]={1,2};

int[1]={1,2,3};

ㅁㅁ

ㅁㅁㅁ

다차원배열 선언과 동시에 값 넣는 방법

int[][] a={{1,2},{1,2}};

int[][] a={{1,2},{1,2,3}};