

**PORTFOLIO**



LEE HYE JOO

## CONTENTS

**JAVA**

Console RPG

**HTML & CSS & JS**

D-Day

**HTML & CSS & JS & JQuery**

BOOK SEARCH

**JSP**

BOARD

**Spring**

LEIM Project

# **JAVA** Console RPG

## JAVA Console RPG

프로젝트 명 : **Console RPG**

개발기간 : 2022.06.20 - 2022.06.27 (7일)

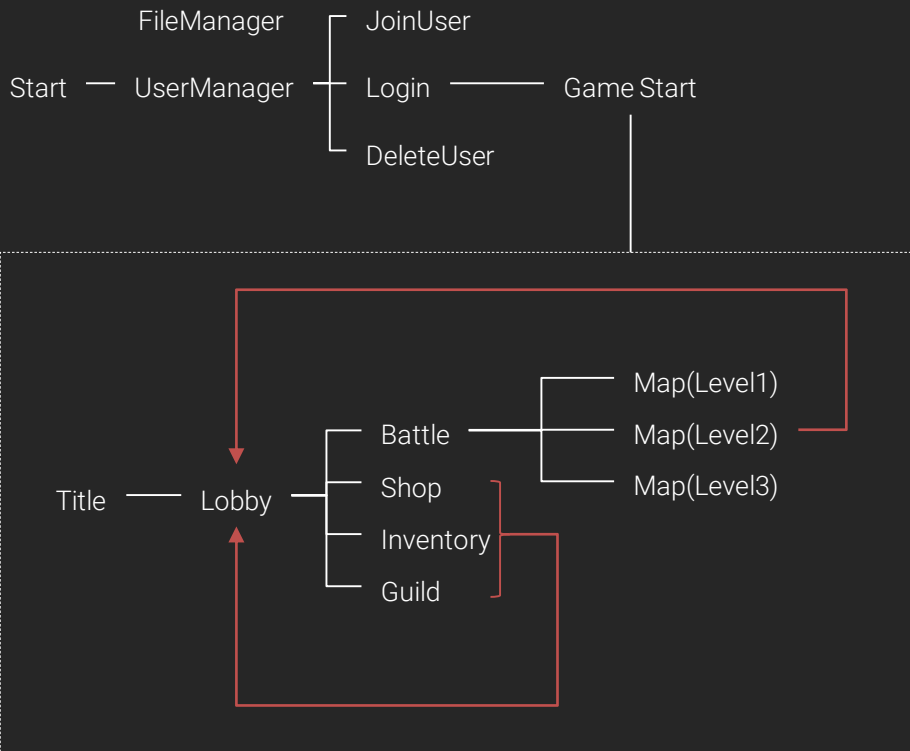
개발인원 : 1명

요약 : RPG 게임 구현

### 개발환경

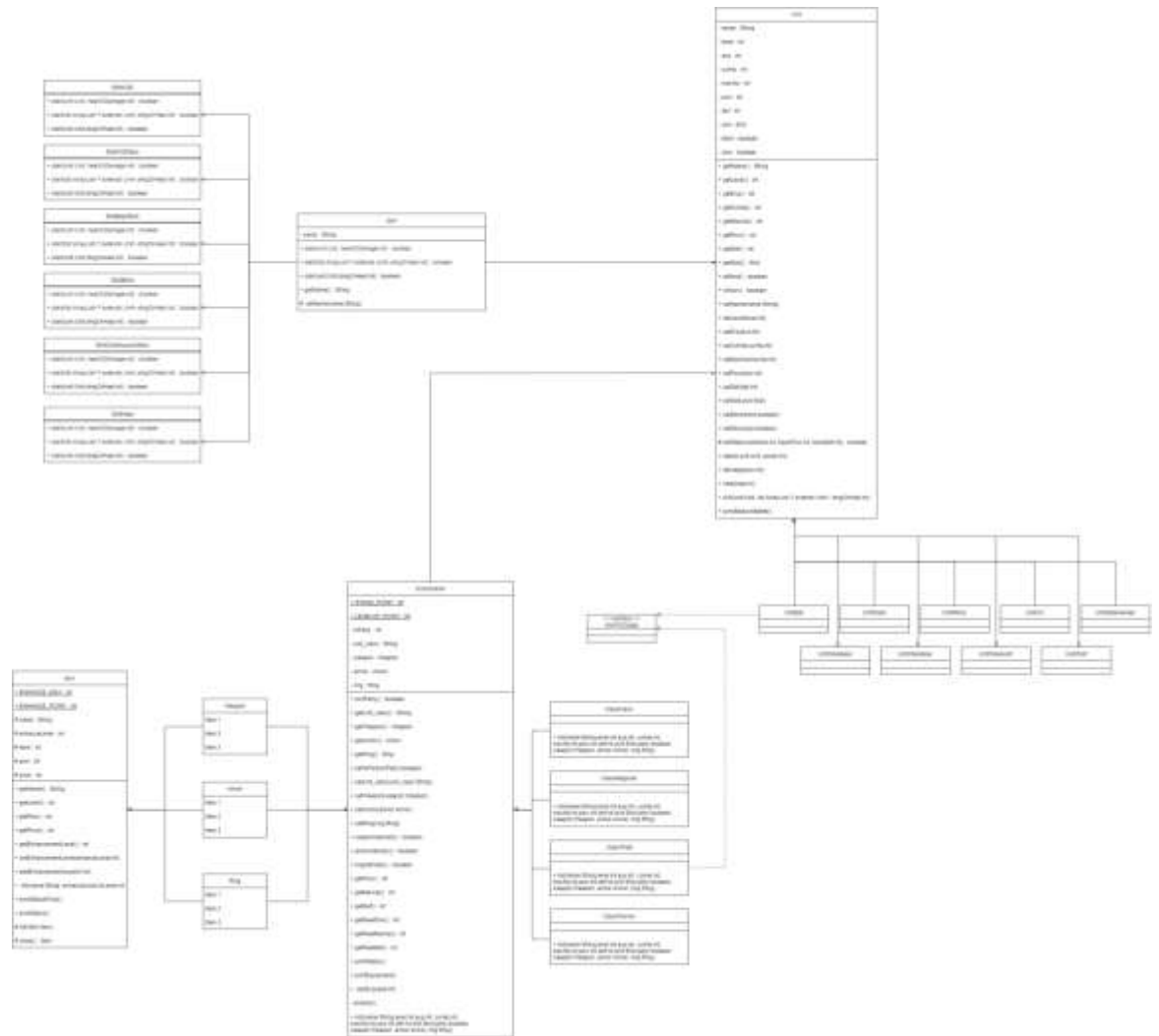
- OS : Windows10
- Development Tool : Eclipse
- Language : JAVA 11

- 추상화 클래스 Stage를 통한 루프 관리
- Join 기능을 통한 다 계정 Data 처리
- Skill과 Unit의 추상화를 통해 Unit 양산
- **Interface**를 통한 특정 분류 관리

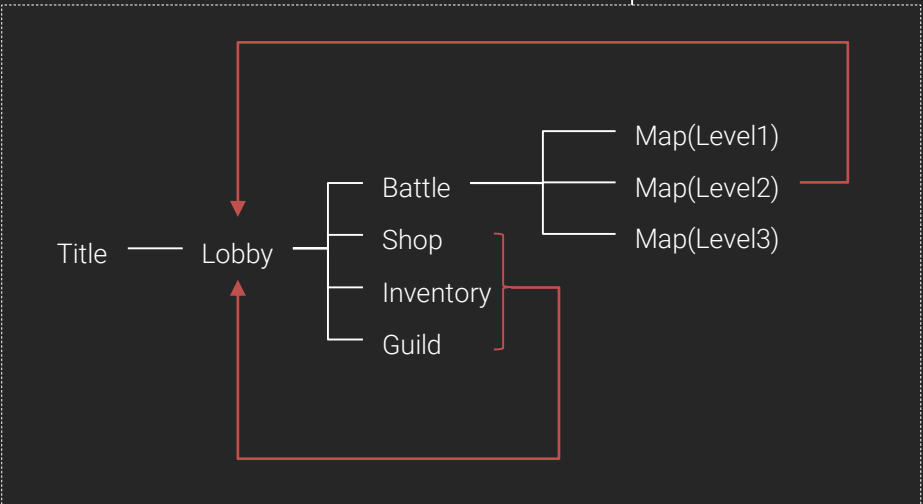
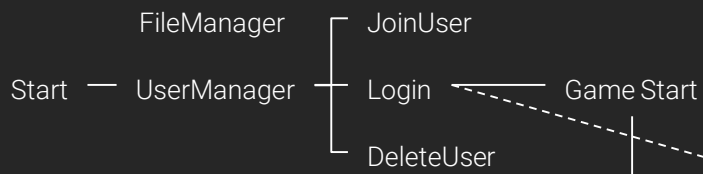


