

PROJECT

당근사

```
LIST<STUDENT> 5줬조 = NEW ARRAYLIST<STUDENT> ( );
```

KEYWORD

MEMBER : 회원 가입 및 회원관리
PRODUCT : 물품관리 및 후기 관리
BOARD : 중고거래, 후기작성
ORDER : 장바구니 및 주문내역



5점조.TOSTRING();

이동건 [LEADER]

"아니야 괜찮아 잘 했네! (주먹을 쥐며)"

중고거래

MAIN 메서드 정리

고은채

"많고 많은 ~개발자들중 ~내가 제일 잘났지!"

쇼핑몰 이용 고객 등록 및 관리 구현

쇼핑몰 관리자 계정 구현

오상현

"커피 시키신 분?"

장바구니에 물건 추가, 삭제, 및 조회

장바구니 물건 구매 및 주문내역 조회

우성준

"우성준 어록: 다 물어 보세요, 저 컴퓨터에요! (화면에 보이는 무수한NULL포인트)

"

쇼핑몰 물류 생성 및 관리

구매물품 후기 등록 구현

HISTORY

02/06

"위대한 여정의 시작"

- 사전기획단계
- 주제 선정 : 의류 쇼핑몰
- member 객체 생성
- 부여받은 객체 생성

02/07

"아니 제꺼에선 잘 돌아갔는데"

- 모델링 시작
- VO생성 및 변수명 추가
- 싱글톤으로 인한 필드 선언부 조정
- 초기화 블록에서 변수 선언

02/08

"중요한건 꺾이지 않는 싱글톤"

- 서비스 메소드 구현
- 로그인정보 싱글톤 처리
- 메인메서드에서 싱글톤 테스트

02/09

"아니, 진짜, 왜, 메인만 오면"

- member 서비스 로그인정보 전달
- cart, product 메서드 구현 및 동기화
- 메인 switch문에서 변수 영속성 문제 발생

02/13

"포모스 선정 뒤틀린 황천의 들여쓰기 1위"

- 주요 메서드 기능 구현 완료
- 피드백 실시
- 중고거래 거리기능 보완
- 메인메소드 재추합

02/14

"함께 해서 즐거웠고..."

- 서비스구축
- 싱글톤 및 영속성 최종수정
- 메서드 상호 호출 및 동기화
- 유효성 검사
- 메인메소드 구동 확인

SKILL

Q

LIST배열

W

파일 영속화

E

싱글 톤

R

카카오API



LIST 배열

- 배열과 유사
 - 길이변경이 자유로움
 - 추가된 순서 유지
- = 게시판, 리뷰, 회원정보,
물품정보, 장바구니, 주문내역

```

    setProducts(List<Product> products) {
        products = products;

        List<Product> ware = new ArrayList<Product>();

        try (ObjectInputStream inputStream = new ObjectInputStream(new File(
            ware = (List<Product>) inputStream.readObject();
        } catch (FileNotFoundException e) {
            ware.add(new Product(1001, "청바지", "하의", 1, 30000, new Date(
            ware.add(new Product(2002, "티셔츠", "상의", 2, 20000, new Date(
            ware.add(new Product(3003, "귀걸이", "악세", 3, 15000, new Date(
            saveWare();
        } catch (ClassNotFoundException | IOException e) {
            e.printStackTrace();
        }
    }

    List<Product> getWare() {
        return ware;
    }

    void setWare(List<Product> ware) {
        ware;
    }

```

```

    BoardServer {

        ArrayList<UBoard>(); // UBoard타입의 글목록을 담을 board
        = new ArrayList<UBoard>();
        // 영속화한 member파일에 값 담을 공간
        memberServiceImpl.getInstance(); // Login한 멤버의 주소값 불러오기
        new RangeException();

        ObjectInputStream oisboard = new ObjectInputStream(new FileInputStream("ubrList.ser"));
        UBoard board = (UBoard) oisboard.readObject();
    }

    UBoard("police", "서울특별시 구로구 가마산로 235(구로동)", boardCnt, "후
    4988407, 37.4945320625106));

    UBoard("center", "서울특별시 구로구 디지털로31길 93(구로동)", boardCnt,
    136892, 37.4868233865633));

```

파일 영속화

- JAVA 내부 시스템에서 사용되는 객체나 데이터를 외부에서 사용할 수 있도록 BYTE형태로 변환
- 바이트와 문자 기반으로 나뉘며, 각각 입력과 출력이 있음
- 입출력을 동시에 할 수 없음

