생성자

생성자

- **생성자(Constructor)**

  객체를 생성할 때 호출되며, 객체를 초기화하는 메소드
  * 멤버 변수들을 기본값으로 초기화

생성자 호출

- **생성자 호출**

아래와 같이 이미 자신도 모르게 생성자를 호출해왔음

```
Calc c = new Calc();
```

→ 생성자 호출(초기화)

- **참고**

생성자를 따로 만들지 않았는데, 위와 같이 사용이 가능했던 이유는 컴파일 시 생성자가 하나도 없을 경우 자동으로 생성자를 만들어줌

생성자 규칙

- **생성자 규칙**

1. 클래스 이름과 동일해야 함
2. 리턴값이 없음 (void 없음)
   → 생성자의 경우 항상 반환되는 값이 없기 때문에 void를 생략하고 사용함
3. 모든 클래스는 최소 1개 이상의 생성자를 가져야 함
   → 이러한 이유로 컴파일 시 생성자가 하나도 없다면 자동으로 생성해줌

생성자의 종류

- **생성자의 종류**

생성자에는 기본 생성자와 매개변수가 있는 생성자가 있으며 기본 생성자는 매개변수가 하나도 없는 생성자를 뜻함.

* 컴파일 시 생성자가 하나도 없을 경우 만들어주는게 기본 생성자

```
public class Constructor {

    public Constructor() {} // 기본 생성자
    public Constructor(int num) {} // 매개변수 있는 생성자

}
```

생성자의 종류

- 생성자의 종류

1. ExData1은 생성자가 하나도 없기 때문에 컴파일할 때 자동으로 기본 생성자를 만들어줌. 생성자 호출 부분 ExData1() 에서 에러를 발생시키지 않음
2. ExData2는 기본 생성자가 없기 때문에 에러 발생, 매개변수가 있는 생성자만 있기 때문에 생성자 호출 부분 ExData2() 에서 에러 발생

```
public class Constructor {
    ExData1 ed1 = new ExData1();
    ExData2 ed2 = new ExData2();
}

class ExData1 {

}

class ExData2 {
    int value;

    ExData2(int x) {
        value = x;
    }
}
```

The constructor ExData2() is undefined

3 quick fixes available:

- Add argument to match 'ExData2(int)'
- Change constructor 'ExData2(int)': Remove parameter 'int'
- Create constructor 'ExData2()'

매개변수 있는 생성자

- **멤버 변수 초기화**

기존에 멤버 변수를 초기화하기 위해서는 객체를 생성하고, 멤버 변수를 하나씩 초기화 해주는 작업을 해야함

```
ExData2 ed2 = new ExData2();
ed2.age = 12;
ed2.name = "김재섭";
```

- **매개변수 있는 생성자**

매개변수 있는 생성자를 만들어둠으로써 코드가 간결 해지고 개발자가 멤버 변수를 초기화 하기가 굉장히 쉬워짐

```
ExData2 ed2 = new ExData2(12, "김재섭");
```

멤버 변수와 지역 변수 초기화

- **멤버 변수와 지역 변수 초기화**

아래의 코드에서 멤버 변수와 지역 변수 둘 다 초기화 없이 선언만 되었으나
멤버 변수는 기본 생성자에 의해 초기화 되므로 사용이 가능함

```
public class Test {

    public static void main(String[] args) {
        Var v = new Var();
        System.out.println(v.num);
        v.varMethod();
    }

}

class Var {
    int num; // 멤버 변수(인스턴스 변수) 선언

    public void varMethod() {
        int local; // 지역 변수 선언
        System.out.println(local);
    }

}
```

생성자 활용하기 - 1

- **Q1. Product 클래스를 만들고 생성자를 사용해 상품 이름, 가격, 수량 변수를 초기화 하고 출력하시오.**

1. Product 클래스를 만드세요.
2. 아래의 변수들을 선언하세요.
   → name : 상품 이름
   → price : 상품 가격
   → count : 상품 수량
3. Product 클래스에 기본 생성자와 매개변수가 있는 생성자를 만드세요.
4. 매개변수가 있는 생성자는 아래 조건에 맞추세요.
   → 매개변수를 name, price, count 받는 생성자
   → 매개변수를 name, price 받는 생성자 (count의 값 : 0)
5. 기본 생성자, 매개변수 있는 생성자(2개)를 각각 객체 생성하고 멤버 변수를 출력하여 입력한 값으로 초기화 되었는지 확인하세요.

