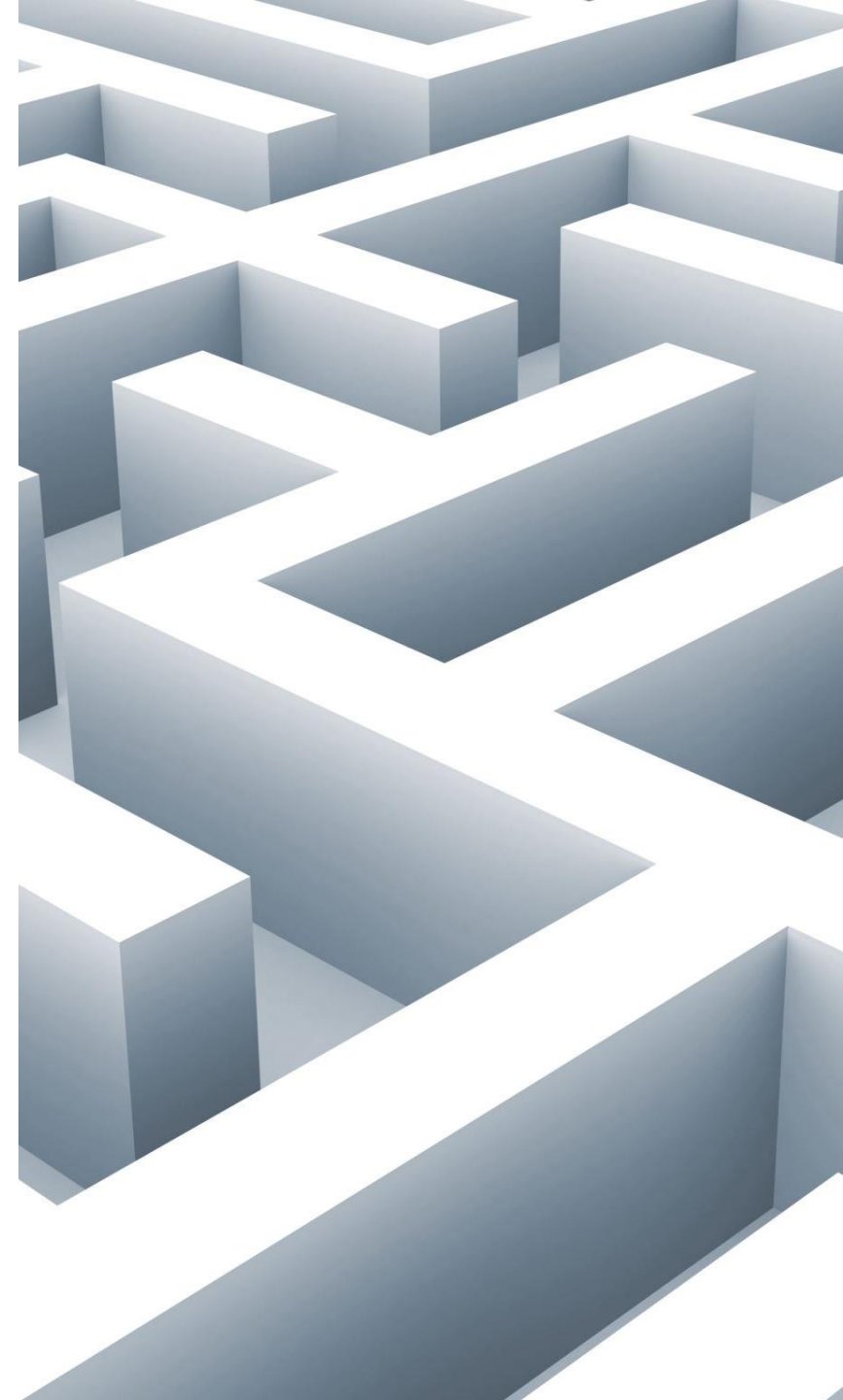


JAVA INTERFACE TASK

한다운



과제 내용

- ◆ Interface 를 사용하기 가장 좋은 예는 게임이다. 아래와 같은 상황을 구현해보도록 한다.

시작	--->	1차 전직	--->	2차 전직
모험가	--->	전사	----->	워 로드
	--->	도적	----->	어쌔신
	--->	마법사	----->	위자드

- ◆ 각각의 캐릭터들은 공격 스킬을 두 개씩 가지고 있다.
- ◆ 전직을 할 때마다 추가적인 전직 스킬 두 개를 얻게 된다.
- ◆ 스킬을 사용할 때 모두가 일관되게 skill1(), skill2(), skill3() 형식으로 데미지가 계산되도록 만들어보자!

