



제가 프로그래밍을 처음 배우기 시작한 이유는 컴퓨터 공학을 완수하거나 그분야의 최고가 되고 싶어서가 아니라



매우 단순한 이유 때문이었어요. 저를 포함한 저희집  
남매들이 서로 같이 재밌게 할 수 있는 뭔가를 만들고 싶었거든요.



간단한 프로그램을 짜고 나서, 조금씩 기능들을 추가했어요. 그리고 새로운 걸 배워야되는 상황이 닥치면



IN ANTONIO SPURS  
처음에는 막막한 느낌이지만 시간이 지나면 감이 오거든요.



악기를 연주하거나 스포츠를 하는것과 별로 다르게 없어요.



## Java 프로그래밍 수업내용 정리 및 실습 과제

책을 보거나 인터넷을 찾아보면서 배우고 거기에 저만의 방식을 추가했어요.



(NBA 농구선수 크리스 보쉬) 프로그래밍은 누구나 배울 수 있는 거예요.  
물론 처음엔 겁이나죠. 하지만 그것은 비단 프로그래밍에만 해당되는게 아닙니다.



처음 배울때 겁이 안나는게 뭐가 있을까요?



프로그래밍은 천재들만 할 수 있는것이 아니예요. 의지력이 필요할 뿐입니다.



• 2020.04.16.목  
• B반 송명훈

더하기 빼기... 그게 거의 전부죠.

# 목 차

1. 수업 내용 정리
2. 실습화면 캡처
3. 연습문제 8-1
4. 연습문제 8-2
5. 연습문제 8-3
6. 연습문제 8-4



## 1. 수업 내용 정리

# 1. 수업 내용 정리 [객체 지향 프로그래밍]

## 객체 지향을 배우는 이유

1. 일반적으로 소프트웨어 개발은 대규모, 팀 단위 개발로 이루어지며 혼자서 개발하는 것이 아니다.
2. 소스코드가 길어지면 개발자 본인도 파악이 어려움. 당연히 다른 개발자들은 더 어려움.  
⇒ 결과적으로 가독성 향상 및 생산성 증가를 위함.
3. 객체 지향을 배우는 요령
  - 생각하는 방법 (사고)
  - 단순 암기가 아닌 이해, 전체적인 그림을 그려가면서.
4. 객체 지향이란?
  - 소프트웨어 개발 시 구성요소들을 "객체(Object)"로 모듈화 하는 것
  - 프로그램 유지 보수 용이, 각 모듈의 재사용 가능
  - 객체 (Object): 현실세계에 존재하는 모든 것 (All, Everything)
  - 클래스 (Class): 컴퓨터 세계에서 객체를 구현한 틀
5. 객체 지향의 3대 기능
  - 캡슐화 (Encapsulation)
  - 상속 (Inheritance)
  - 다형성 (Polymorphism)

# 1. 수업 내용 정리 [인스턴스와 클래스]

## 인스턴스(Instance)와 클래스(Class)

### 1. 오브젝트 도출 순서

- 각 오브젝트가 가지고 있을만 한 "속성"이나 "동작"을 생각해 종류와 내용 정의
- 각 오브젝트를 가상 세계에 도출, 동작 시켜 본다.
- 용어정리
  - 오브젝트(Object) : 현실 세계의 모든 객체
  - 클래스 (Class) : 오브젝트를 가상세계용으로 구체화한 것 (붕어빵 틀)
  - 인스턴스 (Instance): 클래스를 활용해 메모리 상에 생성한 것 (붕어빵)

### 1. 클래스 명과 멤버변수명의 명명 규칙

- 클래스명 [명사]: 단어 맨 처음은 대문자 (Ex] Hero, MonsterInfo)
- 필드 명 [명사]: 최초 이외의 단어의 맨 처음은 대문자 (Camel notation, Ex] mLevel, mItemList)
- 메소드 명 [동사]: 최초 이외의 단어의 맨 처음은 대문자 (Camel notation, Ex] attack, findWeakPoint)

### 2. 클래스 정의에 따른 효과

1. 클래스로 인스턴스 생성 가능 (빵 틀로 빵 굽기)
2. 클래스로 생성한 인스턴스를 새로운 타입의 변수로 사용 가능  
클래스가 새로운 변수 타입이 됨.

