

0907 - 회원 서비스 테스트

- **ctrl + shift + t** - test 클래스 만들기
- 테스트에서는 한글로 작성해도 된다
- **ctrl + alt + v** - 변수 추출
- **Constructor** - **alt + insert**

MemberServiceTest

```
package hello.hello_spring.service;

import hello.hello_spring.domain.Member;
import hello.hello_spring.repository.MemoryMemberRepository;
import org.assertj.core.api.Assertions;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

import java.util.Optional;

import static org.assertj.core.api.Assertions.*;
import static org.junit.jupiter.api.Assertions.*;

class MemberServiceTest {

    MemberService memberService;
    MemoryMemberRepository memberRepository;

    @BeforeEach
    public void beforeEach() {
        memberRepository = new MemoryMemberRepository();
        memberService = new MemberService(memberRepository);
    }
}
```

```

@AfterEach
public void afterEach(){
    memberRepository.clearStore();
}

@Test
void 회원가입() {
    //given
    Member member = new Member();
    member.setName("spring");

    //when(회원가입했을 때)
    Long saveId = memberService.join(member);

    //then(결과)
    Member findMember = memberService.findOne(saveId).get
    assertThat(member.getName()).isEqualTo(findMember.get

}

@Test
public void 중복_회원_예외() {
    //given
    Member member1 = new Member();
    member1.setName("spring");

    Member member2 = new Member();
    member2.setName("spring");

    //when
    memberService.join(member1);
    IllegalStateException e = assertThrows(IllegalStateException

    assertThat(e.getMessage()).isEqualTo("이미 존재하는 회원입니다")

    /*
    try {
        memberService.join(member2);
        fail();
    }

```

```

        } catch (IllegalStateException e) {
            assertThat(e.getMessage()).isEqualTo("이미 존재하는
        }
    */

    //then
}

@Test
void findMembers() {
}

@Test
void findOne() {
}

}

```