0308

06_OOP.pdf 2 MB

생성자

객체가 new 연산자를 통해 Heap 메모리 영역에 할당될 때
객체 안에서 만들어지는 필드 초기화 + 생성 시 필요한 기능 수행

생성자는 일종의 메소드로 전달된 초기값을 받아서 객체의 필드에 기록

생성자는 두 종류가 있다

[접근제한자] 클래스명( ) { };
[접근제한자] 클래스명(매개변수) { (this.필드명.사용자};

매개변수 / Overloding / this
overloding 을 적용하면
한 이름 당 기능 1개 -> 한 이름 당 기능 여러 개
this( ) :

Method :

한개의 메소드는 한개의 기능만

Praction

회원가입
로그인
회원정보 조회
회원정보 수정

Member.Java

Member class 필드 (속성)

변수들을 먼저 선언하고 생성자를 생성한다.
기본생성자 : 클래스와 이름이 같으면서 반환형이 존재 하지 않게 작성한다

- [접근제한자] 클래스명 (매개변수) { } ;

매개변수 생성자 :기본생성자와 기본 구성은 같지만 클래스명 오른쪽 괄호에 매개변수가 들어간다.

- 매개변수 : 다른 클래스의 값을 옴겨 담아온다.

위와 같이 다른 클래스에서 매개 변수를 가져온 다음
해당 클래스 안의 변수에 값을 대입한다
this.
접근 제한자

- **public +** : 모든 부분에서 접근 가능하다.
- **protected #** : 같은 클래스 / 같은 패키지 /상속 받은 자식까지 접근 가능하다
- **(default) ~** :(괄호는 생략된다). 같은 클래스 /같은 패키지 안에서만 접근 가능하다.
- **private -** : 직접 접근 제한의 규칙으로 인해 기본적으로 선언한다.

변수들이 private으로 선언 되어 있음으로 getter setter을 생성하여 사용한다.

MemberService.Java

회원의 정보와 로그인한 회원의 정보를 저장하는 Member Class를 참조하는 변수를 선언했다.

Member Class를 import하여 이런식으로 사용 가능하다.

그 다음 기능으로 프로그램이 실행 되면 가장 먼저 화면에 출력될 MainDisplay 메소드를 만들었다

그 다음 로그인 기능에 해당하는 메소드를 만들었다.

각 변수명에 정보들을 입력 받아
이미 Member클래스를 참조하여 선언한 회원 정보를 기억하는 MemberInfo에 매개변수생성자를 이용하여 저장한다
그 다음
MainDisplay 변수에 signUp의 결과값을 리턴하여 출력한다.

로그인에 해당하는 메소드를 작성한다

고객의 비밀번호를 입력받아 본인이 맞을 경우 정보 열람
String을 비교하는데 Memeber안에 있는 비밀번호를 getter로 가져와서 확인한다 비교한다.
회원정보를 조회하는 메소드는
그냥 비밀번호 맞는지 조회하고 다 출력하면 됨

회원정보 수정