# JAVA Console RPG

## 추상화 클래스를 통한 Unit, Item 관리



# JAVA Console RPG



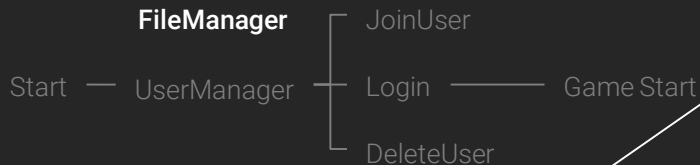
```
private boolean loginOver() {
    System.out.print("id:");
    String id = GameManager.scan.next();
    System.out.print("pw:");
    String pw = GameManager.scan.next();

    if (dao.checkIdPw(id, pw)) {
        System.out.printf("로그인 되었습니다. 환영합니다. [%s]님.\n", dao.returnUser(id, pw).getId());
        new GameManager(dao.returnUser(id, pw)).run();
        return false;
    } else {
        System.out.println("잘못된 정보입니다.");
    }
}

return true;
}
```

```
[1]로그인
[2]회원가입
[3]회원탈퇴
[0]종료
1
id:apple
pw:111
로그인 되었습니다. 환영합니다. [apple]님.
::::: TEXT RPG ::::::
[시작]을 입력하세요.
```

# JAVA Console RPG - File Manager



```
private String dataDirectory = System.getProperty("user.dir") + "\\UserData";

public void saveFile(User user) {
    String fileName = dataDirectory + "\\" + user.getId() + ".txt";
    File = new File(fileName);

    try {
        fw = new FileWriter(File);
        fw.write(user.getDataString());
        fw.close();
        System.out.println(user.getId() + " saved");
    } catch (IOException e) {
        e.printStackTrace();
        System.out.println(user.getId() + " not saved");
    }
}
```

**SAVE**

User Data를 저장하는 Directory를 `System.getProperty()`를 통해 Path를 설정하고 생성

그 후, FileWriter로 현재 로그인 되어 있는 유저의 data를 String으로 변환하는 getDataString()를 통해 data를 txt 파일로 저장.

## JAVA Console RPG - File Manager

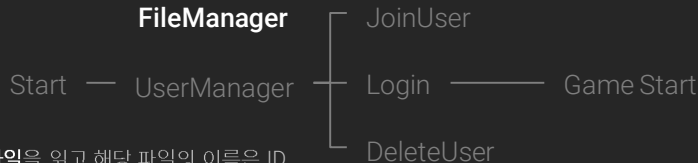
```
Map<String, User> dao = new HashMap<String, User>();  
  
File dir = new File(dataDirectory);  
FilenameFilter filter = new FilenameFilter() {  
    @Override  
    public boolean accept(File dir, String name) {  
        return name.contains(".txt");  
    }  
};  
String[] key = dir.list(filter);  
  
for (int k = 0; k < key.length; k++) {  
    file = new File(dataDirectory + "\\\" + key[k]);  
    try {  
        fr = new FileReader(file);  
        br = new BufferedReader(fr);
```

Data로 읽은 User를 생성하는 방법은 Class와 제네릭을 통해 특정 추상화 클래스를 상속받는 임의의 객체를 생성하고 해당 클래스의 init() 메소드를 통해 정보를 Setting

### LOAD

FilenameFilter를 통해 User의 정보만 담긴 파일을 읽고 해당 파일의 이름은 ID 즉, User의 key값이므로 배열에 따로 저장 후, FileReader를 통해 Data를 읽고 User를 생성, key 배열에 저장된 내용과 함께 HashMap에 저장

### FileManager

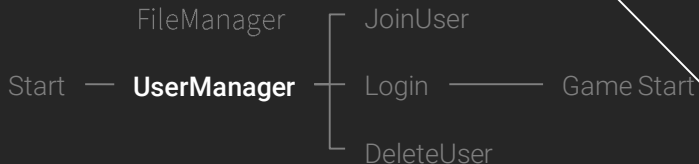


```
[ Class<? extends Employable> clazz = (Class<? extends Employable>) Class.forName(info[0]);  
Employable unit = clazz.getDeclaredConstructor().newInstance();  
  
String name = info[1];  
int level = Integer.parseInt(info[2]);  
int exp = Integer.parseInt(info[3]);  
int curHp = Integer.parseInt(info[4]);  
int maxHp = Integer.parseInt(info[5]);  
int pow = Integer.parseInt(info[6]);  
int def = Integer.parseInt(info[7]);  
  
Class<? extends Skill> skillClazz = (Class<? extends Skill>) Class.forName(info[8]);  
Skill skill = skillClazz.getDeclaredConstructor().newInstance();  
  
boolean party = info[9].equals("1") ? true : false;  
  
unit.init(name, level, exp, curHp, maxHp, pow, def, skill, party, weapon, armor, ring);  
  
User user = new User(id, pw, guild, inventory, gold);  
dao.put(user.getId(), user);
```



## JAVA Console RPG - User Manager

User 회원가입 시, 기본 자금과 4명의 1Level의 Employable Unit 생성



탈퇴 시, FileManager와 연동하여 User Data 파일 삭제 처리

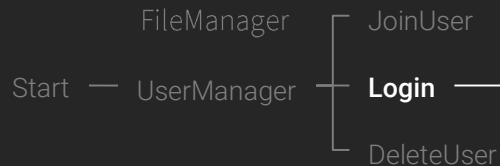
```
public void deleteUser(User user) {
    String fileName = dataDirectory + "\\" + user.getId() + ".txt";
    File file = new File(fileName);
    if (file.exists()) {
        if (file.delete()) {
            System.out.println("파일 삭제 성공");
        } else {
            System.out.println("파일 삭제 실패");
        }
    }
}
```

```
public User(User idPwTemp) {
    this.id = idPwTemp.getId();
    this.pw = idPwTemp.getPw();
    guild = new ArrayList<>();
    inventory = new ArrayList<>();
    // gold = 100000;
    gold = 1000000000;
    init();
}
```

```
public void init() {
    // 초기 골드원 생성
    guild.add(new ClassCleric("프랑이", 1));
    guild.add(new ClassWarrior("토끼", 1));
    guild.add(new ClassMagician("고양이", 1));
    guild.add(new ClassThief("코끼리", 1));
    for (Employable mem : guild) {
        mem.setInParty(true);
    }
}
```

## JAVA Console RPG - Game Manager ( Title, Lobby )

로그인 성공 시, GameManager를 통해 Game Start



```

if (dao.checkIdPw(id, pw)) {
    System.out.printf("로그인 되었습니다. [%s]님, \n", dao.returnUser(id, pw).getId());
    new GameManager(dao.returnUser(id, pw)).run();
    return false;
} else {
    System.out.println("동영상 보았습니다.");
}
  
```

```

private void initStage_list() {
    stage_list = new HashMap<String, Stage>();

    Stage stage = new StageTitle();
    stage_list.put(stage.getTitle(), stage);

    stage = new StageLobby();
    stage_list.put(stage.getTitle(), stage);

    stage = new StageInventory();
    stage_list.put(stage.getTitle(), stage);

    stage = new StageGuild();
    stage_list.put(stage.getTitle(), stage);

    stage = new StageShop();
    stage_list.put(stage.getTitle(), stage);

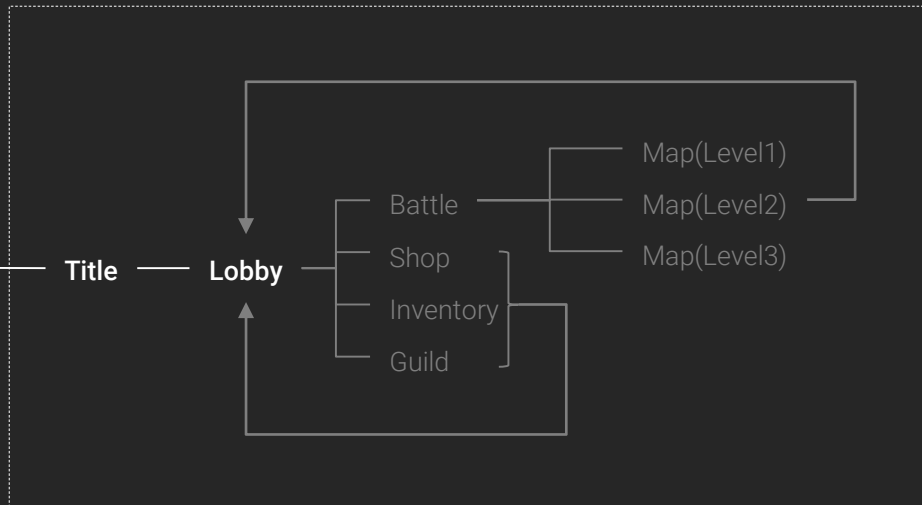
    stage = new StageBattle();
    stage_list.put(stage.getTitle(), stage);

    stage = new StageMap(Map.Level1);
    stage_list.put(stage.getTitle(), stage);
    stage = new StageMap(Map.Level2);
    stage_list.put(stage.getTitle(), stage);
    stage = new StageMap(Map.Level3);
    stage_list.put(stage.getTitle(), stage);
}
  
