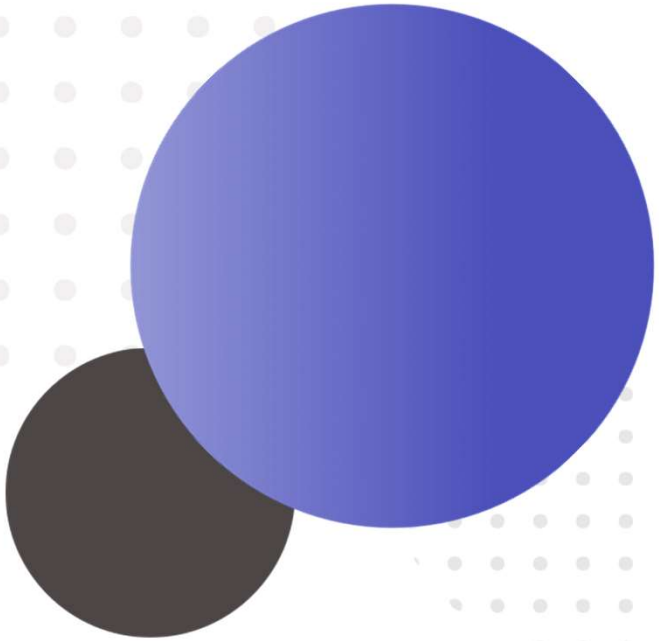


능동적 사고 방식의

ParkJuByeong

java

강사 박주병

Part08 생성자

01 생성자

02 this

03 멤버변수 초기화

04 실습 문제

ParkJuByeong

01

생성자

생성자

```java
public class Student
{
    String name;
    int kor;
    int eng;
    int math;


    Student()
    {

    }

    Student(String name1)
    {
        name = name1;
    }

    Student(String name1, int kor1, int eng1,int math1)
    {
        name =name1;
        kor =kor1;
        eng = eng1;
        math=math1;

    }
```

1.객체 생성시 최초 1번만 실행되는 특별한 메서드
2.클래스와 이름이 같아야한다(대소문자 구분)
3.리턴타입이 없어야 한다(void가 아님)
4.객체는 무조건 생성자를 통해서 생성된다.

생성자의 사용방법

```
Student student1 = new Student();
student3.name = "박주병";
student3.kor = 30;
student3.eng = 50;
student3.math = 20;
```



```
Student student1 = new Student();
Student student2 = new Student("홍길동");
Student student3 = new Student("박주병",30,50,20);
```

지금까지 클래스를 만들면서 생성자를 만든적이 없는데?

기본 생성자

```java
public class Student
{
    String name;
    int kor;
    int eng;
    int math;


    Student()
    {

    }

    Student(String name1)
    {
        name = name1;
    }

    Student(String name1, int kor1, int eng1,int math1)
    {
        name =name1;
        kor =kor1;
        eng = eng1;
        math=math1;

    }
```

생성자가 하나도 없다면 자동으로 만들어 준다.


```java
Student student1 = new Student();
```

```java
public class Student
{
    String name;
    int kor;
    int eng;
    int math;


    Student(String name1)
    {
        name = name1;
    }

    Student(String name1, int kor1, int eng1, int math1)
    {
        name =name1;
        kor =kor1;
        eng = eng1;
        math=math1;

    }
```



```java
Student student1 = new Student();
Student student2 = new Student("홍길동");
Student student3 = new Student("박주병",30,50,20);
```

→ 기본 생성자가 없어 ERROR!

기본 생성자로 생성시 기본값을 셋팅하면 어떨까?

```
Student student1 = new Student();
```

```
public class Student
{
    String name;
    int kor;
    int eng;
    int math;


    Student()
    {
        name = "홍길동";
        kor =0;
        eng = 0;
        math = 0;

    }
```

```
Student(String name1, int kor1, int eng1,int math1)
{
    name =name1;
    kor =kor1;
    eng = eng1;
    math=math1;

}
```


만들어놓은 생성자를 활용할수 없을까?

생성자에서 다른 생성자 호출

```
Student()
{
    System.out.println("생성자 호출");
    this("홍길동");
}
```

1.생성자의 가장 첫줄에 작성해야 한다.
2.메서드 이름을 this로 해야 한다.

```
Student()
{
    this("박주병",30,50,20);
}
```

왜 첫줄에 작성해야 할까?

```
Student()
{
    name = "김길동";
    this("홍길동");
}
```

→ 의미가 없는 코드가 된다.

```
Student()
{
    System.out.println("생성자 호출");
    this("홍길동");
}
```

→ 기능적으로는 문제 없지만 일반
메서드를 통해 멤버변수를 초기화
한다면 컴파일러가 체크할 방법이 없다.
따라서 무조건 첫줄에 작성하게
함으로써
가능성을 완전히 없앤다.