(Ex] Hero class 정의 ⇨ int, double처럼 Hero자료형 변수사용가능)

# 1. 수업 내용 정리 [인스턴스와 클래스]

## 인스턴스(Instance)와 클래스(Class)

### 1. 필드와 메소드

- 「클래스에서의 속성 = 필드, 동작 (or 기능) = 메소드」로 구현
- final 키워드가 붙은 필드(=멤버 변수, 클래스 내에서 전역변수와 동일한 개념)는 상수 필드. 변할 수 없는 값. 초기화 시 저장한 값 그대로 쭉 간다.
- this는 클래스 자신의 인스턴스를 표시하는 키워드

### 2. 인스턴스화

- new 연산자를 사용하여 클래스로부터 인스턴스를 생성
  - ✓ 힙(Heap) 메모리에 데이터를 저장할 공간을 할당하고 그 공간의 참조 값을 객체에 반환  
그러므로 인스턴스는 reference type
  - ✓ C언어의 malloc 함수와 유사
- 어떤 클래스 타입 변수에 인스턴스가 담겨있을 땐 "변수명.필드명" or "변수명.메소드명()"으로 인스턴스의 필드나 메소드 사용 가능.

### 3. 인스턴스(Instance): 클래스에 의해 생성된 객체. 클래스와 다른 개념임을 주의.

- String strName = new String();
  - ✓ 이 때, strName은 String 클래스의 인스턴스이며 String type 변수이기도 하다.



## 2. 실습화면 캡처

## 2. 실습 화면 캡처

### 8-1. main 메소드 작성

```
Ex8_1.java x
1 package Day6Chap8;
2
3 public class Ex8_1 {
4     public static void main(String[] args) {
5         // 용사여, 가상 세계에 탄생하라
6         // 괴물 버섯이여, 가상 세계에 탄생하라
7         // 용사여, 싸워라
8         // 괴물 버섯이여, 도망가라
9         // 이 프로그램 작성을 위해 필요한 클래스 생각 해보자
10        // 1. main 메소드를 가지는 1개의 "신 클래스"
11        // 2. "등장 인물 클래스"
12    }
13 }
```

### 8-2. Hero 클래스를 Java 코드로 표현

```
Ex8_1.java x Hero.java x
1 package Day6Chap8;
2
3 public class Hero {
4     String name;
5     int hp;
6
7     void attack() {}
8     void run() {}
9     void sit(int sec) {}
10    void slip() {}
11    void sleep() {}
12 }
```

### 8-6. 필드를 상수로 선언

```
Ex8_1.java x Hero.java x Kinoko.java x
1 package Day6Chap8;
2
3 public class Kinoko {
4     int hp;
5     final int LEVEL = 10;
6 }
```



## 2. 실습 화면 캡처

```
Hero.java x
1 package Day6Chap8;
2
3 public class Hero {
4     String name;
5     int hp;
6
7     void attack() {}
8     void run() {}
9     void sit(int sec) {}
10    void slip() {}
11    void sleep() {
12        this.hp = 100;
13        System.out.println(this.name + "는 잠을 자고 회복했다.");
14    }
15 }
```

8-7. "잠자기" 동작을 작성

8-8. 메소드 "앞다",  
"도망가다", "넘어지다"를 추가

```
Hero.java x
3 public class Hero {
4     String name;
5     int hp;
6
7     void attack() {}
8     void run() {
9         System.out.println(this.name + "는 도망쳤다!");
10        System.out.println("GAME OVER");
11        System.out.println("최종 HP는 " + this.hp + " 입니다");
12    }
13    void sit(int sec) {
14        this.hp += sec; // 앉은 초 만큼 HP 가 증가
15        System.out.println(this.name + "는 " + sec + "초 앉았다");
16        System.out.println("HP가 " + sec + "포인트 회복되었다");
17    }
18    void slip() {
19        this.hp -= 5;
20        System.out.println(this.name + "는 넘어졌다");
21        System.out.println("5의 데미지!");
22    }
23    void sleep() {
24        this.hp = 100;
25        System.out.println(this.name + "는 잠을 자고 회복했다.");
26    }
27 }
```

## 2. 실습 화면 캡처

### 8-10. "신 클래스"를 작성

```
Ex8_1.java x
1 package Day6Chap8;
2
3 public class Ex8_1 {
4     public static void main(String[] args) {
5         // 가상 세계에 용사를 생성
6
7         // 생성된 용사에게 최초의 HP와 이름을 설정
8
9         // 용사에게 '5초 앓기', '넘어지기', '25초 앓기', '도망'을 지시
10    }
11 }
```