```

게임이 시작되게 되면, 추상화 클래스 Stage를 상속받는 여러 Stage 자식 클래스들을

Stage\_list라는 HashMap에 저장

그 후, Stage를 상속받는 Title로 넘어가고, '시작' 입력 시, 마찬가지로 Stage를 상속받는 Lobby로 이동

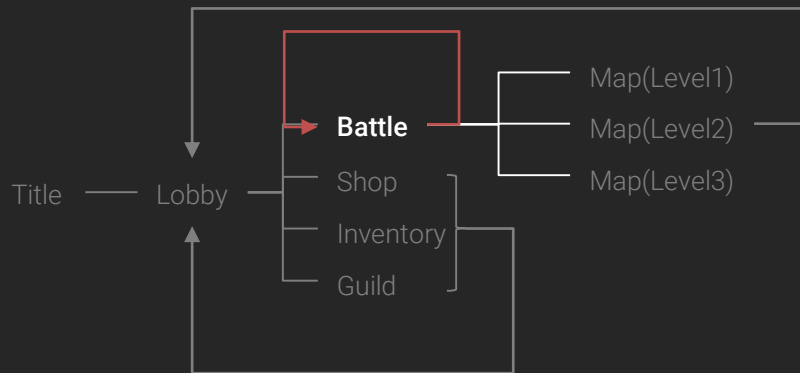


로그인 되었습니다. 환영합니다. [grape]님.  
 :::: TEXT RPG ::::  
 [시작]을 입력하세요.

```

curStage:TITLE
nextStage:LOBBY
----[ LOBBY ]----
[1]전투
[2]상점
[3]길드관리
[4]인벤토리
[5]저장
[0]종료
  
```

## JAVA Console RPG - Game Manager ( Battle )



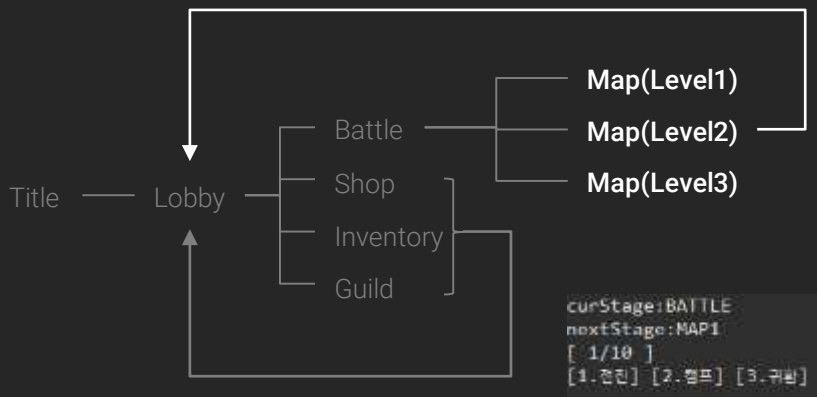
```
private void CheckLevelChangeMap(int sel) {  
    int userLevel = un.getAverageLevel(un.getPartyInGuild(GameManager.getUser().getGuild()));  
    if (sel >= GameManager.MAP_LEVEL1 && sel <= GameManager.MAP_LEVEL3) {  
        if (userLevel >= sel) {  
            GameManager.setNewStage("Map"+sel);  
        }  
        else {  
            System.out.println("도전 레벨에 미달입니다.");  
        }  
    }  
}
```

입장 레벨이 아닙니다.  
curStage: BATTLE  
nextStage: BATTLE

Lobby에서 Battle Stage 진입 시, Level에 의해 나누어진 Map  
Stage 리스트를 출력,

1. 출력된 Map을 선택하게 되면 User가 가지고 있는  
EmployableUnit의 평균 레벨과 같거나 낮다면 입장
2. 아닐 시, 메시지 출력 후 다시 Battle Stage로 루프

## JAVA Console RPG - Game Manager ( Map )



```
public boolean update() {
    init();

    boolean isRun = true;
    while (isRun) {
        isRun = selectPlayer();
        if (site == ENCOUNTER == 0) {
            isRun = isRun ? startBattle() : isRun;
        }
        if (site == SIZE)
            break;
    }
    if (site == SIZE && !m.checkUnitAlive(party)) {
        System.out.println("상대측은 모두 사망했습니다.");
        System.out.printf("보요로 적들을 모두 처치하셨습니다.\n", (level * 10000));
        GameManager.getManager().addGold(level * 10000);
    }

    System.out.println("다음로 귀환합니다.");
    GameManager.setNextStage("LOBBY");
    return true;
}
```

Map 입장 시, 현재 User의 Employable Unit 중 멤버변수 party(boolean)이 true인 Unit을 모아 party에 초기화

총 10 번의 이동 turn, 3, 6, 9 번 째에서 monster를 조우하여 전투를 시작

Map에 입장하는 순간을 기준으로 1번의 camping을 통해 전체 체력 회복 가능

1. 전투 중 모든 Unit의 curHp가 0이 되어 탐험에 실패하는 경우 Lobby Stage로 이동
2. 이동 중 귀환을 선택하는 경우 다시 Lobby Stage로 이동
3. 모든 전투를 마치고 승리하는 경우 성공 보상을 취한 뒤, 다시 Lobby Stage로 이동

## JAVA Console RPG - Game Manager ( Map )

### 전투

몬스터의 변환은 멤버번호로 선언한 monster class name 배열에서 Level을  
파라미터로 받고, 해당 Level을 배열의 index와 매칭하여 랜덤하게 class name을 받고  
해당 class의 몬스터를 생성한 뒤, ArrayList에 담고 반환

파티원과 몬스터의 Status 출력 후, 유저가 먼저 행동을 취하며 각 Unit의 curHp와, stun  
등 활동할 수 있는지의 여부를 체크 한 후, Action을 할 수 있는 상태인 Unit의 Action을  
선택

유저의 행동이 끝나고 각 몬스터의 curHp가 남아있고 stun. 상태가 아니라면  
Random값에 의해 attack 혹은 skill을 사용한다.

[호랑이] [1.공격] [실면술] → blind = true (1 turn 동안 스킬 사용 불가)

[호랑이] 기절중  
[토끼] [1.공격] [2.스킬] → stun = true (1 turn 동안 행동 불가)

```
private String[][] monsterClassName = {
    {"UnitBat", "UnitOrc", "UnitWolf"},
    {"UnitWerewolf", "UnitWerewolf", "UnitWerewolf"},
    {"UnitGiant", "UnitMimic", "UnitSalamander"}
};
```

INDEX(0) = Level 1  
INDEX(1) = Level 2  
INDEX(2) = Level 3

```
public ArrayList<Unit> getRandomMonsters(int level, int size) {
    mon_list.clear();
    int yIdx = level-1;
    for (int i = 0; i < size; i++) {
        int xIdx = GameManager.run.nextInt(monsterClassName[yIdx].length);
        try {
            Class<?> newClassName = Class.forName(path + monsterClassName[yIdx][xIdx]);
            Object obj = newClassName.getDeclaredConstructor().newInstance();
            mon_list.add((Unit)obj);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return mon_list;
}
```

처음과 조우했습니다. 전투를 시작합니다.