참조변수 this

참조변수 this

- **참조변수 this**

  모든 인스턴스 메소드에 숨겨진 채 존재하며 할당된 객체를 가르킴
  → 인스턴스 자기 자신을 가르키는 참조변수

- **참고**

  멤버 변수와 지역 변수를 구분 짓기 위해 사용됨

참조변수 this

- 설명

멤버 변수와 매개 변수의 이름이 같음
매개 변수의 값을 멤버 변수에 넣기 위해서는 this를 사용해야 함

아래의 코드처럼 작성할 경우 지역변수name = 지역변수name 이 됨

```
public class QuizProduct {
    String name;
    int price;
    int count;

    public QuizProduct() {}

    public QuizProduct(String name, int price, int count) {
        name = name;
        price = price;
        count = count;
    }
```

참조변수 this

- 설명

멤버 변수와 지역 변수를 구분 지어 사용하기 위해 this가 사용됨

아래의 코드처럼 작성할 경우 멤버변수name = 지역변수name 이 됨

```java
public class QuizProduct {
    String name;
    int price;
    int count;

    public QuizProduct() {}

    public QuizProduct(String name, int price, int count) {
        this.name = name;
        this.price = price;
        this.count = count;
    }
```

생성자 this()

생성자 this() - 1

- **생성자 this()**

  같은 클래스의 다른 생성자를 호출할 때 사용
  * 반드시 첫 줄에 선언되어야 함

- **참고**

  생성자 this()가 첫 줄에 선언되지 않았기 때문에 에러 발생

```
public Academy() {
    this.age = 0;
    this(19, "김재섭");
};
```

생성자 this() - 2

```java
public static void main(String[] args) {
    Academy academy = new Academy();
    System.out.println(academy.age);
    System.out.println(academy.name);
}
```

기본 생성자 호출

```java
class Academy {
    int age;
    String name;

    public Academy() {
        this(19, "김재섭");
    };

    public Academy(int age, String name) {
        this.age = age;
        this.name = name;
    }
}
```

생성자 this() - 2

```java
public static void main(String[] args) {
    Academy academy = new Academy();
    System.out.println(academy.age);
    System.out.println(academy.name);
}
```

```java
class Academy {
    int age;
    String name;

    public Academy() {
        this(19, "김재섭");
    };

    public Academy(int age, String name) {
        this.age = age;
        this.name = name;
    }
}
```

매개변수 있는 생성자 호출

생성자 this() - 3

```
public static void main(String[] args) {
    Academy academy = new Academy();
    System.out.println(academy.age);
    System.out.println(academy.name);
}
```

멤버 변수 초기화

```
class Academy {
    int age;
    String name;

    public Academy() {
        this(19, "김재섭");
    };

    public Academy(int age, String name) {
        this.age = age;
        this.name = name;
    }
}
```

생성자 this() 사용하는 이유

- 생성자 this() 사용하는 이유

> 똑같은 동작을 하는 코드이지만, 생성자 this()를 사용하여 코드의 중복을 제거

```java
class Academy {
    int age;
    String name;

    public Academy() {
        this.age = 0;
        this.name = "이름 없음";
    };

    public Academy(int age, String name) {
        this.age = age;
        this.name = name;
    }
}
```



```java
class Academy {
    int age;
    String name;

    public Academy() {
        this(0, "이름 없음");
    };

    public Academy(int age, String name) {
        this.age = age;
        this.name = name;
    }
}
```

생성자 활용하기 - 2

- **Q1. Student 클래스를 만들고 생성자를 사용해 이름, 나이, 학년, 학번을 출력하시오.**

1. Student 클래스를 만드세요.
2. 아래의 변수들을 선언하세요.
   → name : 학생 이름
   → age : 학생 나이
   → grade : 학년
   → studentNumber : 학번 (ex. 23052112)
3. Student 클래스에 기본 생성자와 매개변수가 있는 생성자를 만드세요.
   → 매개변수 있는 생성자는 1개 (name, age, grade, studentNumber 4개를 받음)
4. 매개변수 있는 생성자의 매개변수 이름은 멤버 변수와 똑같이 네이밍 하세요.
5. 기본 생성자를 호출했을 때 아래가 출력되게 하세요.
   → name : 이름 없음
   → age : 0
   → grade : 0
   → studentNumber 00000000