### 8-11.

```
Ex8_1.java x
1 package Day6Chap8;
2
3 public class Ex8_1 {
4     public static void main(String[] args) {
5         // 가상 세계에 용사를 생성
6         Hero hero = new Hero();
7
8         // 생성된 용사에게 최초의 HP와 이름을 설정
9
10        // 용사에게 '5초 앓기', '넘어지기', '25초 앓기', '도망'을 지시
11    }
12 }
```

```
Ex8_1.java x
1 package Day6Chap8;
2
3 public class Ex8_1 {
4     public static void main(String[] args) {
5         // 가상 세계에 용사를 생성
6         Hero hero = new Hero();
7
8         // 생성된 용사에게 최초의 HP와 이름을 설정
9         hero.name = "명훈";
10        hero.hp = 100;
11        System.out.println("용사 " + hero.name + " 를 생성했습니다!");
12
13        // 용사에게 '5초 앓기', '넘어지기', '25초 앓기', '도망'을 지시
14    }
15 }
```

### 8-12.

## 2. 실습 화면 캡처

```
.java x
public class Ex8_1 {
    public static void main(String[] args) {
        // 가상 세계에 용사를 생성
        Hero hero = new Hero();

        // 생성된 용사에게 최초의 HP와 이름을 설정
        hero.name = "명훈";
        hero.hp = 100;
        System.out.println("용사 " + hero.name + " 를 생성했습니다!");

        // 용사에게 '5초 앓기', '넘어지기', '25초 앓기', '도망'을 지시
        hero.sit( sec: 5);
        hero.slip();
        hero.sit( sec: 25);
        hero.run();
    }
}
```

8-13.

```
Ex8_1 x
"C:\Program Files\Java\jdk-13.0.2\
용사 명훈 를 생성했습니다!
명훈는 5초 앓았다
HP가 5포인트 회복되었다
명훈는 넘어졌다
5의 데미지!
명훈는 25초 앓았다
HP가 25포인트 회복되었다
명훈는 도망쳤다!
GAME OVER
최종 HP는 125 입니다

Process finished with exit code 0
```

8-13. 결과

## 2. 실습 화면 캡처

### 8-14. 객체지향이 아닌 방법으로 작성한 프로그램

```
4.java ×
public class Ex8_14 {
    public static void main(String[] args) {
        // 가상 세계에 용사를 생성
        int yongsa_hp = 100;
        String yongsa_name = "명훈";

        // 생성된 용사에게 최초의 HP와 이름을 설정
        System.out.println("용사 " + yongsa_name + "를 생성했습니다!");

        // 용사에게 '5초 앓기', '넘어지기', '25초 앓기', '도망'을 지시
        yongsa_hp += 5;
        System.out.println(yongsa_name + "는 5초 앓았다");
        System.out.println("HP가 5포인트 회복되었다");
        yongsa_hp -= 5;
        System.out.println(yongsa_name + "는 넘어졌다");
        System.out.println("5의 데미지!");
        yongsa_hp += 25;
        System.out.println(yongsa_name + "는 25초 앓았다");
        System.out.println("HP가 25포인트 회복되었다");
        System.out.println(yongsa_name + "는 도망쳤다!");
        System.out.println("GAME OVER");
        System.out.println("최종 HP는 " + yongsa_hp + " 입니다");
    }
}
```

Ex8\_14 ×

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" -Djava.class.path=.\Ex8_14.class -Djava.library.path=.\lib\Ex8_14.dll
용사 명훈를 생성했습니다!
명훈는 5초 앓았다
HP가 5포인트 회복되었다
명훈는 넘어졌다
5의 데미지!
명훈는 25초 앓았다
HP가 25포인트 회복되었다
명훈는 도망쳤다!
GAME OVER
최종 HP는 125 입니다
```

Process finished with exit code 0

8-14. 출력 결과

## 2. 실습 화면 캡처

```
3 public class Kinoko {
4     int hp;
5     final int LEVEL = 10;
6     char suffix;
7
8     void run() {
9         System.out.println("괴물 버섯 " + this.suffix + "는 도망갔다!");
10    }
11 }
```

8-15. 괴물 버섯 클래스의 정의