----- 파티원 -----

<사제> [호랑이] [레벨:1] [체력:1500/1500] [공격력:40] [방어력:6]

<전사> [토끼] [레벨:1] [체력:700/700] [공격력:120] [방어력:6]

<마법사> [고양이] [레벨:1] [체력:900/900] [공격력:120] [방어력:4]

<도적> [코끼리] [레벨:1] [체력:700/700] [공격력:120] [방어력:6]

----- 몬스터 -----

[오크] [레벨:1] [체력:500/500] [공격력:50] [방어력:7]

[오크] [레벨:1] [체력:500/500] [공격력:50] [방어력:7]

[늑대] [레벨:1] [체력:500/500] [공격력:60] [방어력:6]

-----

[호랑이] [1.공격] [2.스킬]

1

[호랑이]가 공격합니다.

[오크]가 113의 데미지를 입었습니다.

[토끼]가 94의 데미지를 입었습니다.

1

[토끼]가 기절합니다.

[오크]가 113의 데미지를 입었습니다.

[전사]가 113의 데미지를 입었습니다.

1

[호랑이]가 공격합니다.

[오크]가 113의 데미지를 입었습니다.

[코끼리]가 공격합니다.

[오크]가 113의 데미지를 입었습니다.

[오크]가 스킬을 사용합니다.

[토끼]가 94의 데미지를 입었습니다.

[토끼]가 기절에 걸렸습니다.

[오크]가 스킬을 사용합니다.

[코끼리]가 94의 데미지를 입었습니다.

[코끼리]가 기절에 걸렸습니다.

## JAVA Console RPG - Game Manager ( Map )

또한 User와 monster의 행동에 따라 전투가 계속 진행 될 지의 여부를

Boolean형으로 반환하여 전투의 진행 여부를 결정

```
private boolean actParty() {
    if (um.checkUnitsLive(party)) {
        for (Unit partyone : party) {
            if (um.checkUnitsLive(partyone) && um.checkUnitsLive(mon_list) && !partyone.isStun()) {
                while (true) {

                    System.out.printf("[%s] [1.공격] ", partyone.getName());
                    if (!partyone.isBlind()) {
                        System.out.println("[2.대응] ");
                    } else {
                        System.out.println("[3.대응] ");
                    }

                    int sel = GameManager.inputInt();
                    if (sel == 1) {
                        partyone.attack(um.randomLiveUnit(mon_list), partyone.getPow());
                        break;
                    } else if (sel == 2 && !partyone.isBlind()) {
                        if (partyone.getSkill() instanceof SkillHeal)
                            partyone.skill(um.randomLiveUnit(party), party, partyone.getLevel() * 100);
                        else
                            partyone.skill(um.randomLiveUnit(mon_list), mon_list, partyone.getPow());
                        break;
                    }
                }
            }
        }
    } else if (um.checkUnitsLive(partyone) && partyone.isStun()) {
        System.out.printf("[%s] %SS\n", partyone.getName());
    }
    partyone.setBlind(false);
    partyone.setStun(false);

    if (um.checkUnitsLive(mon_list))
        return false;
    return true;
}
```

```
boolean isRun = true;
while(isRun) {
    printAllUnit();
    isRun = isRun ? actParty() : isRun;
    isRun = isRun ? actMonster() : isRun;
}
```

```
private boolean actMonster() {
    if (um.checkUnitsLive(party)) {
        for (Unit mon : mon_list) {
            if (um.checkUnitsLive(mon) && um.checkUnitsLive(party) && !mon.isStun()) {
                int choice = GameManager.ran.nextInt(2);
                int skill = 0;

                if (choice == skill && !mon.isBlind()) {
                    System.out.printf("[%s]가 스킬을 사용함!\n", mon.getName());
                    if (mon.getSkill() instanceof SkillHeal)
                        mon.skill(um.randomLiveUnit(mon_list), mon_list, mon.getLevel() * 100);
                    else {
                        mon.skill(um.randomLiveUnit(party), party, mon.getPow());
                    }
                } else {
                    mon.attack(um.randomLiveUnit(party), mon.getPow());
                }
            } else if (um.checkUnitsLive(mon) && mon.isStun())
                System.out.printf("[%s] %SS\n", mon.getName());

            mon.setBlind(false);
            mon.setStun(false);

            if (!um.checkUnitsLive(party))
                return false;
        }
    }
    return true;
}
```

## JAVA Console RPG - Game Manager ( Map )

전투 종료 후, 승리 시 모든 Unit이 경험치 획득하며  
경험치를 경험치가 레벨 업 구간에 달성하는 지를 확인하고  
달성되는 경우 레벨을 올린다.

```
if (!um.checkUnitsLive(party)) {  
    System.out.println("길드원들 모두 전투가 불가능합니다.");  
    return false;  
} else if (!um.checkUnitsLive(mon_list)) {  
    System.out.println("전투에서 승리하셨습니다.");  
    um.addExp(level, party);  
    site++;  
}
```



```
public void addExp(int add) {  
    this.setExp(this.getExp() + add);  
    // 0 ~ 4 > 1lv  
    // 5 ~ 9 > 2lv  
    // 10 ~ 14 > 3lv  
    if (this.getExp() % ((8*this.getLevel()*getLevel()) == 0) {  
        this.levelUp();  
        System.out.printf("[%s] level up! [%d]\n", this.getName(), this.getLevel());  
    }  
}
```

## JAVA Console RPG - Game Manager ( Shop )

```
private ArrayList<Item> weapons; 현재 gold : 100020000
private ArrayList<Item> armors; [1] 무기
private ArrayList<Item> rings; [2] 방어구

if (sel == 1) buyItem(weapons); [3] 반지
else if (sel == 2) buyItem(armors); [4] 장비판매
else if (sel == 3) buyItem(rings); [0] 뒤로가기
else if (sel == 4) sellItem();

private void buyItem(ArrayList<? extends Item> list) {
    printItemByNumber(list);
    int index = GameManager.getInputInt() - 1;

    if (index >= 0 && index < list.size() && list.get(index).getPrice() <= GameManager.getUser().getGold()) {
        System.out.printf("[%d] %d %d %s", list.get(index).getPrice(), list.get(index).getName());
        GameManager.getUser().getInventory().add(list.get(index).clone());
        GameManager.getUser().subGold(list.get(index).getPrice());
    }
}
```

### 구매

ArrayList 배열을 통해 무기, 방어구, 반지를 나누고 해당 메뉴 선택 시, 리스트를 출력,  
후에 입력된 숫자가 출력된 리스트의 범위를 넘지 않고, 구매 가능한 재화를 들고 있다면  
구매되며 해당 **아이템**은 **clone()**을 통해 **User**의 **Inventory(Item ArrayList)**에  
추가된다.





## JAVA Console RPG - Game Manager ( Inventory )

```
curStage:LOBBY
nextStage:INVENTORY
---[ INVENTORY ]---
Gold : 100016500
-----
[1]소지품
[2]장비장착
[3]장비하체
[4]장비강화
[0]뒤로가기
```

### 소지품

1. ArrayList<Item>에 아이템이 존재하는 경우 출력
2. 존재하지 않는 경우 "없음" 출력

### 장비장착

1. User가 가진 Guild(ArrayList<Employabe>)의 unit들을 번호와 함께 출력
2. 출력된 Unit 번호를 입력 시, 해당 Unit의 장비 상태와 소지품 출력
3. 출력된 소지품 번호 입력 시, 해당 장비 장착 후 Unit의 장비 상태 출력
4. 후에 상태 출력 시, 해당 장비가 가진 능력치 만큼 + 되어 표시