02

this

ParkJuByeong

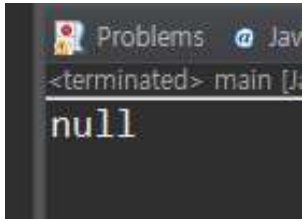
왜 생성자의 이름을 this로 바꿔야 할까?

```
Student()
{
    Student("박주병");
}


Student(String name1)
{
    name = name1;
}
```

ParkJuByeong

```
Student()
{
    new Student("박주병");
}


Student(String name1)
{
    name = name1;
}
```

```
Student()
{
    new  Student("박주병");

    System.out.println(name);
}
```



클래스 내부에서 자기 자신의 객체를 의미하는 변수가 필요하다

```
Student student1 = new Student();

System.out.println(student1);
```

```
<terminated> main [Java Application] C:₩Use
joo.Student@27d415d9
joo.Student@27d415d9
```

ParkJuByeong

```
Student()
{

    System.out.println(this);
}
```

```java
Student student1 = new Student();
Student student2 = new Student();

System.out.println("student1 : "+student1);
System.out.println("student2 : "+student2);
```

```java
void test()
{
    System.out.println(this);

}
```

```java
Student()
{


    System.out.println("this: "+ this);
}
```

ParkJuByeong

```java
void test()
{
    this.eng = 30;
    this.getTotal();
}
```

```
<terminated> main [Java Application] C:₩Users₩zest1₩.p2₩pool₩p
this: joo.Student@27d415d9
this: joo.Student@5c18298f
student1 : joo.Student@27d415d9
student2 : joo.Student@5c18298f
```

this는 자기 자신의 객체주소를 가지는 참조변수이다.

```java
Student(String name1, int kor1, int eng1,int math1)
{
    name =name1;
    kor =kor1;
    eng = eng1;
    math=math1;

}
```

```java
public class Student
{
    String name;
    int kor;
    int eng;
    int math;


    Student()
    {
        System.out.println("this: "+ this);
    }


    Student(String name)
    {
        name = name;
    }
```

ParkJuByeong

```java
Student(String name)
{
    this.name = name;
}
```

```java
Student(String name, int kor, int eng,int math)
{
    this.name =name;
    this.kor =kor;
    this.eng = eng;
    this.math=math;

}
```

퀴즈

```
static void test()
{
    this.eng = 30;
    this.getTotal();
}
```

static은 객체 생성이전에도
사용할수 있다.
따라서 객체를 가리키는
this는 사용할수 없다.

```java
Student student1 = new Student();
```

```java
Student()
{
    this("박주병",30,50,20);
}
```

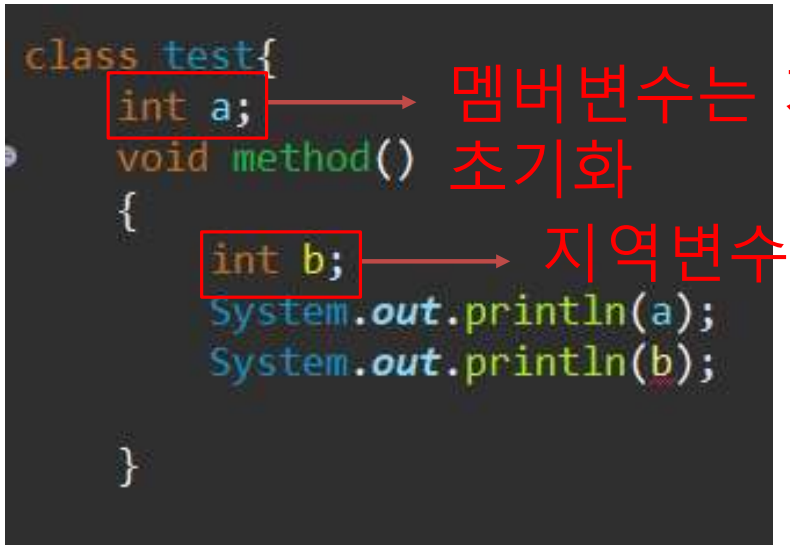
VS

```java
public class Student
{
    String name ="박주병";
    int kor = 30;
    int eng = 50;
    int math =20;

    Student()
    {
        System.out.println("this: "+ this);
    }

```

ParkJuByeong

03

멤버변수 초기화



멤버변수는 기본값 초기화

지역변수는 쓰레기값

ParkJuByeong

자료형	기본값
boolean	false
char	'\u0000'
참조형	null
그외	0

지역변수는 반드시 초기화를 해야 사용할수 있다.