프로그램 개요

- ◆ Javastory
: Main 클래스 - 프로그램을 실행
- ◆ Gameplay
: 실질적인 게임 내 여러 작업들이 이루어짐
- ◆ CharacterLevel0 / CharacterLevel1 / CharacterLevel2
: n차 전직 직업군들의 인터페이스
- ◆ Adventurer || Warrior / Thief / Magician || Warlord / Assassin / Wizard
: n차 전직 직업군들의 특성을 관리

Javastory.java

◆ Main 클래스 - 프로그램을 실행

```
// Main 클래스 : 프로그램을 실행
public class Javastory {
    public static void main(String[] args) {
        Gameplay game = new Gameplay();

        game.gameStart();
        game.gameWork();
    }
}
```

프로그램 실행을 위한
기본적인 코드만 작성

Gameplay.java

◆ gameStart()

```
// 사용자 입력 설정 및 새 캐릭터 생성
public void gameStart() {
    System.out.println("*** 자바스토리에 오신 것을 환영합니다 ***");
    System.out.println("새로운 캐릭터를 생성합니다.");

    setScan();
    setAdventurer();
}
```

사용자 입력 설정 및
새 캐릭터 생성

◆ 실행 화면

```
*** 자바스토리에 오신 것을 환영합니다 ***
새로운 캐릭터를 생성합니다.

'모험가' (으)로 0자 전직했습니다~
```

Gameplay.java

◆ gameStart() 코드

```
// 게임 진행
public void gameWork() {
    while(selectWork != 4) {
        System.out.print("캐릭터의 직업을 확인하려면 1, 공격하려면 2" +
            ", 전직하려면 3, 종료하려면 4를 입력하세요 : ");
        selectWork = scan.nextInt();

        switch (selectWork) {

            // 캐릭터의 직업 확인
            case 1:
                System.out.printf("\n당신의 직업은 '%s' 입니다.\n\n", adv.getJob());
                break;
```

< 게임 진행 >

1. 캐릭터 직업 확인

◆ 실행 화면

```
캐릭터의 직업을 확인하려면 1, 공격하려면 2, 전직하려면 3, 종료하려면 4를 입력하세요 : 1

당신의 직업은 '모험가' 입니다.
```

Gameplay.java

◆ gameStart() 코드

```
// 공격 스킬 사용
case 2:
    System.out.print("첫 번째 스킬 사용은 1, 두 번째 스킬 사용은 2를 입력하세요 : ");
    selectSkill = scan.nextInt();

    // 첫 번째 스킬 사용
    if(selectSkill == 1)
        adv.skill1();
    // 두 번째 스킬 사용
    else if(selectSkill == 2)
        adv.skill2();
    // 잘못된 입력 예외 처리
    else
        System.out.println("\n잘못 입력하셨습니다. 공격 MISS \n\n");
    break;
```

< 게임 진행 >

2. 공격 스킬 사용

Gameplay.java

◆ 실행 화면

```
캐릭터의 직업을 확인하려면 1, 공격하려면 2, 전직하려면 3, 종료하려면 4를 입력하세요 : 2
첫 번째 스킬 사용은 1, 두 번째 스킬 사용은 2를 입력하세요 : 1

'간지럽히기' 기술 사용!
총 5의 데미지를 입혔습니다!!

캐릭터의 직업을 확인하려면 1, 공격하려면 2, 전직하려면 3, 종료하려면 4를 입력하세요 : 2
첫 번째 스킬 사용은 1, 두 번째 스킬 사용은 2를 입력하세요 : 2

'주먹으로 때리기' 기술 사용!
총 15의 데미지를 입혔습니다!!

캐릭터의 직업을 확인하려면 1, 공격하려면 2, 전직하려면 3, 종료하려면 4를 입력하세요 : 2
첫 번째 스킬 사용은 1, 두 번째 스킬 사용은 2를 입력하세요 : 3

잘못 입력하셨습니다. 공격 MISS πππ
```

< 게임 진행 >

2. 공격 스킬 사용

Gameplay.java

◆ gameStart() 코드

```
// 현재 직업의 상위 레벨 직업으로 전직
case 3:
    // 누적 데미지(경험치) 30 이상이면 전직 가능
    if(adv.getAttackDamage() >= 30) {
        System.out.print("전사로 전직하려면 1, 도적은 2, 마법사는 3을 입력하세요 : ");
        selectJob = scan.nextInt();

        adv.changeJob(selectJob);
    }
    // 누적 데미지(경험치) 30 미만이면 전직 불가능
    else {
        System.out.println("\n30 이상의 데미지를 입혀야 전직할 수 있습니다.");
        System.out.println("경험을 더 쌓고 오세요.\n");
    }
    break;
```

< 게임 진행 >

3. 현재 직업의 상위 레벨
직업으로 전직

Gameplay.java

◆ 실행 화면

캐릭터의 직업을 확인하려면 1, 공격하려면 2, 전직하려면 3, 종료하려면 4를 입력하세요 : 3

30 이상의 데미지를 입혀야 전직할 수 있습니다.
경험을 더 쌓고 오세요.