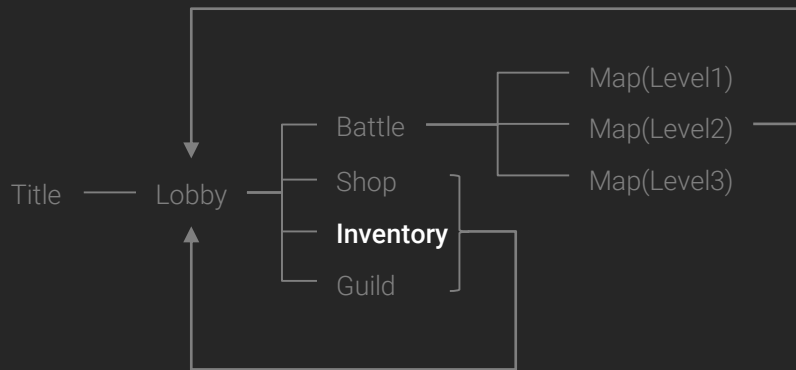
- Weapon > pow
- Armor > def
- Ring > maxHp

```
-----길드원-----
번호(1) <사제> [포함미] [레벨:2] [체력:1600/1600] [공격력:50] [방어력:11] [피리움]
-----
번호(2) <전사> [포기] [레벨:2] [체력:800/800] [공격력:130] [방어력:11] [파리움]
-----
번호(3) <마법사> [고양미] [레벨:2] [체력:1000/1000] [공격력:130] [방어력:9] [피리움]
-----
번호(4) <도둑> [프리티] [레벨:2] [체력:800/800] [공격력:130] [방어력:11] [파리움]
```

```
사제/전사/마법사/도둑
사제: 1명
전사: 1명
마법사: 1명
도둑: 1명
-----
길드원: 4명
사제: 1명
전사: 1명
마법사: 1명
도둑: 1명
```

```
if (item_list.get(index).getLevel() > unit.getLevel()) {
    System.out.println("장비를 + 할 수 없습니다.");
} else {
    Item item = item_list.get(index);
    if (item instanceof Weapon) {
        if (unit.weaponIsEmpty()) {
            item_list.add(unit.getWeapon());
        }
        unit.setWeapon((Weapon)item);
        item_list.remove(index);
    } else if (item instanceof Armor) {
        if (unit.armorIsEmpty()) {
            item_list.add(unit.getArmor());
        }
        unit.setArmor((Armor)item);
        item_list.remove(index);
    } else if (item instanceof Ring) {
        if (unit.ringIsEmpty()) {
            item_list.add(unit.getRing());
        }
        unit.setRing((Ring)item);
        item_list.remove(index);
    }
    unit.printEquipment();
}
```

```
<사제> [포함미] [레벨:2] [체력:1600/1600] [공격력:50+5] [방어력:11] [피리움]
```



## JAVA Console RPG - Game Manager ( Inventory )

```
System.out.println("1)무기 (2)장비 (3)판매");
System.out.println("종료");
int num = GameManager.getInput();
if (num == 1 && num != 3) {
    Item temp = null;

    if (num == 1 && Unit.isWeaponEmpty()) {
        temp = Unit.getWeapon();
        Unit.setWeapon(null);
    } else if (num == 2 && Unit.isArmorEmpty()) {
        temp = Unit.getArmor();
        Unit.setArmor(null);
    } else if (num == 3 && Unit.isRingEmpty()) {
        temp = Unit.getRing();
        Unit.setRing(null);
    }

    if (temp != null) {
        Item_list.add(temp);
        System.out.println("판매 성공");
        temp=null;
    }

    Unit.printStatus();
}
```

```
***** 판매물 *****
번호(1) <무기> [레벨:1] [레프:1500/1500] [공격력:40~5] [방어력:6] [속도증]
번호(2) <장비> [레벨:1] [레프:700/700] [공격력:120] [방어력:6] [속도증]
번호(3) <장비> [레벨:1] [레프:900/900] [공격력:120] [방어력:4] [속도증]
*****
번호(4) <무기> [레벨:1] [레프:700/700] [공격력:120] [방어력:6] [속도증]
*****
판매할 번호:5
[무기] 없음
[장비] 없음
[판매] 없음

<1>무기 <2>장비 <3>판매
현재 보유:1
무기:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000,1001,1002,1003,1004,1005,1006,1007,1008,1009,1010,1011,1012,1013,1014,1015,1016,1017,1018,1019,1020,1021,1022,1023,1024,1025,1026,1027,1028,1029,1030,1031,1032,1033,1034,1035,1036,1037,1038,1039,1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055,1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071,1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087,1088,1089,1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103,1104,1105,1106,1107,1108,1109,1110,1111,1112,1113,1114,1115,1116,1117,1118,1119,1120,1121,1122,1123,1124,1125,1126,1127,1128,1129,1130,1131,1132,1133,1134,1135,1136,1137,1138,1139,1140,1141,1142,1143,1144,1145,1146,1147,1148,1149,1150,1151,1152,1153,1154,1155,1156,1157,1158,1159,1160,1161,1162,1163,1164,1165,1166,1167,1168,1169,1170,1171,1172,1173,1174,1175,1176,1177,1178,1179,1180,1181,1182,1183,1184,1185,1186,1187,1188,1189,1190,1191,1192,1193,1194,1195,1196,1197,1198,1199,1200,1201,1202,1203,1204,1205,1206,1207,1208,1209,1210,1211,1212,1213,1214,1215,1216,1217,1218,1219,1220,1221,1222,1223,1224,1225,1226,1227,1228,1229,1230,1231,1232,1233,1234,1235,1236,1237,1238,1239,1240,1241,1242,1243,1244,1245,1246,1247,1248,1249,1250,1251,1252,1253,1254,1255,1256,1257,1258,1259,1260,1261,1262,1263,1264,1265,1266,1267,1268,1269,1270,1271,1272,1273,1274,1275,1276,1277,1278,1279,1280,1281,1282,1283,1284,1285,1286,1287,1288,1289,1290,1291,1292,1293,1294,1295,1296,1297,1298,1299,1300,1301,1302,1303,1304,1305,1306,1307,1308,1309,1310,1311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321,1322,1323,1324,1325,1326,1327,1328,1329,1330,1331,1332,1333,1334,1335,1336,1337,1338,1339,1340,1341,1342,1343,1344,1345,1346,1347,1348,1349,1350,1351,1352,1353,1354,1355,1356,1357,1358,1359,1360,1361,1362,1363,1364,1365,1366,1367,1368,1369,1370,1371,1372,1373,1374,1375,1376,1377,1378,1379,1380,1381,1382,1383,1384,1385,1386,1387,1388,1389,1390,1391,1392,1393,1394,1395,1396,1397,1398,1399,1400,1401,1402,1403,1404,1405,1406,1407,1408,1409,1410,1411,1412,1413,1414,1415,1416,1417,1418,1419,1420,1421,1422,1423,1424,1425,1426,1427,1428,1429,1430,1431,1432,1433,1434,1435,1436,1437,1438,1439,1440,1441,1442,1443,1444,1445,1446,1447,1448,1449,1450,1451,1452,1453,1454,1455,1456,1457,1458,1459,1460,1461,1462,1463,1464,1465,1466,1467,1468,1469,1470,1471,1472,1473,1474,1475,1476,1477,1478,1479,1480,1481,1482,1483,1484,1485,1486,1487,1488,1489,1490,1491,1492,1493,1494,1495,1496,1497,1498,1499,1500,1501,1502,1503,1504,1505,1506,1507,1508,1509,1510,1511,1512,1513,1514,1515,1516,1517,1518,1519,1520,1521,1522,1523,1524,1525,1526,1527,1528,1529,1530,1531,1532,1533,1534,1535,1536,1537,1538,1539,1540,1541,1542,1543,1544,1545,1546,1547,1548,1549,1550,1551,1552,1553,1554,1555,1556,1557,1558,1559,1560,1561,1562,1563,1564,1565,1566,1567,1568,1569,1570,1571,1572,1573,1574,1575,1576,1577,1578,1579,1580,1581,1582,1583,1584,1585,1586,1587,1588,1589,1590,1591,1592,1593,1594,1595,1596,1597,1598,1599,1600,1601,1602,1603,1604,1605,1606,1607,1608,1609,1610,1611,1612,1613,1614,1615,1616,1617,1618,1619,1620,1621,1622,1623,1624,1625,1626,1627,1628,1629,1630,1631,1632,1633,1634,1635,1636,1637,1638,1639,1640,1641,1642,1643,1644,1645,1646,1647,1648,1649,1650,1651,1652,1653,1654,1655,1656,1657,1658,1659,1660,1661,1662,1663,1664,1665,1666,1667,1668,1669,1670,1671,1672,1673,1674,1675,1676,1677,1678,1679,1680,1681,1682,1683,1684,1685,1686,1687,1688,1689,1690,1691,1692,1693,1694,1695,1696,1697,1698,1699,1700,1701,1702,1703,1704,1705,1706,1707,1708,1709,1710,1711,1712,1713,1714,1715,1716,1717,1718,1719,1720,1721,1722,1723,1724,1725,1726,1727,1728,1729,1730,1731,1732,1733,1734,1735,1736,1737,1738,1739,1740,1741,1742,1743,1744,1745,1746,1747,1748,1749,1750,1751,1752,1753,1754,1755,1756,1757,1758,1759,1760,1761,1762,1763,1764,1765,1766,1767,1768,1769,1770,1771,1772,1773,1774,1775,1776,1777,1778,1779,1780,1781,1782,1783,1784,1785,1786,1787,1788,1789,1790,1791,1792,1793,1794,1795,1796,1797,1798,1799,1800,1801,1802,1803,1804,1805,1806,1807,1808,1809,1810,1811,1812,1813,1814,1815,1816,1817,1818,1819,1820,1821,1822,1823,1824,1825,1826,1827,1828,1829,1830,1831,1832,1833,1834,1835,1836,1837,1838,1839,1840,1841,1842,1843,1844,1845,1846,1847,1848,1849,1850,1851,1852,1853,1854,1855,1856,1857,1858,1859,1860,1861,1862,1863,1864,1865,1866,1867,1868,1869,1870,1871,1872,1873,1874,1875,1876,1877,1878,1879,1880,1881,1882,1883,1884,1885,1886,1887,1888,1889,1890,1891,1892,1893,1894,1895,1896,1897,1898,1899,1900,1901,1902,1903,1904,1905,1906,1907,1908,1909,1910,1911,1912,1913,1914,1915,1916,1917,1918,1919,1920,1921,1922,1923,1924,1925,1926,1927,1928,1929,1930,1931,1932,1933,1934,1935,1936,1937,1938,1939,1940,1941,1942,1943,1944,1945,1946,1947,1948,1949,1950,1951,1952,1953,1954,1955,1956,1957,1958,1959,1960,1961,1962,1963,1964,1965,1966,1967,1968,1969,1970,1971,1972,1973,1974,1975,1976,1977,1978,1979,1980,1981,1982,1983,1984,1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021,2022,2023,2024,2025,2026,2027,2028,2029,2030,2031,2032,2033,2034,2035,2036,2037,2038,2039,2040,2041,2042,2043,2044,2045,2046,2047,2048,2049,2050,2051,2052,2053,2054,2055,2056,2057,2058,2059,2060,2061,2062,2063,2064,2065,2066,2067,2068,2069,2070,2071,2072,2073,2074,2075,2076,2077,2078,2079,2080,2081,2082,2083,2084,2085,2086,2087,2088,2089,2090,2091,2092,2093,2094,2095,2096,2097,2098,2099,2100,2101,2102,2103,2104,2105,2106,2107,2108,2109,2110,2111,2112,2113,2114,2115,2116,2117,2118,2119,2120,2121,2122,2123,2124,2125,2126,2127,2128,2129,2130,2131,2132,2133,2134,2135,2136,2137,2138,2139,2140,2141,2142,2143,2144,2145,2146,2147,2148,2149,2150,2151,2152,2153,2154,2155,2156,2157,2158,2159,2160,2161,2162,2163,2164,2165,2166,2167,2168,2169,2170,2171,2172,2173,2174,2175,2176,2177,2178,2179,2180,2181,2182,2183,2184,2185,2186,2187,2188,2189,2190,2191,2192,2193,2194,2195,2196,2197,2198,2199,2200,2201,2202,2203,2204,2205,2206,2207,2208,2209,2210,2211,2212,2213,2214,2215,2216,2217,2218,2219,2220,2221,2222,2223,2224,2225,2226,2227,2228,2229,2230,2231,2232,2233,2234,2235,2236,2237,2238,2239,2240,2241,2242,2243,2244,2245,2246,2247,2248,2249,2250,2251,2252,2253,2254,2255,2256,2257,2258,2259,2260,2261,2262,2263,2264,2265,2266,2267,2268,2269,2270,2271,2272,2273,2274,2275,2276,2277,2278,2279,2280,2281,2282,2283,2284,2285,2286,2287,2288,2289,2290,2291,2292,2293,2294,2295,2296,2297,2298,2299,2300,2301,2302,2303,2304,2305,2306,2307,2308,2309,2310,2311,2312,2313,2314,2315,2316,2317,2318,2319,2320,2321,2322,2323,2324,2325,2326,2327,2328,2329,2330,2331,2332,2333,2334,2335,2336,2337,2338,2339,2340,2341,2342,2343,2344,2345,2346,2347,2348,2349,2350,2351,2352,2353,2354,2355,2356,2357,2358,2359,2360,2361,2362,2363,2364,2365,2366,2367,2368,2369,2370,2371,2372,2373,2374,2375,2376,2377,2378,2379,2380,2381,2382,2383,2384,2385,2386,2387,2388,2389,2390,2391,2392,2393,2394,2395,2396,2397,2398,2399,2400,2401,2402,2403,2404,2405,2406,2407,2408,2409,2410,2411,2412,2413,2414,2415,2416,2417,2418,2419,2420,2421,2422,2423,2424,2425,2426,2427,2428,2429,2430,2431,2432,2433,2434,2435,2436,2437,2438,2439,2440,2441,2442,2443,2444,2445,2446,2447,2448,2449,2450,2451,2452,2453,2454,2455,2456,2457,2458,2459,2460,2461,2462,2463,2464,2465,2466,2467,2468,2469,2470,2471,2472,2473,2474,2475,2476,2477,2478,2479,2480,2481,2482,2483,2484,2485,2486,2487,2488,2489,2490,2491,2492,2493,2494,2495,2496,2497,2498,2499,2500,2501,2502,2503,2504,2505,2506,2507,2508,2509,2510,2511,2512,2513,2514,2515,2516,2517,2518,2519,2520,2521,2522,2523,2524,2525,2526,2527,2528,2529,2530,2531,2532,2533,2534,2535,2536,2537,2538,2539,2540,2541,2542,2543,2544,2545,2546,2547,2548,2549,2550,2551,2552,2553,2554,2555,2556,2557,2558,2559,2560,2561,2562,2563,2564,2565,2566,2567,2568,2569,2570,2571,2572,2573,2574,2575,2576,2577,2578,2579,2580,2581,2582,
```

