05/24

for 문

반복횟수를 알고 있을 때 주로 사용한다. 선언부, 조건식, 증강식 다 없어도

for문은 작동함, for(시작에 int 넣어야함 단위형을 설정)

1. 반복 횟수가 정해진 경우

2. 배열과 함께 주로 많이 사용

🡺 for ( 초기식; 조건식 ; 증감식 ) {

반복하고자 하는 문장들🡺 중괄호안을 실행하고 증감식을 실행후, 끝내고

} 조건식으로 다시돌아가서 false가 되면 종료

while 문

for문이 정해진 횟수만큼 반복한다면, while 문은 조건식이 true 일 경우에

계속해서 반복합니다. 조건식에는 비교 또는 논리 연산식이 주로오는데, 조건식이

false가 되면 반복 행위를 멈추고 while문을 종료합니다.

1. 무한 루프나 특정 조건에 만족할 때까지 반복해야하는 경우

2. 주로 파일을 읽고 쓰기에 많이 사용

🡺 While ()괄호안이 true면 {} 중괄호 안을 실행하고 나와서 다시 ()괄호 시작

**int** length = line.length();

int count = 0;

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

char c = line.charAt(i);

.length();

문자열의 길이를 구하는 함수

.charAt(i); 메소드

charAt 함수란?  
String 타입의 데이터(문자열)에서 특정 문자를 char 타입으로 변환할 때 사용하는 함수이다.

charAt(i)  
i 자리에는 int 형 변수를 넣어서 원하는 위치의 문자를 가져올 수 있다.

do-while 문

조건에 따라 반복을 계속할지를 결정할 때 사용하는 것은 while 문과 동일합니다.

다만, 무조건 중괄호 do{m럭을 한번은 실행하고, 조건을 검사하여 반복을 결정합니다.

// 무조건 한번은 실행하고' 그 후에 잘못 입력 했을 때 다시 한번 더 입력할 수 있는 기회를 줄 때

// do while문을 씀

Do {

If ()

System.out.println("제시한 정수가 높습니다.");

} while ();

문자열은 참조형 변수고 문자열을 비교 할땐 참조형

(aaa == bbb) 와 aaa.equal(bbb)차이??

==은 aaa 와 bbb의 참조형 변수의 주소값을 비교한다

Equal는 aaa 와 bbb의 값 그 자체를 비교한다.

break문

for문, while문, do-while문을 종료합니다.(반복을 취소)

switch문을 종료합니다.

대개 if 문과 같이 사용되어 if 문의 조건식에 따라서 for문과 while문을 종료할 때 사용합니다.

break문에 이름을 붙이면 그 이름을 붙인 반복

first :

~1

~2

~3

Break first; 🡺 ~2로 빠져나가는게 아니라 first로 빠져나감

Continue문

Continue를 만나면 증감식으로 돌아감

아래 코드를 생략하고 다음 반복을 실행한다.

continu문은 skip이라고 할 수 있다.

continu문 아래 코드들을 수행하지 않고 다음 조건으로 넘어간다.

05/24

들여쓰기 : Tap

내어쓰기 :shift+Tap

System.out.printf(AAA); 줄바꿈 할 때 그 다음

System.out.println(); 넣어주기

Switch case문

Switch 조건식에는 정수가 온다. String 객체 사용 가능(문자형)

이 정수값에 따라서 해당 case를 찾고 그 case와 그밑으로 쭈욱 실행 시킴

그래서 Break를 사용하면 if, else if 문처럼 사용 가능

switch (i) {

case 1:

System.out.println("일");

break;

case 2:

System.out.println("이");

break;

default:

System.out.println("일이삼 아님");

default는 switch에서 해당하는 정수값을 찾을수 없을 때 출력

inner loop

out loop

for (int j = 0; j < 5; j ++) {

for (int i = 0; i < 5; i ++){

System.out.print('\*');

}

}

05/25

동작하는 문장에 이름을 붙이는 것을 ‘메소드’ 라고 함

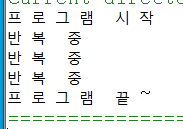
##메소드를 호출하는 방법##

public static void printRepeat() { // 메소드 머리(head)

// 메소드 몸통(body)

for(int i = 0; i < 3; i++) {

System.out.println("반복 중");

public static void main(String[] args) {

System.out.println("프로그램 시작");

// 메소드 호출

printRepeat();

System.out.println("프로그램 끝~");

}

// 메소드의 이름. 영소문자. 변수이름 지을때랑 똑같음. 두번 째 단어는 첫글자 대문자

// 메소드의 이름은 기능에 대한 이름을 담다 보니 주로 ##동사## 가 이름으로

// 메소드 호출은 printRepeat(); 메소드 이름과 ()로 불러온다.

##TestMethod1~3## 보기

Public : ‘접근제어자’

Static : ‘정적 메소드’ 그냥 존재하고 있어서 언제나 불러서 사용할 수 있음

Ex) String.valueOf(123) 정적 메소드

Concat : 연결

Compare : 비교하다

Method Ref Param : 메소드 참조형 매개변수

// 메소드 헤드 : 반환형 이름 (파라미터 목록)

// 메소드 이름 -> 영소문자

// 메소드바디 : {--문장--}

// 메소드에서 값 반환하기

// 메소드 호출

// 호출 시의 값 전달방식

05/26

인스턴스

Ex 가로세로높이를 가질수 있는 공간이 생김

기본형

실제값을 저장

논리형, 문자형, 정수형, 실수형 8개

참조형

어떤 값이 저장되어 있는 메모리 주소를 값으로 저장

이전에 알아본 기본형이랑 다른 속성이라고 생각하면 됩니다. 참조형은 객체의 저장 공간을 주소를 저장합니다. 기본형 8가지 이외에는 모두 참조형이라고 생각하면 쉽습니니다. 참조형 변수는 null 값을 기본으로 가지고 있고 정의된 생성자를 통해 생성 됩니다. 힙 메모리 공간에 저장 됩니다.

- 기본형을 제외한 자료형

- 참조 값을 가진다.

- 클래스를 자료형으로 선언할때

- 배열, 스트링등

- 힙(heap)의 할당 된다 .Reference type(참조형)

와플기계로 와플을 찍어낸다 (클래스로 객체를 만듦)

🡺 인스턴스화 ex) box = new Box();

클래스란? 특정한 종류의 객체들을 찍어내는 형틀, 청사진 이라고 할 수 있다.

객체의 생성

1. 참조변수선언

Car 타입의 객체를 참조할 수 있는 변수 myCar를 선언한다.

1. 객체 생성

New 연산자를 이용하여 객체를 생성하고 객체 참조값을 반환한다.

1. 참조 변수와 객체의 연결

생성된 새로운 객체의 참조값을 myCar라는 참조 변수에 대입한다.

myCar.speed = 100;

myCar가 참조하는 객체로부터

speed라는 필드에 접근

myCar.print();

myCar가 참조하는 객체로부터

print()라는 메소드 호출