```
Ex8_16.java x
3 public class Ex8_16 {
4     public static void main(String[] args) {
5         Hero hero = new Hero();
6         hero.name = "명훈";
7         hero.hp = 100;
8
9         Kinoko kinoko1 = new Kinoko();
10        kinoko1.hp = 50;
11        kinoko1.suffix = 'A';
12
13        Kinoko kinoko2 = new Kinoko();
14        kinoko1.hp = 48;
15        kinoko1.suffix = 'B';
16
17        // 모험의 시작
18        hero.sit();
19        kinoko1.run();
20        kinoko2.run();
21        hero.run();
22    }
```

8-16. 가상세계에서 용사와 괴물 버섯 2마리를 생성

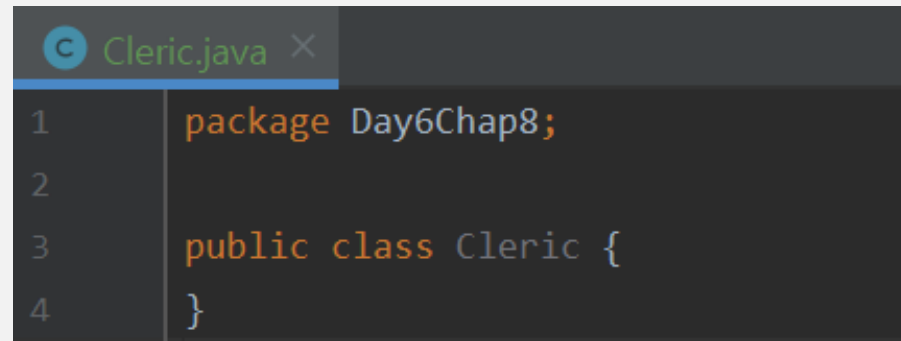


### **3. 연습문제 8-1**

## 3. 연습문제 8-1

### 8-1. 문제

- 현실세계의 성직자 "클레릭"를 표현하는 클래스 Cleric를 작성 하시오.  
속성이나 동작은 선언할 필요 없습니다. (속은 아무것도 작성하지 않아도 됨)



```
Cleric.java x
1 package Day6Chap8;
2
3 public class Cleric {
4 }
```



## 4. 연습문제 8-2



## 4. 연습문제 8-2

### 8-2. 문제

- 성직자는 용사처럼 이름과 HP를 가지고 있고, 추가로 마법을 사용하기 위한 MP를 가지고 있다.
- 거기에 연습 8-1에서 선언한 내용이 없는 Cleric 클래스에 "이름", "HP", "최대 HP", "MP", "최대 MP"를 속성으로 추가 하시오.
- 또한 HP와 최대 HP는 정수로 초기치 50, MP와 최대 MP는 정수로 초기치 10으로 하고, 최대 HP와 최대 MP는 상수 필드로 선언 하시오.

```
Cleric.java ×  
3     public class Cleric {  
4         // "이름", "HP", "최대 HP", "MP", "최대 MP"를 속성으로 추가 하시오.  
5         // 또한 HP와 최대 HP는 정수로 초기치 50, MP와 최대 MP는  
6         // 정수로 초기치 10으로 하고, 최대 HP와 최대 MP는 상수 필드로 선언 하시오.  
7         String name;  
8         int hp = 50;  
9         int mp = 10;  
10        final int hpMax = 50;  
11        final int mpMax = 10;
```



## 5. 연습문제 8-3

## 5. 연습문제 8-3

### 8-3. 문제

- 성직자는 "셀프 에이드"라는 마법을 사용할 수 있고, MP를 5 소비하는 것으로 자신의 HP를 최대 HP까지 회복할 수 있다.
- 연습 8-2에 선언한 Cleric 클래스에 "selfAid()" 메소드를 추가 하시오
- 또한, 이 메소드는 인수가 없고 리턴 값도 없다.

```
Cleric.java x
3   public class Cleric {
4       String name;
5       int hp = 50;
6       int mp = 10;
7       final int hpMax = 50;
8       final int mpMax = 10;
9       // 성직자는 "셀프 에이드"라는 마법을 사용할 수 있고,
10      // MP를 5 소비하는 것으로 자신의 HP를 최대 HP까지 회복할 수 있다.
11      // 연습 8-2에 선언한 cleric 클래스에 "selfAid()" 메소드를 추가 하시오
12      // 또한, 이 메소드는 인수가 없고 리턴 값도 없다.
13      void selfAid() {
14          this.mp -= 5;
15          this.hp = hpMax;
16      }
```



## 6. 연습문제 6-4

## 6. 연습문제 8-4

### 8-4. 문제

- 성직자는 "기도하기" (pray)라는 행동을 취할 수 있고, 자신의 MP를 회복한다.
- 회복량은 기도한 시간(초)에 랜덤하게 0 ~ 2포인트의 보정을 한 양이다 (3초 기도하면 회복량은 3 ~ 5포인트).
- 단, 최대 MP보다 더 회복하는 것은 불가능 하다.
- 그리고, 연습 8-3에서 선언한 Cleric 클래스에 "Pray()" 메소드를 추가하시오.
- 이 메소드는 인수에 "기도할 시간(초)"를 지정할 수 있고, 리턴 값은 "실제로 회복된 MP 양"을 반환한다.

## 6. 연습문제 6-4

