

```
public class CharacterManager {
```

3 5 ^ v

```
// 왜 CharacterManager와 SelectedCharacter를 클래스 나눠야 하는 이유가 뭘까..?
```

```
// -> ArrayList가 다른 클래스 타입이 필요해서?
```

```
// 파티할 멤버 선택하기.
```

```
// 파티할 멤버를 어디에 담을 것 인가?
```

```
// ArrayList에 담을 것이다.
```

```
// 배열을 사용하지 않고 ArrayList를 사용하는 이유는 뭘까?
```

```
// 다른 타입이 클래스라서?
```

```
// 근데 안에 Object 타입으로 선택해서 객체화된 객체를 적여야하는거 아니가?
```

```
// ArrayList가 다른 타입의 클래스를 만들고
```

```
// 그 클래스의 매개변수를 Object로 하면 된다. |
```

```
public SelectedCharacter(Integer selectedNum, Object character){
```

```
    this.selectedNum = selectedNum;
```

```
    this.character = character;
```

```
}
```

1 related problem

```
public void chooseMember () {  
    int cnt = 0;  
    System.out.println("1번: 기사, 2번: 워자드, 3번: 스나이퍼,  
  
    for (int i = 0; i < 3; i++) {  
        System.out.print("번호를 입력하세요: ");  
        int num = scan.nextInt();  
  
        switch (num) {  
            case KNIGHT:  
                raidMember.add(kni);  
                memberCheckList.add(KNIGHT);  
                break;  
            case WIZARD:  
                raidMember.add(wiz);  
                memberCheckList.add(WIZARD);  
                break;  
            case SNIPER:  
                raidMember.add(sni);  
                memberCheckList.add(SNIPER);  
                break;  
            case HOLYKING:  
                raidMember.add(hk);  
                memberCheckList.add(HOLYKING);  
                break;  
            case ASSASSIN:
```

1 related problem

```
public void chooseMember () {  
    int cnt = 0;  
    System.out.println("1번: 기사, 2번: 워자드, 3번: 스나이퍼, 4번:  
  
    while(cnt < 3){  
        System.out.print("번호를 입력 >>");  
        int num = scan.nextInt();  
  
        if(num >= 6 || num <= 0)  
            continue;  
  
        cnt++;  
  
        // 일단 멤버를 선택하는걸 먼저 완성.  
    }
```

이제 멤버를 선택했으니 ArrayList에 값을 입력해줘야함.

ArrayList에 값을 입력할 수있는 메소드만들기.

```
public void procUserInput (int num) { // 선택된 직업을 넘겨 받음.  
    SelectedCharacter sc;  
    // SelectedCharacter(num, obj)  
    switch (num) {  
        case CharacterNumber.KNIGHT:  
            Knight kni = new Knight();  
  
            sc = new SelectedCharacter(CharacterNumber.KNIGHT, kni);  
            member.add(sc);  
            break;
```

```
public class RolePlayingGame extends MacroSet {  
  
    CharacterManager cm;
```

```
public void chooseMember(){  
    System.out.println("멤버를 선택하세요.");  
    cm.chooseMember();  
}
```

여기서 생각이 멈췄습니다..

여기서 MonsterManager을 만들어야 할까요..?

MonsterManager에서 하는일은 보스를 관리.

아직 fieldMonster을 안만들었는데 계속 해도 될까?

먼저 클래스를 만들어두고 fieldMonster을 추가해주는 형식으로 해볼까..

```
System.out.printf("%d 턴입니다\n", turnCnt++);  
fenryl.raidTurnStart(raidMember, memberCheckList);  
Thread.sleep( millis: 2000);  
}
```

파티원 참여한 직업이 돌아가면서 Fenryl을 때린다.

이걸 bm에서 어떻게 관리를 해야할까...

```
public void raidTurnStart(){  
    switch (sc.getSelectedNum()){  
        case MonsterNumber.FENRYL:  
            ((Fenryl)sc.getCharacter()).raidTurnStart();  
    }  
}
```

여기까지는 알겠다. Fenryl을 선택해야하고 그 캐릭터를 공격해야하니.

저 위에 raidMember와 memberCheckList는 CharacterManager에 있는 ArrayList에 있다. -> cm을 보낸다.