캐릭터의 직업을 확인하려면 1, 공격하려면 2, 전직하려면 3, 종료하려면 4를 입력하세요 : 2
첫 번째 스킬 사용은 1, 두 번째 스킬 사용은 2를 입력하세요 : 2

'주먹으로 때리기' 기술 사용!
총 35의 데미지를 입혔습니다!!

캐릭터의 직업을 확인하려면 1, 공격하려면 2, 전직하려면 3, 종료하려면 4를 입력하세요 : 3
전사로 전직하려면 1, 도적은 2, 마법사는 3을 입력하세요 : 1

'전사' (으)로 1차 전직했습니다~

캐릭터의 직업을 확인하려면 1, 공격하려면 2, 전직하려면 3, 종료하려면 4를 입력하세요 : 1

당신의 직업은 '전사' 입니다.

< 게임 진행 >

3. 현재 직업의 상위 레벨
직업으로 전직

Gameplay.java

◆ gameStart() 코드

```
// 게임 종료
case 4:
    System.out.println("\n*** 자바스토리를 종료합니다 ***");
    System.out.println("다음에 꼭 다시 만나요~");
    break;

// 잘못된 입력 예외 처리
default:
    System.out.println("\n잘못 입력하셨습니다.\n");
    break;
```

< 게임 진행 >

4. 게임 종료

5. 잘못된 입력 예외 처리

Gameplay.java

◆ 실행 화면

```
캐릭터의 직업을 확인하려면 1, 공격하려면 2, 전직하려면 3, 종료하려면 4를 입력하세요 : 0  
  
잘못 입력하셨습니다.  
  
캐릭터의 직업을 확인하려면 1, 공격하려면 2, 전직하려면 3, 종료하려면 4를 입력하세요 : 4  
  
*** 자바스토리를 종료합니다 ***  
다음에 꼭 다시 만나요~
```

< 게임 진행 >

4. 게임 종료

5. 잘못된 입력 예외 처리

CharacterLevel0~2.java

◆ n차 전직 직업군들의 인터페이스

```
// 0차 전직 (모험가) 직업군의 인터페이스
public interface CharacterLevel0 {
    public void skill1();
    public void skill2();
}
```

```
// 1차 전직 (전사, 도적, 마법사) 직업군의 인터페이스
public interface CharacterLevel1 {
    public void skill3();
    public void skill4();
}
```

```
// 2차 전직 (워로드, 어쌔신, 워자드) 직업군의 인터페이스
public interface CharacterLevel2 {
    public void skill5();
    public void skill6();
}
```

- 공격 스킬 사용시 사용되는
공통적인 매서드 관리

N차 전직 직업군들의 클래스

◆ Warrior.java 와 Thief.java 클래스 비교

```
// 전사 공격 스킬 1 : 베기 (데미지 15)
public void skill3() {
    setAttackDamage(15);
    System.out.printf("\n'%s' 기술 사용!\n 총 %2d의 데미지를 입었습니다!!\n\n", SKILLSLASH, getAttackDamage());
}
```

```
// 전사 공격 스킬 2 : 들진 (데미지 20)
public void skill4() {
    setAttackDamage(20);
    System.out.printf("\n'%s' 기술 사용!\n 총 %2d의 데미지를 입었습니다!!\n\n", SKILLCHARGE, getAttackDamage());
}
```

```
// 도적 공격 스킬 1 : 단검 날리기 (데미지 15)
public void skill3() {
    setAttackDamage(15);
    System.out.printf("\n'%s' 기술 사용!\n 총 %2d의 데미지를 입었습니다!!\n\n", SKILLDAGGER, getAttackDamage());
}
```

```
// 도적 공격 스킬 2 : 독 뿌리기 (데미지 20)
public void skill4() {
    setAttackDamage(20);
    System.out.printf("\n'%s' 기술 사용!\n 총 %2d의 데미지를 입었습니다!!\n\n", SKILLPOISON, getAttackDamage());
}
```

- 직업군별로 각 직업마다
특성 스킬을 설정함

후기

◆ 게임의 세부적인 내용은 미구현 (시간 부족)

- 전직하면 바뀐 직업의 객체로 작업들이 실행되어야 하는데 아직 불가능
- Adventurer.java 는 모험가의 특성만 저장되어야 하는데
이 클래스가 모든 직업의 부모 클래스로 기본값들을 저장한 상태임
모든 직업을 관리하는 클래스가 필요함