```
Cleric.java x
3 public class Cleric {
4     String name;
5     int hp = 50;
6     int mp = 10;
7     final int hpMax = 50;
8     final int mpMax = 10;
9
10    void selfAid() {
11        this.mp -= 5;
12        this.hp = hpMax;
13    }
14    // 성직자는 "기도하기" (pray)라는 행동을 취할 수 있고, 자신의 MP를 회복한다.
15    // 인수로 기도한 시간(초)를 지정
16    int pray(int praySec) {
17        // 회복량 = 기도한 시간(초) + 0 ~ 2 (랜덤)
18        // (3초 기도하면 회복량은 3 ~ 5포인트).
19        // incMp = pray에 의해 회복하는 MP 양
20        int incMp = praySec + new java.util.Random().nextInt( bound: 3);
21        // 최대 MP보다 더 회복하는 것은 불가능.
22        // 총 4가지 경우의 수 존재
23        if(mp == mpMax || praySec <= 0) {
24            incMp = 0;
25        } else if(mp + incMp >= mpMax) {
26            incMp = mpMax - mp;
27            mp = mpMax;
28        } else {
29            mp += incMp;
30        }
31        return incMp; // 리턴 값은 "실제로 회복된 MP 양"
32    }
33 }
```

```
// 성직자는 "기도하기" (pray)라는 행동을 취할 수 있고, 자신의 MP를 회복한다.
// 인수로 기도한 시간(초)를 지정
int pray(int praySec) {
    // 회복량 = 기도한 시간(초) + 0 ~ 2 (랜덤)
    // (3초 기도하면 회복량은 3 ~ 5포인트).
    // incMp = pray에 의해 회복하는 MP 양
    int incMp = praySec + new java.util.Random().nextInt( bound: 3);
    // 최대 MP보다 더 회복하는 것은 불가능.
    // 총 4가지 경우의 수 존재
    if(mp == mpMax || praySec <= 0) {
        incMp = 0;
    } else if(mp + incMp >= mpMax) {
        incMp = mpMax - mp;
        mp = mpMax;
    } else {
        mp += incMp;
    }
    return incMp; // 리턴 값은 "실제로 회복된 MP 양"
}
```

먼저 incMp 값 결정  
각 경우에 따라 incMp 값 수정

최종적으로 결정된 incMp값 return

## 6. 연습문제 6-4

# Cleric 클래스 테스트

Cleric class test를 위한  
main 코드 작성

```

3 ▶ public class PracticMain {
4   @ static void printStatus(Cleric cleric) {
5       System.out.println("cleric " + cleric.name + "의 상태");
6       System.out.println("HP/maxHP: " + cleric.hp + "/" + cleric.hpMax);
7       System.out.println("MP/maxMP: " + cleric.mp + "/" + cleric.mpMax + "
8   }
9 ▶   public static void main(String[] args) {
10       Cleric cleric = new Cleric();
11       cleric.name = "명훈";
12       printStatus(cleric);
13       cleric.hp -= 25;    // selfAid test를 위해 hp 감소
14       printStatus(cleric);
15       cleric.selfAid();
16       printStatus(cleric);
17       cleric.pray( praySec: 3);
18       printStatus(cleric);
19   }
20 }

```

## 첫번째 test

## 두번째 test

PracticecMain × "C:\Program Files\Java\bin\java.exe" -Djava.class.path=.\classes\ -Djava.library.path=.\lib\

cleric 명훈의 상태  
HP/maxHP: 50/50  
MP/maxMP: 10/10

cleric 명훈의 상태  
HP/maxHP: 25/50  
MP/maxMP: 10/10

cleric 명훈의 상태  
HP/maxHP: 50/50  
MP/maxMP: 5/10

cleric 명훈의 상태  
HP/maxHP: 50/50  
MP/maxMP: 8/10

cleric 명훈의 상태  
HP/maxHP: 50/50  
MP/maxMP: 10/10

랜덤 보정 확인