여기서 또 어떻게 해야할까..

```
public void raidTurnStart (CharacterManager cm) {  
    SelectedCharacter sc;  
  
    for (int i = 0; i < cm.memberSize(); i++) {  
        switch (check.get(i)) {  
            case KNIGHT:  
                hp -= ((Knight)member.get(i)).qSkill(obj: this);  
                break;  
        }  
    }  
}
```

```
public SelectedCharacter(Integer selectedNum, Object character){  
    this.selectedNum = selectedNum;  
    this.character = character;  
}  
  
private ArrayList<SelectedCharacter> member;
```

```
public CharacterManager(){  
    // 생성자, 반환값이 없죠~  
    member = new ArrayList<>();  
}
```

member에 selectedNum, character 두개 다 들어있는데 어떻게 하나씩만 꺼낼 수 있을까?
그걸 ArrayList에서 가져온후 get SelectedNum과 getCharacter으로 가져올 수 있다.

getMemberArrayList를 통해서 cm을 다시 불러와 sc를 만든다.

```
public SelectedCharacter (Integer selectedNum, Object character)
    this.selectedNum = selectedNum;
    this.character = character;
}
```

```
sc = cm.getMemberArrayList().get(i);
```

```
public ArrayList<SelectedCharacter> getMemberArrayList() {
    return member;
}
```

```
public class DamageCalcRequestObject {  
    // 18주차는 fenny만 공격했는데  
    // 그걸 선택해준 몬스터로 알아서 정하기 위해서  
    |
```

```
SelectedCharacter monsterSc = new SelectedCharacter(MonsterNumber.FENRYL, character;  
for (int i = 0; i < cm.memberSize(); i++) {
```

```
    sc = cm.getMemberArrayList().get(i);
```

```
    switch (sc.getSelectedNum()) {
```

```
        case CharacterNumber.KNIGHT:
```

```
            hp -= ((Knight) sc.getCharacter()).qSkill(monsterSc);
```

```
            break;
```

```
public int qSkill(SelectedCharacter monsterSc) {
```

```
    dcro.procDamageCalcRequestObject(monsterSc);
```

```
    System.out.printf("%10d - 돌팔매(원거리)\n",
```

```
        calcQuackDamage(dcro));
```

```
    return 0;
```

```
}
```


파티원이랑 하는 보스전은 끝. 이제 원래하던거 그대로 되도록.

```
public void huntStart () throws InterruptedException {  
  
    chooseMember();  
  
    while (true) {  
        printPartyInfo();  
        //wiz.printInfo();  
        doMacroSet(cm);  
        wiz.calcCharacterExp(calcExps(), THIRD);  
    }  
}
```

Adventurer 에 calcCharacterExp가 있음.

```
public void calcCharacterExp (int gettingExps, int charNum) {  
  
    curExp += gettingExps;  
  
    while (curExp - reqExp > 0) {  
        level++;  
  
        incStat(charNum);  
  
        curExp -= reqExp;  
        reqExp *= 1.1;  
    }  
}
```

```
// cm이 필요한 이유는 내가 선택한 캐릭터가  
// 레벨업을 해야하기 때문이다. |  
public void calcCharacterExp(int Exp, CharacterManager cm){  
    //public void calcCharacterExp (int gettingExps, int charNum)  
}
```

질문

```
// cm이 필요한 이유는 내가 선택한 캐릭터가
// 레벨업을 해야하기 때문이다.
public void calcCharacterExp(int exps, CharacterManager cm){
    //public void calcCharacterExp (int gettingExps, int charNum)
    ArrayList<SelectedCharacter> member = cm.getMemberArrayList();
    SelectedCharacter sc;

    for(int i = 0; i < member.size(); i++){
        sc = member.get(i);

        ((Adventurer) sc.getCharacter()).calcCharacterExp(exps, sc.getSelectedNum());
    }
}
```

멤버를 선택했다. 근데 왜 Adventurer로 형변환을 하는걸까요..?

가져온 캐릭터를 Adventurer에 넣기 위해서 인가요? 근데.. 보내는건 sc.getSelectedNum()으로 넘어가는거 아닌가요?