## JAVA Console RPG - Game Manager ( Guild )

```
17 (guildIdx >= 0 && guildIdx < guildMembers.size()) && !guildMembers.get(guildIdx).isInParty()) {
    party.get(partyIdx).setInParty(false);
    guildMembers.get(guildIdx).setInParty(true);
    party.set(partyIdx, guildMembers.get(guildIdx));
    System.out.println("교체가 완료되었습니다.");
}
```

## 파티원교체

inParty = true인 파티원 출력 후, 모든 길드원 출력

교체를 위해 길드원의 inParty 여부를 확인 후,

```
int avgLv = um.getAverageLevel(guildMembers);
int price = 1000 * avgLv;
```

## 용병모집

비용은 고용 레벨과 비례하며, 고용되는 용병의 레벨은

User의 Unit들의 평균 레벨에 의해 결정

**HTML & CSS & JS D-Day**

## HTML & CSS & JS D-Day

프로젝트 명 : 나만의 디데이 계산기

개발기간 : 2022.06.29 - 2022.07.01 (2일)

개발인원 : 1명

요약 : D-Day 계산기

<https://bit.ly/3yuA1Xh>

### 개발환경

- OS : Windows10
  - Development Tool : Visual Studio Code
  - Language : HTML & CSS & JavaScript
- 
- JavaScript를 통한 날짜 계산
  - Html 작성 및 CSS를 통한 비주얼 요소 추가
  - CSS media를 통한 반응형



**HTML & CSS & JS** BOOK SEARCH

## HTML & CSS & JS & JQuery BOOK SEARCH

프로젝트 명 : **BOOK SEARCH**

개발기간 : 2022.07.02 – 2022.07.04 (2일)

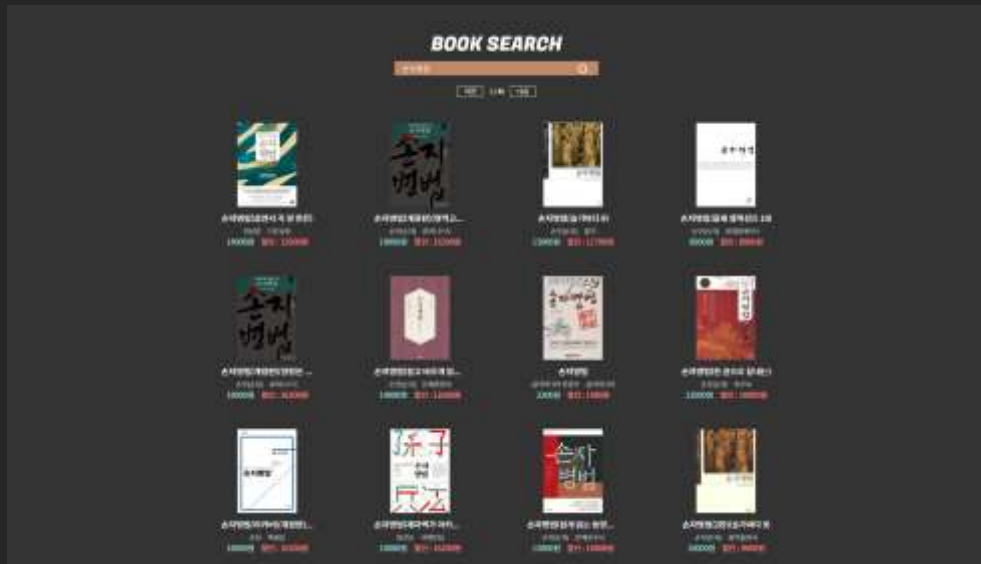
개발인원 : 1명

요약 : 책 검색

<https://bit.ly/3zQiDhZ>

개발환경

- OS : Windows10
- Development Tool : Visual Studio Code
- Language : HTML & CSS & JavaScript & JQuery
- Jquery의 Ajax를 통해 **Daum Kakao API** 사용
- JavaScript를 통한 불러온 **Data** 페이징 처리