싱글톤

- 무분별한 생성자를 막기 위해 사용
 - 객체의 인스턴스가 오직 1개
 - 클래스 간 데이터 공유에 용이
- = 로그인 한 회원의 값을 공유하여,
주문내역, 물품후기 작성
- = 물품 구매시 물품 총량의 변동 적용

```
class OrderServiceImpl implements OrderService {
    // 싱글톤
    private static OrderService orderService = new OrderService();

    private OrderServiceImpl() {
    }

    public static OrderService getInstance() {
        return orderService;
    }

    private List<Order> receipts = new ArrayList<Order>();

    public List<Order> getReceipts() {
        return receipts;
    }

    public void setReceipts(List<Order> receipts) {
        this.receipts = receipts;
    }
}
```

```
private CartService cartService = CartServiceImpl.getInstance();
```

메서드 (경도, 위도)

```
public void findAddress(String addr) throws UnsupportedEncodingException {
    // 검색 갯수
    int size = 30;

    // 검색 키워드
    String keyword = addr;

    String result = getJSONData(
        String.format("https://dapi.kakao.com/v2/local/search/geojson.json?query=%s",
            URLEncoder.encode(keyword, "utf-8"), size));

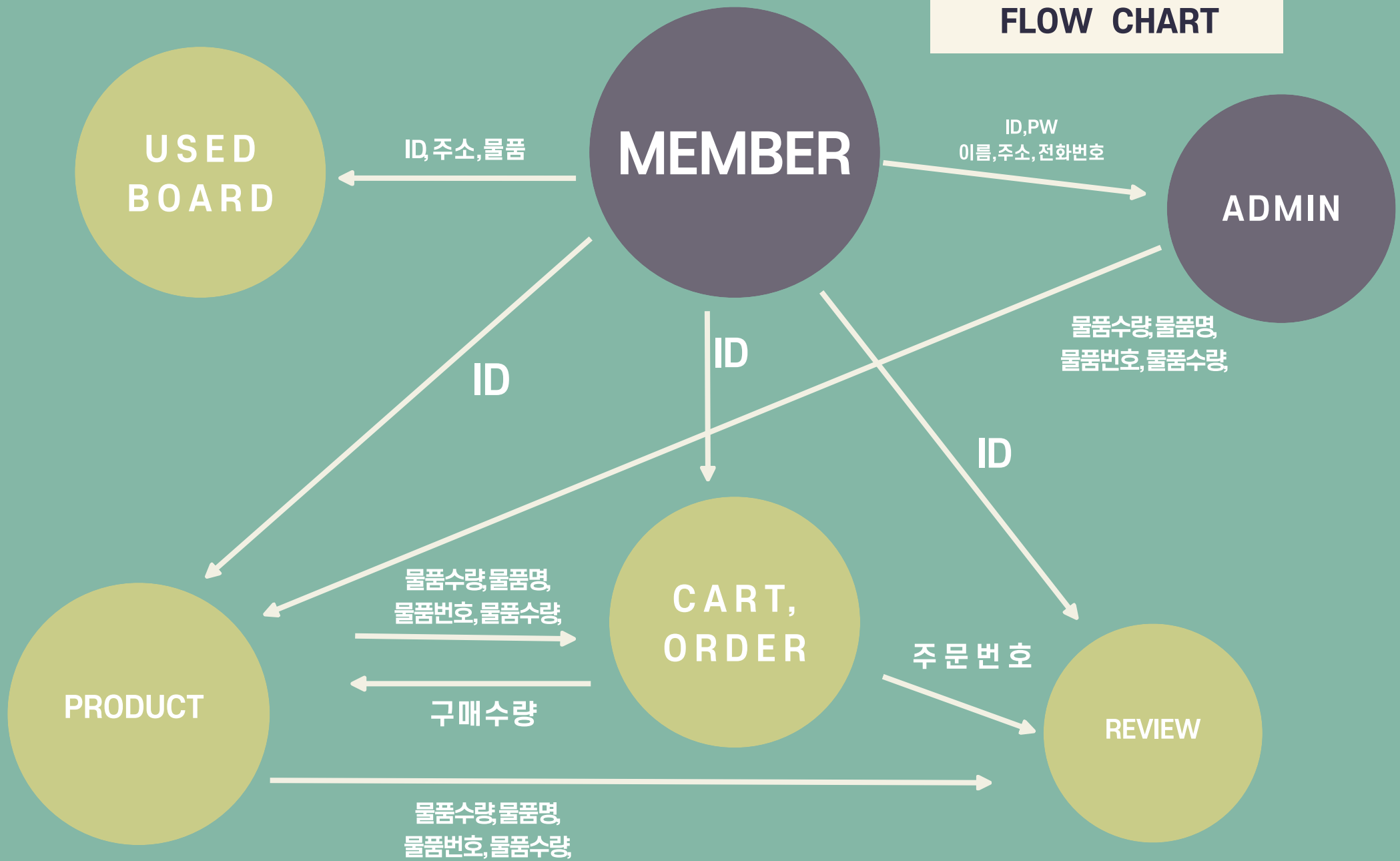
    Gson gson = new Gson();
    Map<String, ?> map = gson.fromJson(result, Map.class);

    ((List<?>) (map.get("documents"))).forEach(
        o -> {
            if (o instanceof Map) {
                Map<String, ?> m = (Map<String, ?>) m;
                x = Double.valueOf(m.get("x"));
                y = Double.valueOf(m.get("y"));
            }
        }
    );
}
```

