

문제은행[5] 질문 - 3

2022-01-17

1. 이 단계에서 사용자가 선택한 캐릭터(번호)에 따라 -> 해당 캐릭터의 클래스를 객체화 시키고 -> sc에 캐릭터 번호, 캐릭터 객체 저장 -> **member ArrayList에 캐릭터 객체 저장** -> **member를 부르면 그 객체가 소환되는 것?** <<이 루틴이 맞는지?

```
public void procUserInput (int num) {
    SelectedCharacter sc;

    switch (num) {
        case CharacterNumber.KNIGHT:
            Knight kni = new Knight();
            sc = new SelectedCharacter(
                CharacterNumber.KNIGHT, kni);
            member.add(sc);
            break;

        case CharacterNumber.WIZARD:
            Wizard wiz = new Wizard();
            sc = new SelectedCharacter(
                CharacterNumber.WIZARD, wiz);
            member.add(sc);
            break;

        case CharacterNumber.SNIPER:
            Sniper sni = new Sniper();
            sc = new SelectedCharacter(
                CharacterNumber.SNIPER, sni);
            member.add(sc);
            break;
    }
}
```

2.

(1)아래 `cm.getMemberArrayList();`를 호출하면 이전에 `procUserInput` 매서드에서 저장된 `member ArrayList`를 불러오는 것인가?

(ex. 만약 `proUserInput`단계에서 사용자가 1번(Knight)를 입력했다면 `getMemberArrayList`로 return되는 member값은 Knight만? 아니면 모든 캐릭터가 다 불러와지는 것?)

(2) 그 `ArrayList`를 `<SelectedCharcter>` 데이터타입으로 형변환시키는 이유는

`SelectedCharcter`클래스가 캐릭터의 번호와 캐릭터명을 다루는 클래스이기 때문에 아래의 switch문을 수행할 때 캐릭터를 식별하기 위함인가?

```
public void doMacroSet (CharacterManager cm) throws InterruptedException {  
    ArrayList<SelectedCharacter> member = cm.getMemberArrayList();  
    FieldMonster fm = new FieldMonster();  
  
    SelectedCharacter userSc;  
    SelectedCharacter monsterSc = new SelectedCharacter(MonsterNumber.FIELD, fm);  
  
    for (int i = 0; i < member.size(); i++) {  
        userSc = member.get(i);  
  
        switch (userSc.getSelectedNum()) {  
            case CharacterNumber.KNIGHT:  
                System.out.println(((Knight) userSc.getCharacter()).qSkill(monsterSc));  
                break;  
        }  
    }  
}
```

3. **DamageCalcRequestObject dcro**를 쓰는 이유는 알겠지만 역할이 이해가 잘 안 된다. 이 클래스를 통해 Fenryl과 Field가 입력됐을 때를 구분 지을 수 있고 dcro라는 매개변수가 Assassin 클래스에서 Damage를 계산할 때 적용되는 것?

왜 이렇게 해야하는지 이해가 잘 안되는데...만약 ReqeustObject방법을 쓰지 않으면 Assassin 클래스에서 Damage계산할 때 Fenryl일 때 따로 Field 일 때 각각의 코드를 중복해서 사용해야하며 후에 확장성에도 문제가 생기기 때문인가?

```
public class DamageCalcRequestObject {  
    private float pAtk, mAtk;  
    private float hp, mp;  
    private float pDef, mDef;  
    private float str, con, dex, agi, iq, men;  
  
    public void procDamageCalcRequestObject (SelectedCharacter monsterSc) {  
        switch (monsterSc.getSelectedNum()) {  
            case MonsterNumber.FENRYL:  
                procAllData((Fenryl) monsterSc.getCharacter());  
                break;  
  
            case MonsterNumber.FIELD:  
                procAllData((FieldMonster) monsterSc.getCharacter());  
                break;  
        }  
    }  
}
```

**DamageClacRequestObject
class**

```
public int calcDisasterStepDamage (DamageCalcRequestObject dcro) {  
    float sum = 0;  
  
    final int MAX = 1;  
    final int MIN = 0;  
  
    int range = MAX - MIN + 1;  
    float criticalCoef;  
    int criticalFlag;  
  
    for (int i = 0; i < 10; i++) {  
        criticalFlag = (int) (Math.random() * range + MIN);  
        if (criticalFlag % 2 == 0) {  
            criticalCoef = 1.5f;  
        } else {  
            criticalCoef = 1.0f;  
        }  
    }  
  
    sum += ( criticalCoef * (12 * (pAtk - dcro.getpDef()) *  
        ( (dex - dcro.getCon()) * 2.3 + (agi - dcro.getCon()) * 1.5) ) );  
}  
  
return (int) sum;  
}
```

Assassin class