**HTML & CSS & JS** CRAZY ACADE



## HTML & CSS & JS CRAZY ACADE

프로젝트 명 : **CRAZY ACADE**

개발기간 : 2022.07.02 - 2022.07.03 (1일)

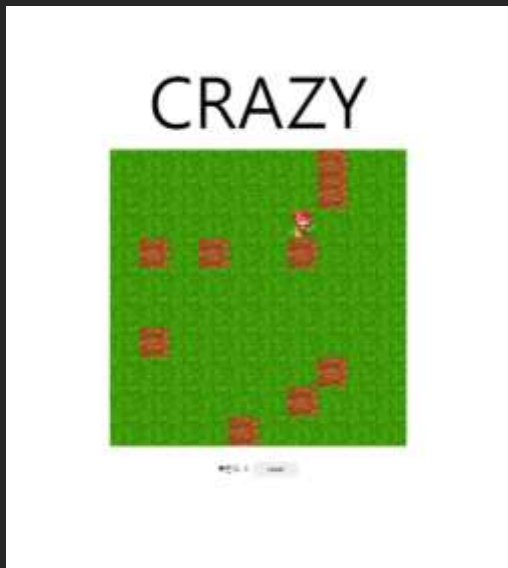
개발인원 : 1명

요약 : 간단한 아케이드 게임

<https://bit.ly/3J1pxD>

### 개발환경

- OS : Windows10
  - Development Tool : Visual Studio Code
  - Language : HTML & CSS & JavaScript
- 
- 키보드 ADWS와 화살표를 입력 시, 사방으로 이동 가능
  - Q 설치, E 폭파
  - 반응형 구현 완료. 작은 화면의 기기, 모바일로 접속 시, 터치로 이동 가능
  - 하단의 버튼을 통해 폭탄을 설치하고 폭파



# JSP MVC2 SB Project

## JSP SB Project

프로젝트 명 : **SB Project**

개발기간 : 2022.07.08 - 2022.07.24 (16일)

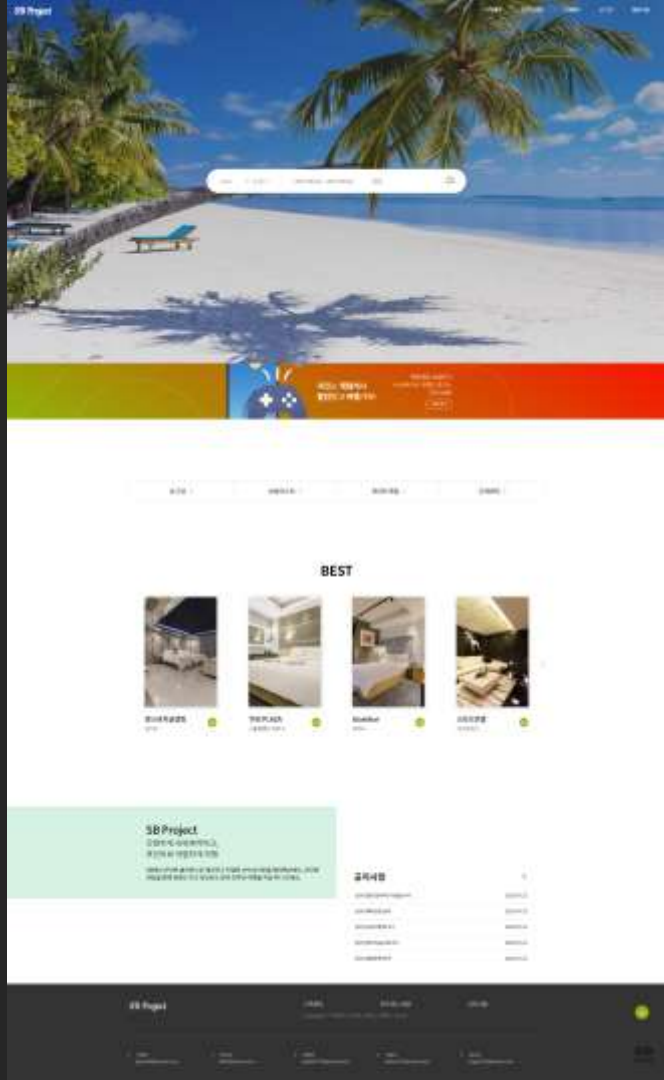
개발인원 : 5명

요약 : 숙박 중개 사이트

### 개발환경

- OS : Windows10
- Development Tool : Eclipse, Apache-tomcat-9.0.65, AWS RDS
- Language : JSP & MySQL & HTML & CSS & JavaScript & JQuery

- 사용자와 사업장 이용자 구분
- 사용자는 숙박 예약 기능, 사업장은 숙박 업소 등록 기능 제공
- 작업 효율을 위한 기본 프레임, function 제공



ISD 기보 형태를 제공하고 자다한 고틀 기보는 미키

함수(function)로 선언하여 작업 효율 증진

#wrap 내부에 class name이 title\_name인 <h3> 태그를

추가하고 내부에 text를 입력하는 경우 자동 title 수정

Html에 class name이 topBtn인 <button> 태그를

추가하기만 해도 top rolling 버튼 생성

# JSP SB Project

```

1  <%page import="green_project.LikeListDAO"%>
2  <%page import="green_project.ReservationDAO"%>
3  <%page import="green_project.BoardDAO"%>
4  <%page import="green_project.ItemDAO"%>
5  <%page import="green_project.UserDAO"%>
6  <%page language="java" contentType="text/html; charset=UTF-8"
7    pageEncoding="UTF-8"%>
8  <% UserDAO dao = UserDAO.getInstance(); %>
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13 <meta charset="UTF-8">
14 <link rel="stylesheet" href="/green_project/css/common.css">
15 <link rel="stylesheet" href="/green_project/css/reset.css">
16 <title></title>
17 <script
18   src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
19 </head>
20 <body>
21 <%
22   String log = dao.getLog();
23
24   UserDAO dao = UserDAO.getInstance();
25   ItemDAO idao = ItemDAO.getInstance();
26   BoardDAO bdao = BoardDAO.getInstance();
27   ReservationDAO rdao = ReservationDAO.getInstance();
28   LikeListDAO ldao = LikeListDAO.getInstance();
29 %>

```

## 공통의 meta, css, script, library

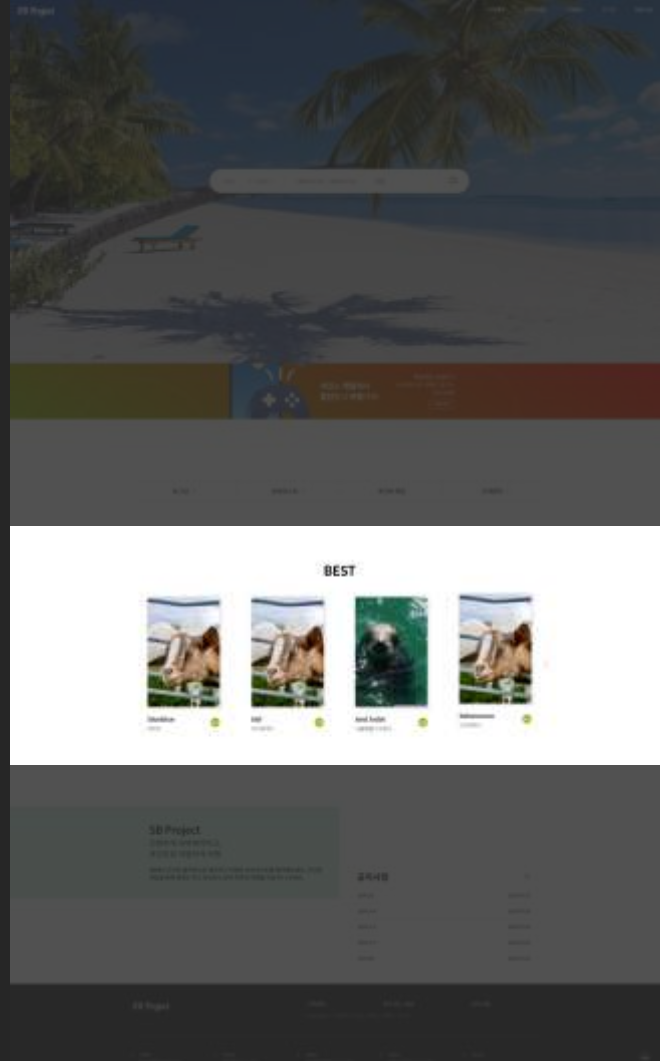
제작 후 팀원에게 제공한 sub frame만으로 공통으로 적용되는 css와 js를  
적용시켜야 할 필요가 없으며, 작업 시 바로 library사용