카카오 API

- 콘솔에 입력한 주소를 바탕으로
카카오 맵으로부터 위도와 경도값으로
전환
- 좌표값으로 거리를 산출하여 지역 내
인근 중고거래 게시글 활성화

FLOW CHART



RUN

프로그램 구동



우성준

정보처리기사 공부 할 때만 보았던 디자인패턴을 실제로 사용해서 신기했습니다.

코드를 짜면서 팀원들과의 소통이 중요하다는 것을 깨달을 수 있었습니다.

각자 구조부터 생각이 달라서 조정하는 작업이 필수였기 때문입니다.

무엇보다 다양한 사람들과 다양한 관점에서 코드를 볼 수 있다는 점이 좋았습니다.

덕분에 막혔던 부분에서 조금씩 해결할 수 있었습니다.

어려운 시간이었지만, 이 경험들을 바탕으로 발전해서 차기 프로젝트가 마냥 걱정만 되는것이 아닌, 더욱 좋은 결과를 볼 수 있을것 같다는 기대감과 자신감을 얻었습니다.

감사합니다.

오상현

처음 주제선정이 되고 역할이 정해진 뒤 자리에 앉아 이클립스 화면을 보고 무엇을 해야 할지 몰랐습니다. 다행히 다른 조원분들의 도움으로 늦었지만 구조를 할 수 있었습니다.

프로젝트를 통해 느낀점은 팀원들간의 원활한 의사소통을 통해 상호 협력의 중요성을 알게 되었습니다. 상대방에게 무엇이 필요한지 알아내는 것도 중요하지만 내가 무엇이 필요하고 부족한지 설명하는 것 또한 중요하다는것을 깨달았습니다.

결과도 당연히 중요하지만 그 결과까지의 효율적인 방법이 무엇인지 탐색하는 것의 중요성을 배울 수 있었습니다.

이번 프로젝트에서 느낀 부족함과 배운점을 토대로 앞으로의 과정에서도 열심히 임하겠습니다. 감사합니다.

고은채

주제를 선정하는 것 부터 쉽지 않았고, 특히나 자바에 대한 감을 잡지 못하는 상태로 미니 프로젝트를 시작하게 되었습니다.

주어진 역할을 어떻게 수행해 나가야 하는지 정하는 초반 설계단계에서 막막함을 느꼈으나 팀원들의 도움으로 변수 생성부터 마지막 단계인 예외처리까지 많은 것을 배우며 프로젝트를 완성해 나갈 수 있었습니다.

혼자 프로그래밍 하는 것이 아닌, 팀원들과 서로 비교하며 의논하고 결과를 찾아가는 과정을 통해 어떤 점이 문제이고, 효율적인 방법은 무엇인지 고민해보며 이를 통해 실질적인 구조를 많이 접해 볼 수 있어서 더욱 큰 경험이 되었습니다.

마지막 순간까지 큰 도움을 주었던 조원분들과 선생님께 감사드립니다.

이동건

초반에 구성했던 변수명이나 메소드들이 주체가 누가 되어야 하는 것이 힘들었습니다.
선생님의 피드백 덕분에 빠르게 정리가 되었고, 피드백 내용처럼 초반에 주체가 누구 인지 파악하고 정리하는 작업의 중요성을 다시금 깨달을 수 있었습니다.

중간중간 마주치는 문제들로 인해 추가, 삭제, 수정 작업을 하며 같은 내용을 여러번 다시 구현해야 하는 부분들도 있었지만, 팀원들이 서로서로 도와주는 모습과 본인이 맡은 역할 이상으로 팀에 기여를 하는 부분을 보며 팀워크의 조화가 잘 이루어졌다고 생각했고, 5조와 함께 미니프로젝트를 진행하며 협력하는 관계를 더 배울 수 있어 좋았습니다.

감사합니다.

END

감사합니다!

BREAK;