멤버변수 초기화 방법 3가지

명시적 초기화

생성자

초기화 블럭

명시적 초기화

```
2
3 public class Student
4 {
5     String name ="박주병";
6     int kor = 30;
7     int eng = 50;
8     int math =20;
9
0
1●    Student()
2     {
3         System.out.println("this: "+ this);
4     }
5
```

ParkJuByeong

값을 하드코딩으로 바로 초기화 할경우 좋다.

초기화 블럭

```
public class Student
{
    String name ;
    int kor ;
    int eng ;
    int math ;
    static int number;

    Student()
    {
        number++;
    }


    Student(String name)
    {
        number++;
        this.name = name;
    }

    Student(String name, int kor, int eng,int math)
    {
        number++;
        this.name =name;
        this.kor =kor;
        this.eng = eng;
        this.math=math;

    }
```

ParkJuByeong

```
public class Student
{
    String name ;
    int kor ;
    int eng ;
    int math ;
    static int number;

    static { System.out.println("클래스 초기화 블럭");}

    {
        System.out.println("인스턴스 초기화 블럭");
        number++;
    }

    Student()
    {
        System.out.println("생성자");
    }

    Student(String name)
    {
        this.name = name;
    }

    Student(String name, int kor, int eng,int math)
    {
        this.name =name;
```

클래스 초기화 블럭

클래스가 메모리에 처음 로딩될때 한번만 실행

(객체가 메모리에 생성될때가 아니다!)

인스턴스 초기화 블럭

인스턴스가 만들어질때 마다 생성자보다 먼저 실행


```
Student student1 = new Student();

Student student2 = new Student();

Student student3 = new Student();
```

```
<terminated> main [Java Applicatio
클래스 초기화 블럭
인스턴스 초기화 블럭
생성자
인스턴스 초기화 블럭
생성자
인스턴스 초기화 블럭
생성자
```

```java
public class Student
{
    String name ;
    int kor ;
    int eng ;
    int math ;
    static int number;

    static {
        System.out.println("클래스 초기화 블럭");

        name ="김길동";
    }
```


클래스 초기화 블록은 객체 생성보다 이전에 실행되므로 인스턴스 변수는 사용할수 없다.

04

연습문제

ParkJuByeong

연습문제1

1. Student 클래스가 생성 될때마다 학생수 카운트를 올려보자

```
Student student1 = new Student();

Student student2 = new Student("박주병");

Student student3 = new Student("홍길동",32,13,23);

System.out.println(Student.count);
```

```
<terminat
3
```

클래스명	Marine,Zergling	
멤버변수	int hp	40
	int power	5
	int armor	0
메서드	powerUp()	매개변수:없음 내용: 모든 유닛의 power를 1증가 시킨다. 리턴:없음
	armorUp()	매개변수:없음 내용: 모든 유닛의armor를 1증가 시킨다. 리턴:없음
	attack(Marine) attack(Zergling)	
	showState()	유닛의 상태 출력

2. 오른쪽의 클래스를 만든후 각각 2개의 객체를 생성하여랜덤하게 공격하자.
(한팀씩 번갈아 가며 공격하며 어떤마린이 어떤 저글링을 공격할지는 랜덤이다.
즉 마린의 공격턴일때 공격안하는 마린이 있을수도 있다
어느 한쪽이 전멸할때까지 진행되며 승리팀이 어디인지 그리고 승리팀의 남은 유닛들의 상태를 출력하자


```
마린5 이 저글링2을 공격하였습니다.
마린1 이 저글링3을 공격하였습니다.
저글링3이 죽었습니다.
마린팀 공격턴 종료
저글링2 이 마린5을 공격하였습니다.
마린5이 죽었습니다.
저글링팀 공격턴 종료
마린1 이 저글링2을 공격하였습니다.
저글링2이 죽었습니다.
마린팀 승리 ! 남은 유닛 1개
```

THANK YOU

ParkJuByeong

강사 박주병