## JSP SB Project

```
<div class="recommend_item_wrap">
<div>BEST</div>
<ul>
<li>
ArrayList<ItemDTO> bestlist = idan.getBestlikeItem();
for (int i = 0; i < bestlist.size(); i++) {
ItemDTO room = bestlist.get(i);
String thumbel = room.getThumbnail();
if (thumbel.equals("")) {
thumbel = "resources/empty_room_img.png";
}
}
<li><li>
href="/arwen/project/ServiceServlet?command=roomInfo&grade=<li>room.getCode()<li>";
<span class="img_wrap">room.getName()<li> @<li>";</span>
<div class="recommend_item_info">
<p class="item_title"><li>room.getName()<li></p>
<p class="item_address"><li>room.getAddress()<li></p>
<span class="grade"><li>room.getRate()<li></span>
</div>
</li></li>
</ul>
<div>
<span class="prev_btn"></span> <span class="next_btn"></span>
</div>
</div>
```

### Like가 가장 많은 상품 노출

DB에서 가져오는 사이즈가 의도한 최대 개수보다 적다면 해당 상품이 존재하는 만큼만 노출되게 만들어 빈 상품리스트가 생기지 않도록 설계



[illegible][illegible]

The image shows a web design mockup for a company named "SD Project".

- Header:** Features a large background image of a tropical beach with palm trees and a blue lounge chair. A white search bar is centered at the top.
- Navigation Bar:** Located below the header, it includes links for "HOME", "ABOUT US", "SERVICES", "CONTACT", and "FAQ".
- Banner Section:** A horizontal band with a dark red background. It contains a blue circular icon with a white plus sign and text in Korean: "SD Project" and "SD Project".
- Main Content Area:** Displays four service cards arranged horizontally. Each card has a small image of a cow and a title in Korean: "SD Project", "SD Project", "SD Project", and "SD Project".
- Footer/Bottom Section:**
  - On the left, a light green box contains the text "SD Project" followed by a paragraph in Korean.
  - On the right, a vertical list of services or projects is shown, each with a small icon and a title in Korean.

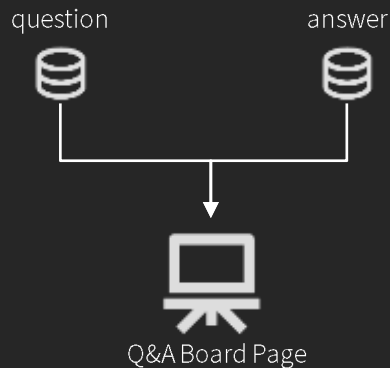
List Page → Write(insert board) Action

View Page → Delete Action

Write Page → Write(insert board) Action  
Update Action

### 공통된 Page를 사용하는 두 개의 Action

따라서 수정과 작성이라는 같은 Action을 취할 때 구분이 필요



### 두 개의 Board (Question-Answer Board)

게시판은 하나지만, 사용되는 Board는 두 개로 이용자가 질문하는 Question Board와 관리자가 답변하는 Answer Board. 두 개의 DB Table을 연동.



# JSP SB Project

# Board Page

## 페이지 기능을 구현.

최대 사이즈를 정하고 해당 사이즈 만큼만 보여주며 초과된 board는 다음 페이지로 넘어 갈 수 있는 버튼을 제공, list page에 parameter “page”를 넘겨 다시 해당 Page의 값을 기준으로 노출될 인덱스의 범위를 설정



1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 26

또한 페이지 이동 버튼 사이에 현재 페이지 기준 2페이지를  
바로 이동 할 수 있는 리스트를 출력하여 누르는 즉시 해당  
페이지로 바로 이동 할 수 있게 구현

```
<%
if (request.getParameter("no") == null) {
    %>
    <script>window.history.back();</script>
    <%
}

int no = Integer.parseInt(request.getParameter("no"));

QuestionDTO question = (QuestionDTO)bdao.getBoard("Question", no);
String title = question.getTitle();
String id = question.getId();
String contents = question.getContents().replace("<br>", "\n\n");
Date written = question.getWritten();
%>
```

### 예외처리와 replace를 통한 contents계행처리

백슬래시(\)를 통해 입력되는 값 중 계행("\n")을 태그 <br>로 변환하여 DB에 저장.

<br> 태그로 변환하여 입력하였기 때문에 추후 해당 Data를 노출할 때 별도의 replace 없이 바로 append로 html에 확장하여 계행 상태를 노출 할 수 있다.

### 버튼들의 노출 예외처리

대부분의 버튼들은 button태그를 통해 만들어졌으며, 삭제의 경우 question과 answer의 구분이 필요하기에 여러 예외처리를 통해 저장되는 DB의 구분을 결정.

```
<div class="view_btn_group">
    <%
    if (log != null) {
        %>
        <%
        if (log.equals(id)) { 비로그인
            %>
            <button class="modify_btn"
                onclick="location.href='/green_project/qboard/qboardwrite.jsp?no=<no>&type=Question';">수정하기</button>
            %>
        }
        if (log.equals(id) || dao.getAuth(log) == 0) { 관리자가 아니며 해당 게시글의 작성자 또는 관리자
            %>
            <form action="/green_project/serviceservlet" method="post">
                <input type="hidden" name="command" value="deleteQ" ><input
                    type="hidden" name="type" value="Question"> <input
                    type="hidden" name="no" value="<no>" > <input
                    type="submit" class="delete_btn" value="삭제" id="delete_btn">
            </form>
            %>
        }
        if (dao.getAuth(log) == 0) { 관리자
            %>
            <button class="answer_btn"
                onclick="location.href='/green_project/qboard/qboardwrite.jsp?questionno=<no>&type=Answer';">답변달기</button>
            %>
        }
        %>
        <%
        <button class="back_btn"
            onclick="window.history.back();">이전게시판</button>
        %>
    }
</div>
```

## JSP SB Project

```
<%
String title = "";
String contents = "";
int no = -1;
String type = "Answer";
int questionno = -1;

// no를 받아옴
if(request.getParameter("no") != null) {
    no = Integer.parseInt(request.getParameter("no"));
}

// type 받기
if(request.getParameter("type") != null) {
    type = request.getParameter("type");
}

if(request.getParameter("no") != null) {
    Board question = bdao.getBoard(type, no);
    title = question.getTitle();
    contents = question.getContents().replace("<br>", "\n");
}

if (request.getParameter("questionno") != null) {
    questionno = Integer.parseInt(request.getParameter("questionno"));
}

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

```
if(request.getParameter("no") != null && request.getParameter("questionno") == null) {
    %>
    <input type="hidden" name="command" value="updateQuestion">
    <input type="hidden" name="type" value="<%=type%>">
    <input type="hidden" name="no" value="<%=no%>">
    <%
} else {
    if (questionno != -1) {
        %>
        <input type="hidden" name="type" value="<%=type%>">
        <input type="hidden" name="questionno" value="<%=questionno%>">
        <%
    }
    %>
    <input type="hidden" name="command" value="writeQuestion">
    <%
}
%>
```

파라미터를 통한 Question과 Answer,  
Update와 Write Action의 구분

여러 파라미터들을 통해 존재 여부와 type 자체를 String으로 받아 받아지는 값과  
파라미터명에 따라 해당 command를 바꿀 수 있도록 설정



### 특정 구간 클릭 시, Action 동작

모바일의 터치를 통한 조작과 유사하게 동작을 하는 버튼 영역을 넓게 하여 해당 부분을 클릭하기만 해도 사용자가 의도한 동작을 할 수 있도록 구현

